

Crea tu propio proxy para navegar desde la escuela o el trabajo

Por Guillermo Esteves — 04/02/2007

(Nota: Modifiqué el título porque creo que el anterior era un poco confuso. El permalink sigue siendo el mismo en caso de que alguien haya enlazado aquí.)

Un lector me ha escrito para preguntarme si existe alguna forma de poder navegar en Internet en sitios donde su uso está regulado, como oficinas, escuelas y universidades donde bloquean ciertas páginas y/o puertos. Puesto que ya he escrito antes acerca de como controlar [Azureus](#) y [uTorrent](#) remotamente, pensé que esta es una buena excusa para sacar a Sozter! del semi-retiro y ayudar al amigo lector a ~~evadir sus responsabilidades laborales~~ a navegar en Internet para usos totalmente legítimos. Aclaro que este tutorial es una traducción/parfraseo de un excelente [tutorial](#), escrito en el difunto blog Unrequited Narcissism.

En fin, lo que vamos a hacer, en pocas palabras, es tomar [el tráfico](#) de nuestro navegador en la escuela/oficina/universidad y mandarlo a través de un túnel SSH encriptado a nuestra PC en la casa, la cual estará corriendo un servidor proxy. Nuestra PC en casita se conectará a la página a la que queremos acceder usando nuestra conexión casera, y enviará los datos de vuelta a través del túnel. Nuestro jefe/profe/administrador no verá ni sabrá nada acerca del tráfico ni de los sitios que estamos viendo, excepto que todo está misteriosamente oculto. Hmm.

Para este tutorial necesitaremos:

1. La PC que tendremos en casa (que llamaré **PC1**), y una en la oficina/escuela/universidad (que la llamaré en adelante **PC2**). Ambas deben tener [Windows XP](#). Los geeks de Linux probablemente tienen los conocimientos para hacer esto por sí mismos, y los usuarios de Mac... bueno, si hay suficiente demanda más adelante haré un tutorial específico para Mac.

2. [OpenSSH para Windows](#)
3. [PuTTY](#)
4. [Privoxy](#)
5. Voy a asumir que sabes como abrir los [puertos en el firewall](#) de Windows (o sea cual sea el que uses), así como en el router, en caso de tenerlo. En

[PortForward](#) hay bastante información al respecto.

