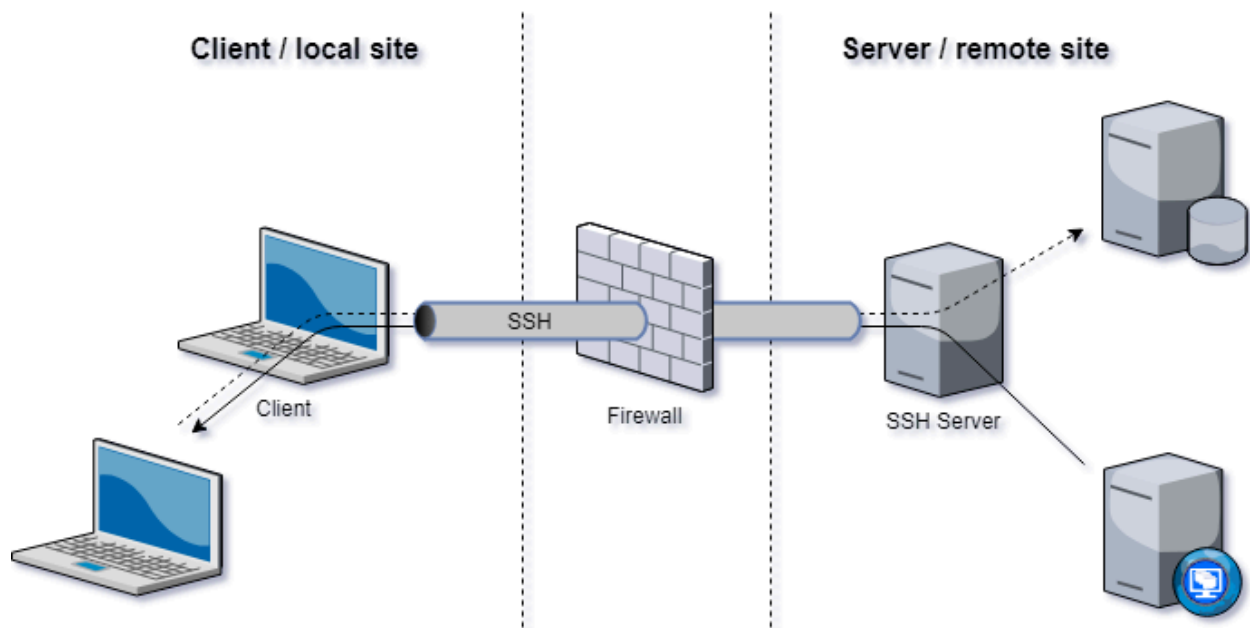




IES Zaidín Vergeles

\*\*\*\*\*

# SSH Tunneling



Jose Almirón López, 26 de Abril de 2024

# Tabla de contenidos

<b>Servidor SSH</b>	<b>2</b>
<b>Configuración del Túnel SSH con PuTTY</b>	<b>2</b>
<b>Configura Firefox para que utilice ese túnel al navegar</b>	<b>4</b>

.....