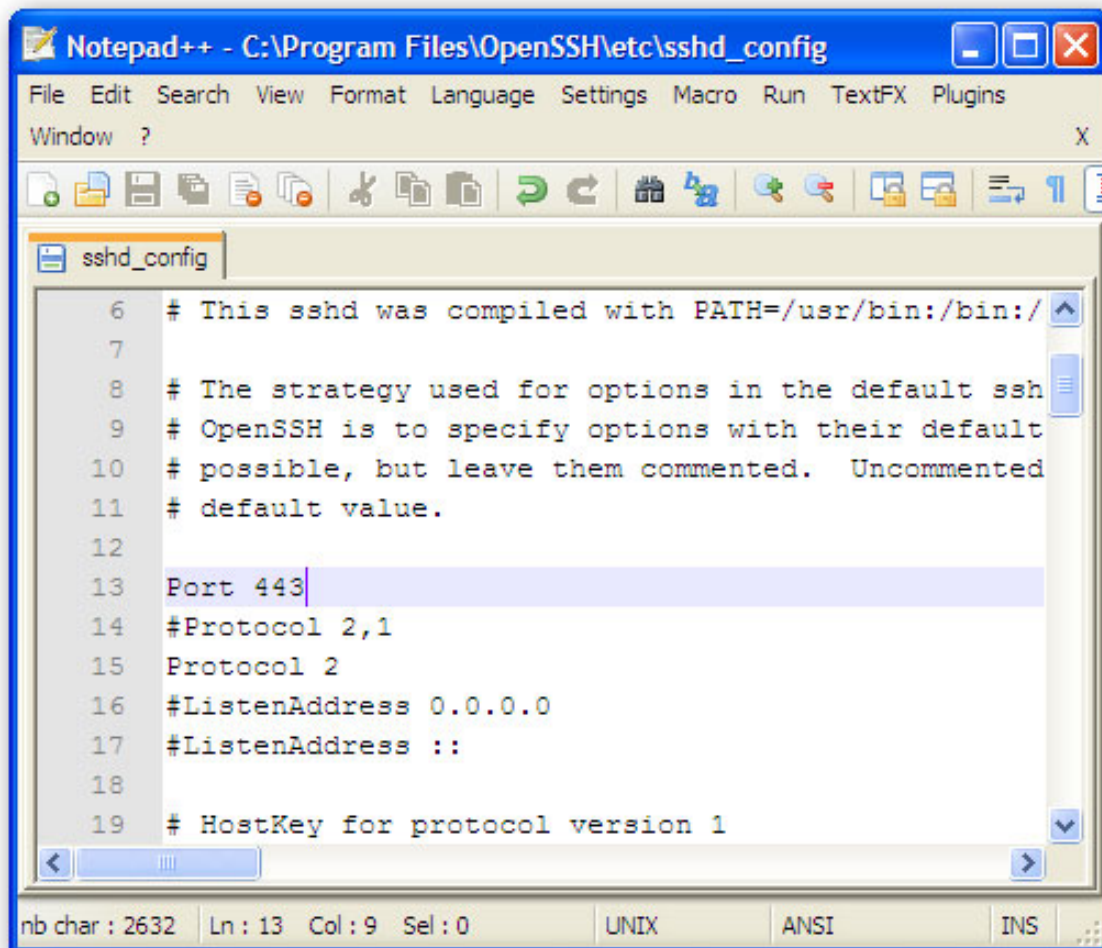
Instalando OpenSSH

Lo primero que vamos a hacer es instalar [OpenSSH](#), un servidor SSH gratuito para Windows, en la PC1. SSH (Secure Shell) es un protocolo y un programa que permite acceder remotamente a otra computadora a través de una red, ejecutar comandos, transferir archivos, etc. Puedes leer más información acerca de SSH en [Wikipedia](#). Descárgalo, instálalo y ya.



Una vez instalado, probablemente vamos a querer cambiar el puerto en el que esta cosa va a funcionar. SSH, por defecto, corre en el puerto 22. Sin embargo si estás en un lugar en el que Internet está restringido es probable que este puerto esté cerrado, ya sea por razones de productividad o de seguridad. Otra opción es el puerto 80, que es casi totalmente seguro que esté abierto. Sin embargo, este puerto puede traerte problemas con tu proveedor de Internet, puesto que algunos proveedores no les agrada que uno sirva tráfico web en computadoras caseras. Finalmente, el puerto 443 puede ser una buena opción, ya que es usado por páginas con SSL así que es probable que esté abierto. En caso de decidir cambiar el puerto, abre el archivo `\Archivos de Programa\OpenSSH\etc\sshd_config` con un editor de texto (me gusta [Notepad++](#)), busca una línea que dice "#Port 22", descoméntala (es decir, quita el #) y cambia el número del

puerto de 22 al que vayas a usar. Guarda el archivo.

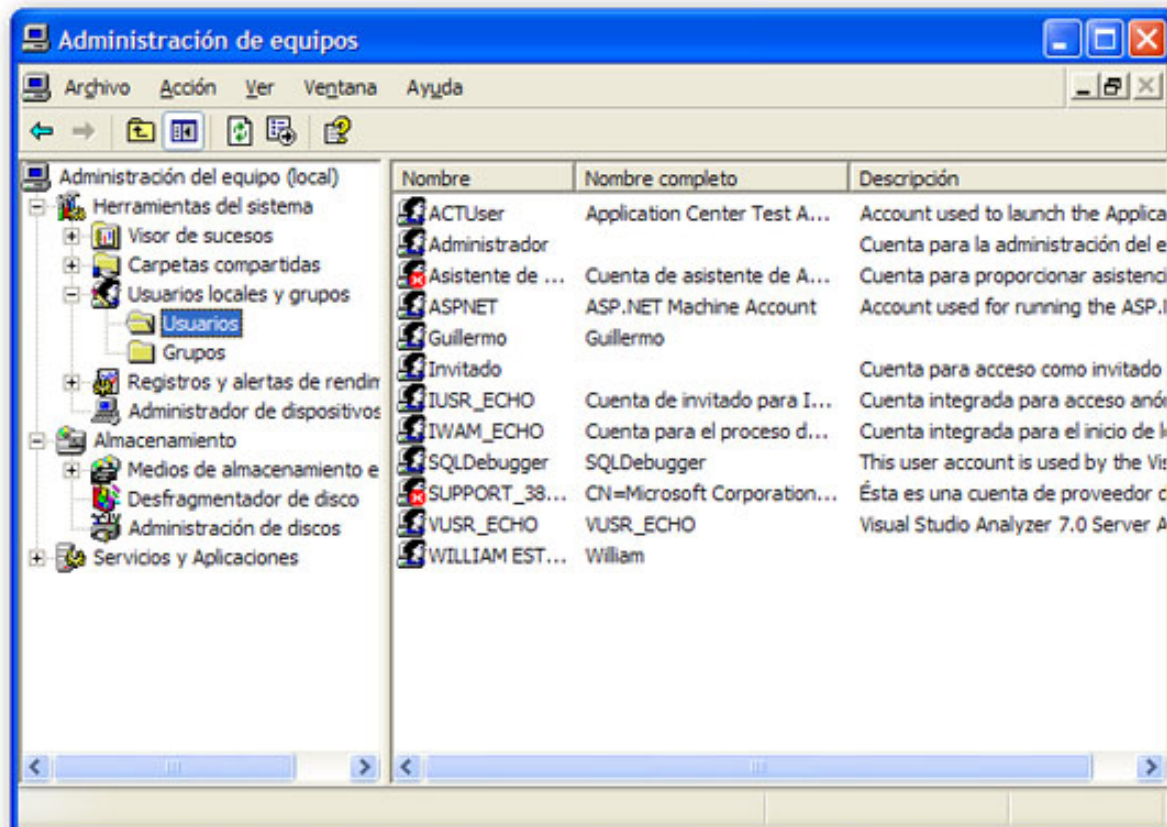


```
6 # This sshd was compiled with PATH=/usr/bin:/bin:/
7
8 # The strategy used for options in the default ssh
9 # OpenSSH is to specify options with their default
10 # possible, but leave them commented. Uncommented
11 # default value.
12
13 Port 443
14 #Protocol 2,1
15 Protocol 2
16 #ListenAddress 0.0.0.0
17 #ListenAddress ::
18
19 # HostKey for protocol version 1
```

nb char : 2632 Ln : 13 Col : 9 Sel : 0 UNIX ANSI INS

Sólo nos falta agregar un usuario para este servidor. Para este paso inicia sesión en Windows como un administrador, has clic con el botón derecho en "Mi PC" y

selecciona "Administrar". Ve a "Usuarios locales y grupos", has clic con el botón derecho en Usuarios y selecciona "nuevo usuario".



Ingresa un nombre de usuario (en este caso elegí "usuariossh") y una buena contraseña. Desmarca la casilla de "El usuario debe cambiar la contraseña en el

siguiente inicio de sesión" y marca las casillas de "El usuario no puede cambiar la contraseña" y "La contraseña nunca caduca". Clic en "crear". Listo, puedes cerrar la

Administración de Equipos.

Usuario nuevo ? X

Nombre de usuario:

Nombre completo:

Descripción:

Contraseña:

Confirmar contraseña:

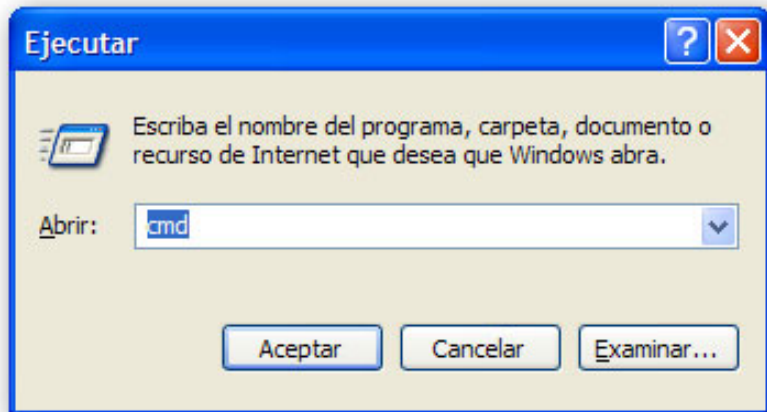
El usuario debe cambiar la contraseña en el siguiente inicio de sesión

El usuario no puede cambiar la contraseña

La contraseña nunca caduca

Cuenta deshabilitada

Ve al menú Inicio, Ejecutar, escribe cmd, aceptar.