# Servidor SSH

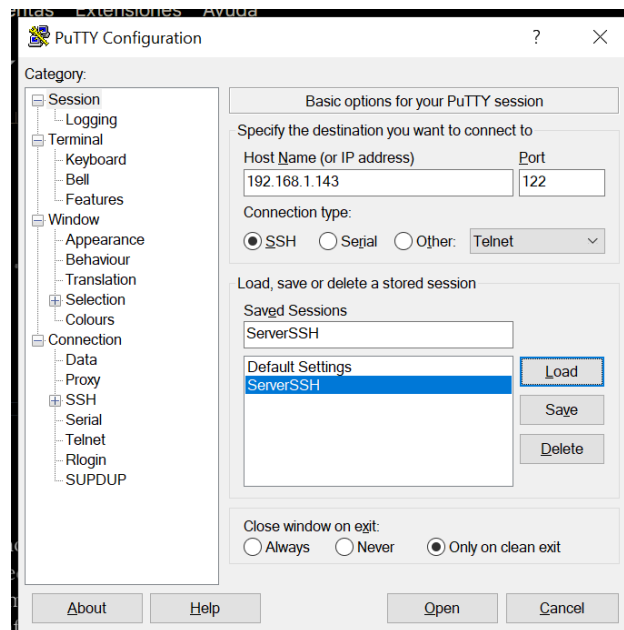
Para empezar, necesitaremos configurar nuestro servidor de SSH, asegurándonos de que esté activo.

```
jose@jose:~$ sudo service ssh status
[sudo] password for jose:
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-04-26 10:18:29 UTC; 8h ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 620 (sshd)
     Tasks: 1 (limit: 2221)
    Memory: 6.7M
       CPU: 165ms
   CGroup: /system.slice/ssh.service
           └─620 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

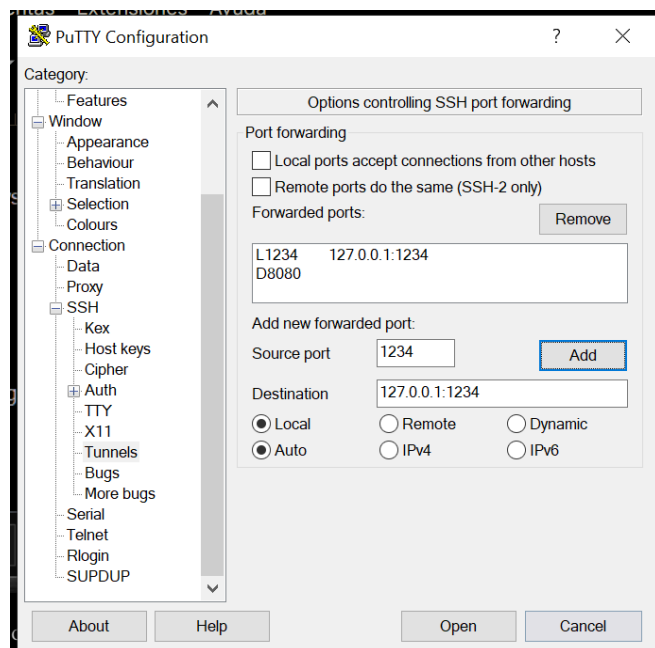
Apr 26 10:18:29 jose systemd[1]: Started OpenBSD Secure Shell server.
Apr 26 10:19:30 jose sshd[739]: User jose from 192.168.1.133 not allowed because not listed in AllowUsers
Apr 26 10:19:31 jose sshd[739]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.133 user=jose
Apr 26 10:19:33 jose sshd[739]: Failed password for invalid user jose from 192.168.1.133 port 53711 ssh2
Apr 26 10:19:37 jose sshd[739]: Failed password for invalid user jose from 192.168.1.133 port 53711 ssh2
Apr 26 10:19:39 jose sshd[739]: error: Received disconnect from 192.168.1.133 port 53711:13: Unable to authenticate [preauth]
Apr 26 10:19:39 jose sshd[739]: Disconnected from invalid user jose 192.168.1.133 port 53711 [preauth]
Apr 26 10:19:39 jose sshd[739]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.133 user=jose
Apr 26 14:25:11 jose sshd[1459]: Accepted password for security from 192.168.1.133 port 58860 ssh2
Apr 26 14:25:11 jose sshd[1459]: pam_unix(sshd:session): session opened for user security(uid=1001) by (uid=0)
jose@jose:~$
```

## Configuración del Túnel SSH con PuTTY

Una vez hayamos iniciado el servicio, abrimos PuTTY y procedemos a utilizar la dirección IP y el puerto como lo haríamos normalmente para establecer la conexión.