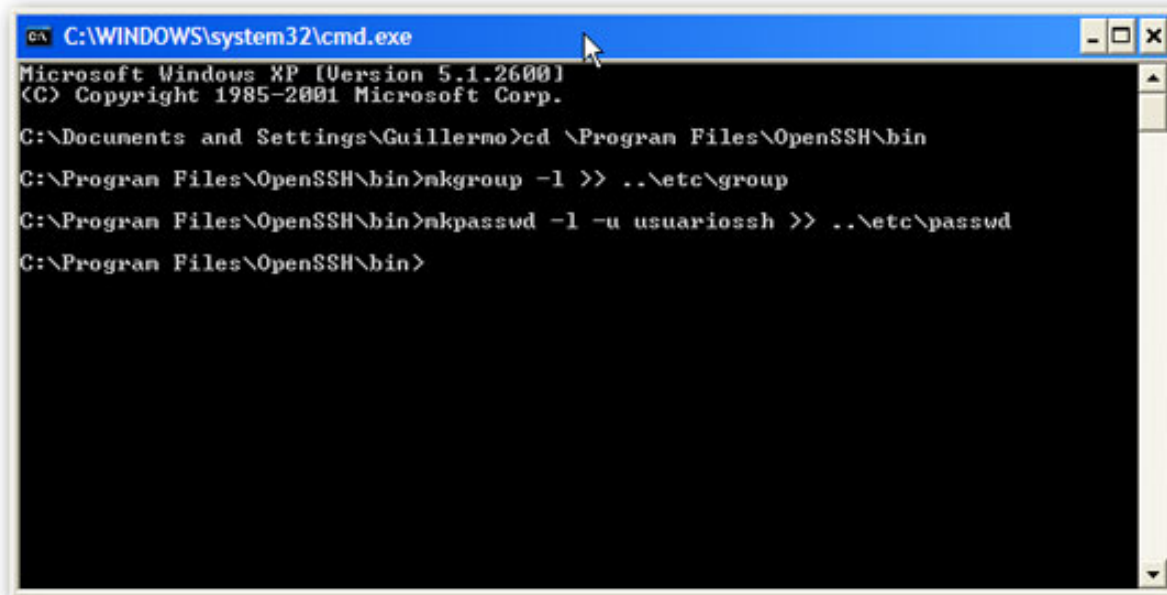


Escribe lo siguiente:

```
cd \Archivos de Programa\OpenSSH\bin
```

```
mkgroup -l >> ..\etc\group
```

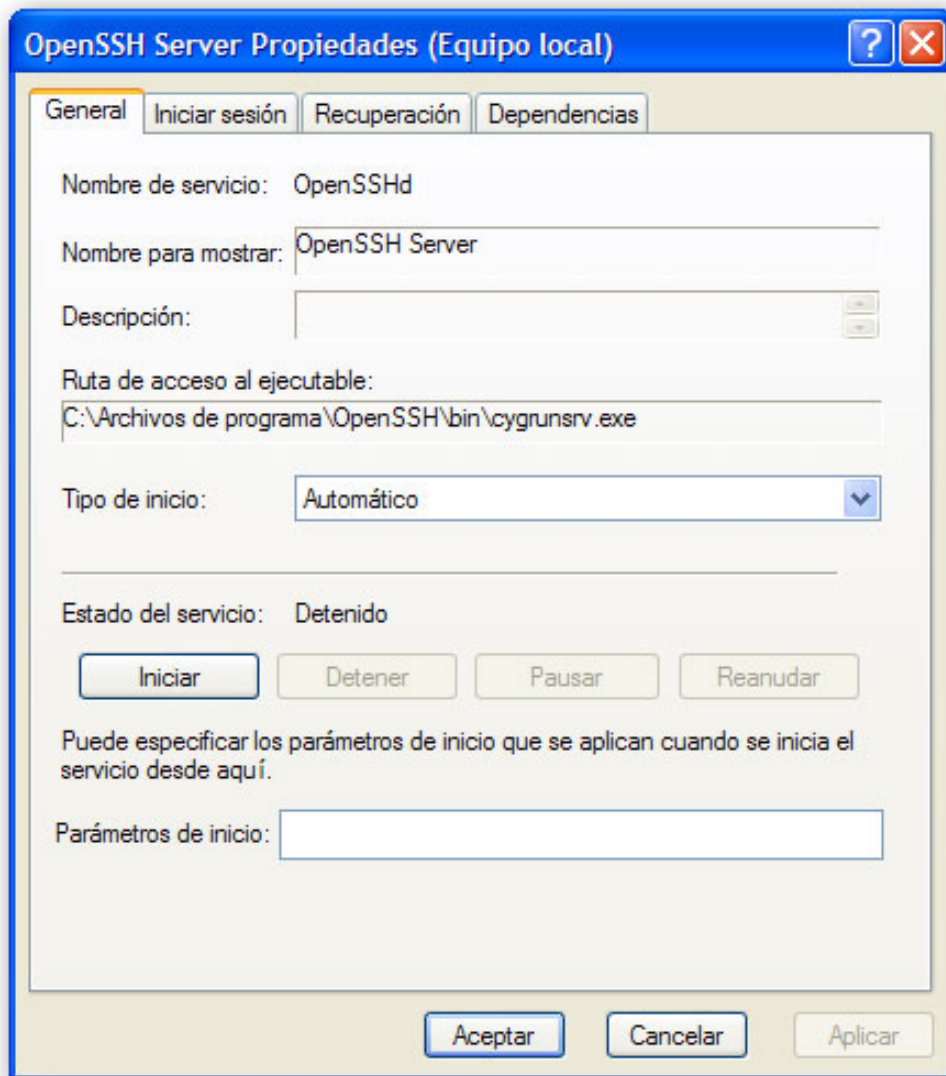
```
mkpasswd -l -u usuariossh >> ..\etc\passwd
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Guillermo>cd \Program Files\OpenSSH\bin
C:\Program Files\OpenSSH\bin>mkgroup -l >> ..\etc\group
C:\Program Files\OpenSSH\bin>mkpasswd -l -u usuariossh >> ..\etc\passwd
C:\Program Files\OpenSSH\bin>
```