Sin embargo, para crear un túnel SSH, añadimos un paso adicional. En el árbol de navegación a la izquierda, seleccionamos **Connection > SSH > Tunnels**. Aquí ingresamos el puerto que deseamos conectar localmente como el 'Source port' (puerto fuente). Luego, agregamos la dirección de destino (el servidor disponible a través de nuestra puerta SSH, pero no en nuestro equipo local) como 'Destination' (destino), tal y como se ilustra en la siguiente imagen:



Al establecer un túnel SSH dinámico, creamos un proxy SOCKS en la máquina local que dirige todo el tráfico a través del servidor SSH remoto. Mediante el comando `SSH -D 8080`, configuramos este proxy SOCKS en el puerto 8080 de la máquina local.

Sin embargo, algunos programas o aplicaciones no pueden utilizar directamente un proxy SOCKS. En su lugar, requieren un proxy HTTP o un proxy SOCKS con una configuración específica de dirección en lugar de simplemente un puerto. En tales casos, es necesario configurar un túnel local para redirigir el tráfico hacia una dirección específica en la máquina local.

Para finalizar, hacemos clic en 'Open' (abrir) para establecer la conexión con el servidor SSH. Ingresamos las credenciales como lo haríamos normalmente. Ahora, todo el tráfico del 'Source Port' (puerto fuente) de tu localhost será dirigido hacia el 'Destination' (destino). Si el puerto de destino proporciona un servicio web, podemos abrir una pestaña en el navegador y visitar `http://127.0.0.1` (en este ejemplo).

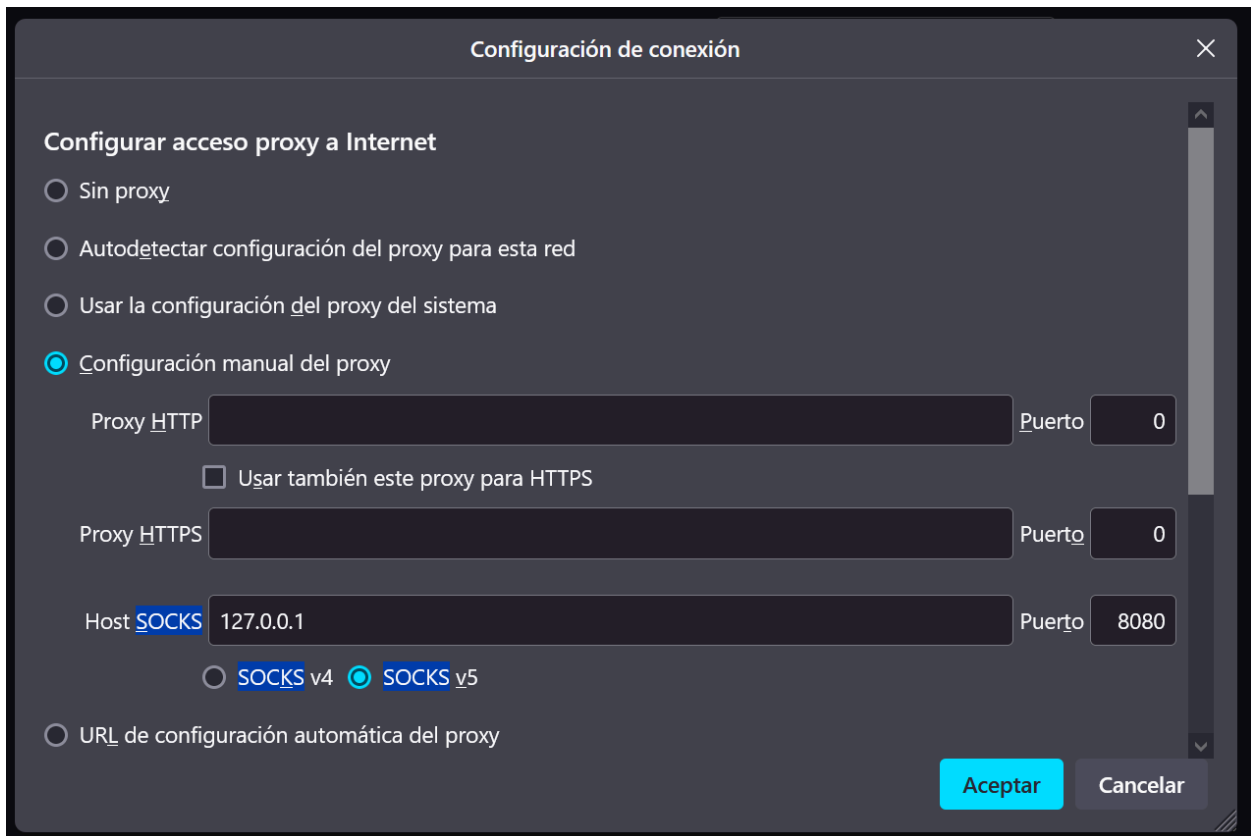
Una alternativa más simple a esto sería utilizar la línea de comandos o una terminal en lugar de usar PuTTY. Con un solo comando, podemos obtener el mismo resultado de manera más directa

```
ssh -p 122 -ND 8080 security@192.168.1.143
```

# Configura Firefox para que utilice ese túnel al navegar

Para que este proceso funcione correctamente, necesitaremos utilizar un proxy. Una opción conveniente es utilizar Firefox, ya que nos permite configurar uno de forma muy sencilla. Para ello, nos dirigimos a los ajustes del navegador y buscamos en la barra de búsqueda la opción '**SOCKS**'.

En la configuración de conexión, seleccionaremos '**Configurar manualmente el proxy**'. En el campo '**Host SOCKS**', utilizaremos la dirección de localhost junto con el puerto 8080, que es el que habíamos configurado previamente en PuTTY.



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  Puerto

☐ Usar también este proxy para HTTPS

Proxy HTTPS  Puerto

Host **SOCKS**  Puerto

☐ SOCKS v4 ☒ SOCKS v5

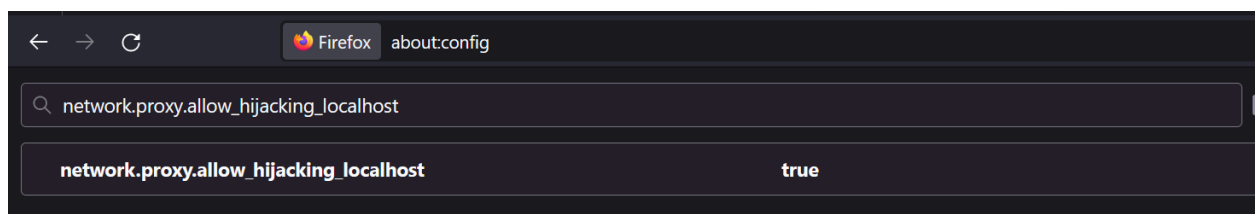
☐ URL de configuración automática del proxy

Aceptar Cancelar

He instalado el servicio Apache en el servidor para verificar el funcionamiento del túnel SSH."

```
jose@jose: ~  
jose@jose:~$ sudo apt install apache2  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils bzip2 file libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.3-0 libmagic-mgc  
  libmagic1 mailcap mime-support  
Suggested packages:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw bzip2-doc  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils bzip2 file libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.3-0  
  libmagic-mgc libmagic1 mailcap mime-support  
0 upgraded, 16 newly installed, 0 to remove and 6 not upgraded.  
Need to get 2520 kB of archives.  
After this operation, 16.2 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

De esta manera al usar localhost desde firefox, nos debería salir el servicio apache que acabamos de instalar, en caso de que esto no funciona tendremos que ir a **about:config**, y activar la opción de **network.proxy.allow\_hijacking\_localhost** que controla si se permite que los complementos de proxy modifiquen la configuración de proxy para localhost.



Finalmente, gracias al túnel y a las configuraciones realizadas, podemos acceder a localhost y veremos el servicio de Apache como si estuviéramos utilizando realmente el servidor