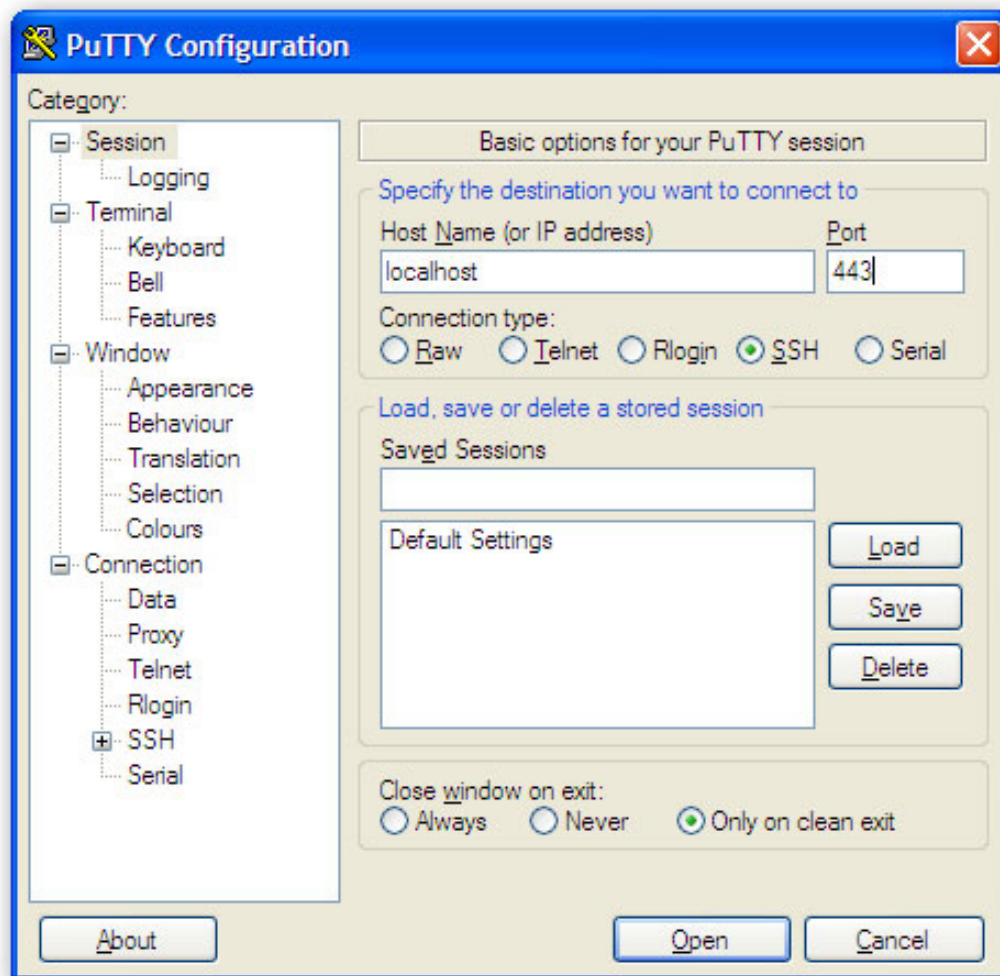
Seguidamente, ve al Panel de Control, Herramientas Administrativas, Servicios. Busca "OpenSSH server" en la lista de servicios y has doble-clic para abrir sus propiedades. Asegúrate que el tipo de inicio sea "automático", y presiona el botón de "Iniciar".



Excelente, ya tu PC de casa es un servidor SSH. Sólo queda un par de cosas que hacer, que es abrir el puerto correspondiente (el que configuraste del SSH) en el firewall de la PC y en el router.

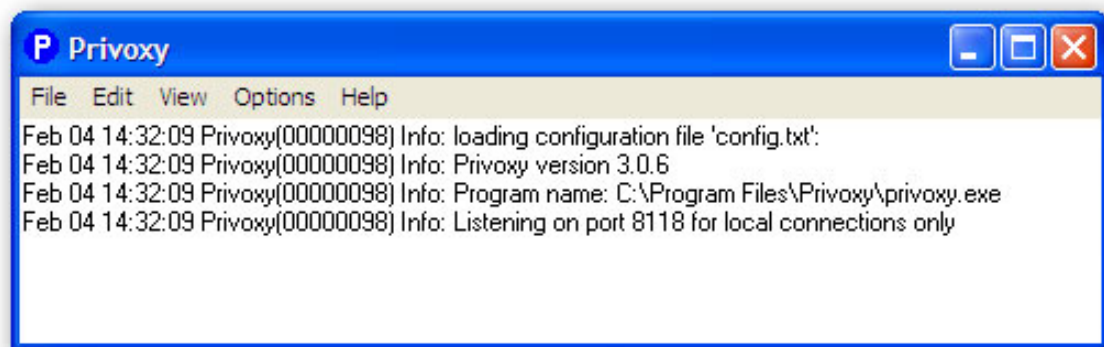
¿Qué tal si lo probamos? Descarga PuTTY (un cliente gratuito de SSH para Windows), y ejecútalo en la PC1 (no requiere instalación). En "host name", coloca

localhost (porque estamos en casa todavía), y en "port" el puerto que seleccionaste para el SSH, en mi caso 443. Presiona el botón de Open. Debería salir una advertencia que sólo sale la primera vez, y después debería pedirte el nombre de usuario y contraseña. Usa los que escogiste para SSH. ¡Woohoo! Deberías ver una consola parecida al command prompt de Windows.



Instalando Privoxy

Ok, sigamos, ya la parte difícil está hecha. Es hora de instalar el servidor proxy en la PC1, y para esto vamos a usar [Privoxy](#). Descarga la versión más reciente e instálala, no te preocupes por las opciones, ya que las que están por defecto funcionan bien. Cuando termines de instalarlo, tendrás un proxy HTTP que sólo escucha conexiones locales en el puerto 8118. Ya veremos por qué esto es útil.



Configurando el túnel SSH

Finalmente, solo nos queda crear el túnel SSH propiamente dicho. En la PC2, es decir, en la oficina/escuela/universidad, vamos a ejecutar PuTTY. Como no necesita instalación, simplemente podemos llevarlo en un flash drive o hasta en un diskette y ejecutarlo desde ahí. En fin, coloca la información para conectarte a tu servidor en casa. Esta vez no será `localhost`, sino el número de IP de casa. Lo mejor en este caso es instalar (en la PC1) un servicio de DNS dinámico como [No-IP](#), para no tener que recordar el número cada vez que nos queramos conectar.

Specify the destination you want to connect to

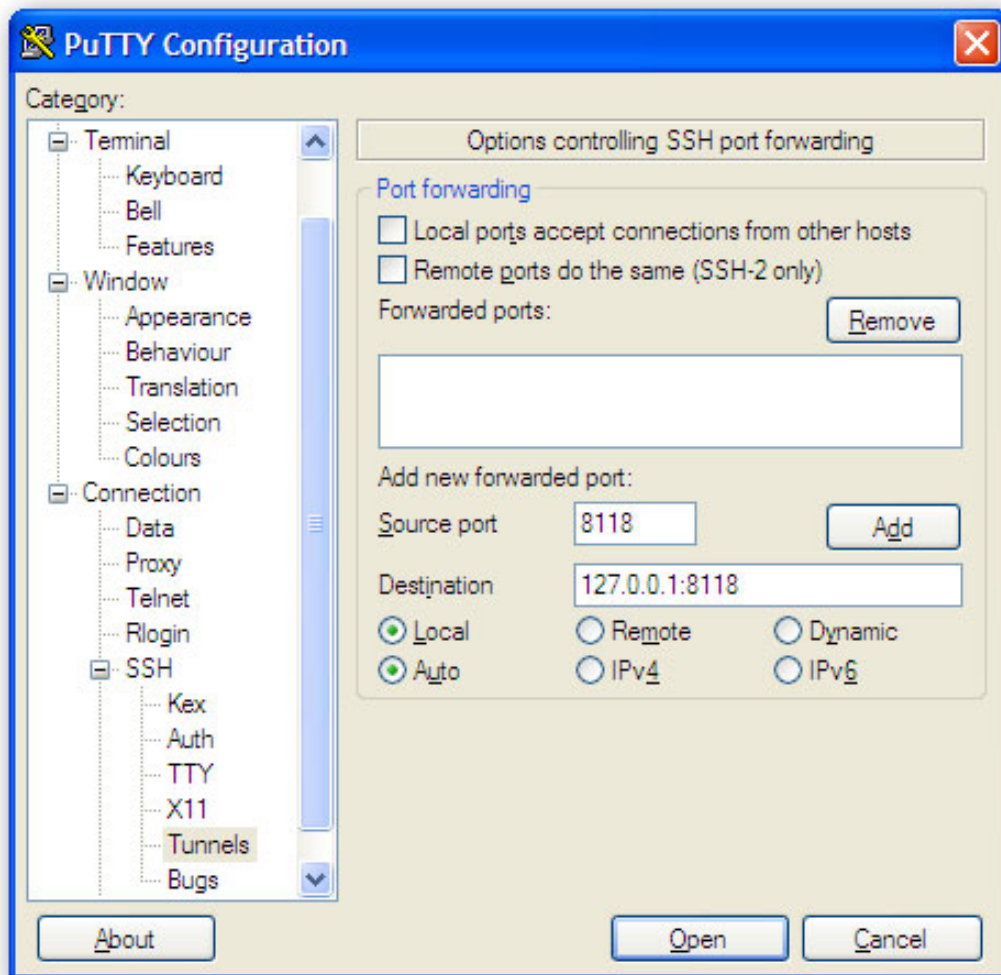
Host Name (or IP address)	Port
<input type="text" value="████████████████████"/>	<input type="text" value="443"/>

Connection type:

Raw Telnet Rlogin SSH Serial

Antes de conectarte, en el menú de la izquierda selecciona Connection, SSH, Tunnels. En "Source port", ingresa 8118, y en "Destination", ingresa 127.0.0.1:8118.

Presiona "Add".



¿Qué es todo esto? Sin profundizar en detalles técnicos, SSH permite redireccionar puertos entre el cliente (la PC2) y el servidor (PC1, donde instalamos OpenSSH y Privoxy). El tráfico que entra en el puerto "Local" es enviado al servidor SSH, y de ahí es enviado a la dirección indicada en el campo "Destination". Y como muchos sabrán, 127.0.0.1 es una dirección comúnmente llamada "localhost" que es usada por una computadora para referirse a sí misma. En cristiano, lo que estamos haciendo es agarrar todo el tráfico que entra a nuestro cliente por el puerto 8118, mandarlo a través del túnel SSH al servidor en casa, el cual procede a enviarlo a

Entonces, es abrir a sí misma en el puerto 8118. La cada ventana siendo una conexión local en el puerto 8118, que es lo único que necesita el proxy.

Ok, ahora vuelve a la ventana inicial de PuTTY (seleccionando Session en la parte superior del menú izquierdo), verifica los datos del host y el puerto. Escribe algo

en Saved Sessions y has clic en Save, de esa manera no tienes que volver a configurar el túnel cada vez que quieras usarlo. Clic en Open para iniciar la sesión.

Obviamente, debes ingresar el nombre de usuario y contraseña y mantener PuTTY abierto todo el tiempo que quieras usar el túnel, si no el agujero de seguridad

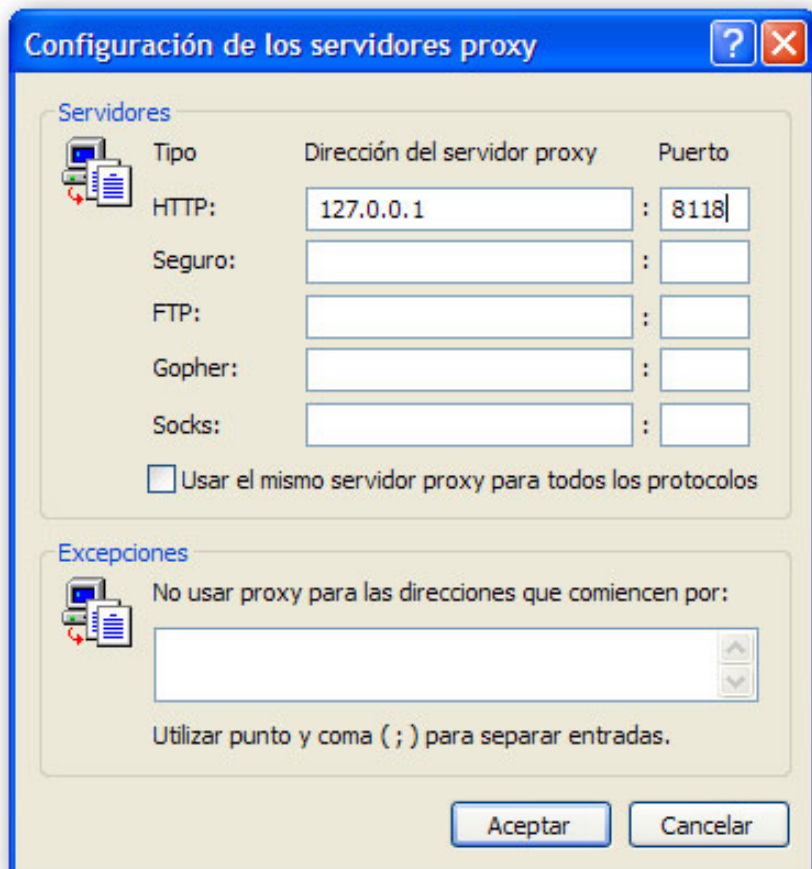
sería del tamaño del mundo.

Configurando el navegador para usar nuestro proxy

Bueno, ya que tenemos nuestro túnel SSH funcionando, lo único que nos queda es configurar el navegador en PC2 para usar el proxy HTTP. En Internet Explorer 6,

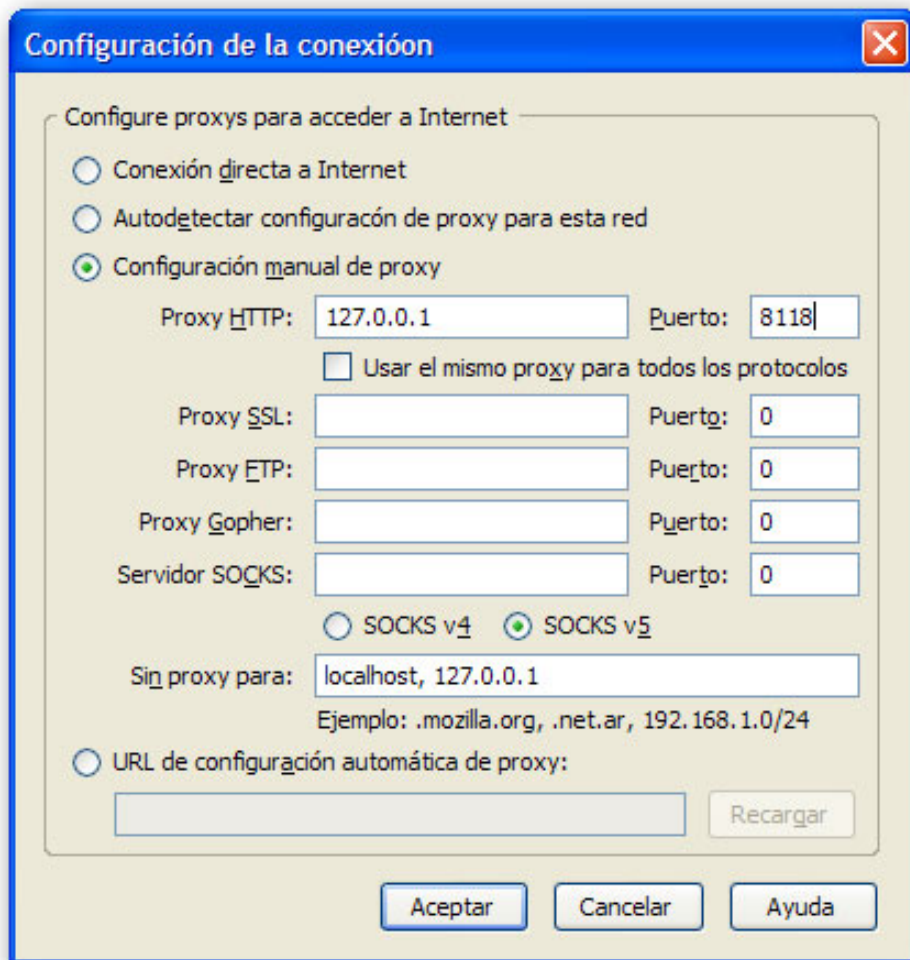
debes ir a Herramientas, Opciones de Internet, Conexiones, Configuración de LAN, Avanzadas. Coloca en HTTP, Dirección del servidor proxy: 127 . 0 . 0 . 1,

puerto: 8118.



En Firefox, debes ir a Herramientas, Opciones, Avanzadas, Red, Opciones, Configuración manual de proxy, y en proxy HTTP: 127 . 0 . 0 . 1, puerto: 8118.

También tienes la opción de instalar la extensión [SwitchProxy](#) para seleccionar entre varias configuraciones de proxy rápidamente.



Si no tienes permiso en tu PC para modificar la configuración de proxy de Internet Explorer, puedes usar [Portable Firefox](#). Este navegador puedes guardarlo en un flash drive y ejecutarlo directamente desde ahí.

Y listo, ya puedes navegar.

Observaciones y conclusiones

Algunas observaciones finales: En primer lugar, este método no es precisamente rápido, puesto que la mayoría de las conexiones caseras son asimétricas. Es decir, la velocidad de bajada es mayor que la velocidad de subida. Normalmente esto no es un problema ya que descargar una página web no es tan pesado como realizar la solicitud a dicha página, pero en este caso cada página que es descargada es subida de vuelta por el túnel SSH. Así que los mejores resultados se logran con una conexión que tenga buena velocidad de subida, pero aún así la diferencia de velocidad se notará. Pero es mejor que nada, supongo.

Segundo, a pesar de que este método es bastante seguro ya que el contenido del túnel está encriptado, igual recomiendo usarlo únicamente para fines legítimos, como revisar algún correo personal (muchas veces me ha pasado que necesito algo que tengo en Gmail para lo que estoy haciendo en ese momento y por mala suerte está bloqueado). De esa forma, si algún superior pregunta la razón de esa conexión desde tu terminal, por lo menos tienes una excusa válida. [Si en verdad](#) piensas usar esto para bajar porno en el trabajo, recuerda que aún queda un registro de tu navegación en la PC (historial, caché, etc.), y ten en cuenta que cuando te boten no tienes más nadie a quien culpar sino a ti mismo.

Finalmente, si en tu trabajo, universidad o escuela no bloquean el acceso a Internet no significa que no puedas usar este método. Si estás en un sitio público, más bien creo que tienes todas las razones del mundo para usar un túnel encriptado para navegar en Internet con mayor privacidad. Nunca sabes quien puede estar usando [Ethereal](#) para espiar tu tráfico web.