# Servicios de acceso remoto I: Telnet.

**Autor: Enrique V. Bonet Esteban**

## *Introducción.*

El servicio de telnet permite que un usuario se conecte desde su ordenador a otro ordenador con el fin de utilizar los recursos disponibles en el mismo. Esto, que inicialmente parece muy sencillo, debió enfrentarse al problema de que, durante mucho tiempo, los fabricantes de ordenadores vendían un entorno completamente propietario, de forma que sólo se podía acceder a una aplicación en el ordenador del fabricante desde terminales del mismo fabricante. Este hecho planteaba problemas de compatibilidad en el momento en que apareció la red y con ello la posibilidad de conectarse entre ordenadores de distintos fabricantes.

Para salvar esas diferencias entre los terminales de los fabricantes y permitir utilizar un ordenador desde cualquier otro se desarrollo un programa, que simula ser una terminal en red, conocido como telnet (TErminaL NETworking), siendo la emulación de terminal que proporciona telnet la primera aplicación desarrollada sobre TCP/IP. Además, telnet se diseñó como base para las comunicaciones entre cualquiera dos aplicaciones, de forma que telnet es el soporte de las interacciones cliente/servidor en la transferencia de archivos (FTP), el correo electrónico (SMTP), la World Wide Web (WWW), etc.

## *Negociación de la emulación del terminal. Protocolo NVT.*

Telnet fue desarrollado para trabajar con cualquier tipo de terminal propietaria existente y, aunque desde la primera versión de telnet se han añadido muchas emulaciones de terminales distintas, estas pueden seguir clasificándose en dos grandes conjuntos:

- Terminales ASCII[1], utilizados por los sistemas operativos Linux/UNIX y VAX, y cuyas características principales son el uso de ASCII de 8 bits, el eco remoto de cada carácter enviado, la transmisión dúplex y la posibilidad de soportar aplicaciones interactivas en pantalla completa.

- Terminales IBM 3270 y 5250, utilizados por ordenadores IBM[2] y cuyas características principales son el uso de caracteres EBCDIC de 8 bits, transmisión en modo semidúplex y envío de datos en modo de bloque, esto es, todos los datos de la pantalla son transmitidos a la vez.

Sin embargo, para poder iniciar una conexión entre un cliente y un servidor de telnet, ambos deben ponerse de acuerdo en que modelo de terminal van a emular, pero al existir múltiples modelos de terminal, la comunicación debe comenzar mediante un

[1] Sus características básicas se especifican en las normas ANSI X.3.64, ISO 6429 e ISO 2022, siendo los más comunes los conocidos como ANSI, VT52, VT100, VT220, TVI950, TVI955 y WYSE50.
[2] Los terminales IBM 3270 son los utilizados por la mayoría de equipos de gama alta de IBM, mientras que los IBM 5250 son los utilizados por las computadoras AS/400 de gama media.

modelo de terminal común preestablecido. Este modelo de terminal común es conocido como terminal virtual de red (Network Virtual Terminal).

NVT se desarrolló utilizando como modelo un teclado semidúplex y una impresora funcionando en modo línea a línea, lo cual confiere a NVT unas características bien definidas, como son:

- Datos formados por caracteres USASCII de 7 bits aumentados a 8 bits mediante la inclusión de un bit inicial de valor 0. De esos 128 caracteres posibles, los comprendidos entre los códigos 32 al 126 ambos inclusive corresponden a letras, números y símbolos y signos de puntuación imprimibles, mientras que los restantes caracteres permiten controlar la presentación en la pantalla del cliente.

- El protocolo es semidúplex, enviando el cliente al servidor los datos línea a línea, terminando todas las líneas mediante los caracteres retorno de carro (Carriage Return) y salto de línea (LineFeed). Después de enviar una línea, el cliente cede el control al servidor, el cual elabora la respuesta a enviar y empieza a enviarla línea a línea con el mismo formato y, para indicar que ya ha terminado de enviar líneas, envía un comando de "Adelante" (Go Ahead), devolviendo la comunicación al cliente.

- Los bytes cuyo bit inicial (bit más significativo) es 1 se usan para códigos de comandos. Los códigos de comandos se utilizan para establecer una negociación entre el cliente y el servidor, de forma que estos puedan establecer las características de emulación de terminal con las que van a trabajar.

La negociación de la terminal a emular se establece mediante el intercambio de comandos, siendo los principales comandos los siguientes:

| Petición de negociación | Valor hexadecimal |
|---|---:|
| WILL | FB |
| WON'T | FC |
| DO | FD |
| DON'T | FE |
| SB (Inicio de subnegociación) | FA |
| SE (Fin de subnegociación) | F0 |

Cualquiera de los extremos puede pedir a su corresponsal que realice (DO) una opción en particular. El otro extremo puede aceptar o rehusar. Cualquiera de los extremos puede ofrecerse para realizar (WILL) alguna opción. De nuevo, el otro extremo puede aceptar o no[3].

Una vez el cliente y el servidor han negociado el tipo de terminal a emular, la conexión continúa con la características de ese terminal. Sin embargo, si ambas partes rechazan todas las posibilidades de opciones en la negociación, la sesión se mantendrá abierta, funcionando con las características de terminal de NVT.

[3] Una información más detallada sobre la negociación que se produce se encuentra en el apéndice A de este tema.

## Envío de comandos.

Uno de los problemas que debe resolver un cliente de telnet es poder enviar determinadas combinaciones de teclas que fuerzan alguna acción sobre el servidor. Así, por ejemplo, Ctrl-C indica al sistema operativo que la aplicación que se esta ejecutando en ese terminal debe finalizar.

Estas acciones, que requieren una atención inmediata del sistema operativo del servidor, son conocidas como comandos. Los comandos de telnet se representan con un byte, llamado "interpretar como comando" (Interpret As Command), seguido de uno o más bytes de código. El byte IAC tiene como valor 0xFF[4]. Las secuencias de comandos permiten al cliente de telnet ejecutar funciones en el servidor como:

| Acrónimo | Comando | Código |
|---|---|---|
| EOF | End Of File (Fin de archivo) | 0xEC |
| SUSP | Suspend Current Process (Suspender proceso en curso) | 0xED |
| ABORT | Abort Process (Abortar proceso) | 0xEE |
| EOR | End Of Record (Fin de registro) | 0xEF |
| NOP | No Operation (No operación) | 0xF1 |
| DM | Data Mark (Marca de datos) | 0xF2 |
| BRK | Break (Pausa) | 0xF3 |
| IP | Interrupt Process (Interrumpir proceso) | 0xF4 |
| AO | Abort Output (Abortar la salida) | 0xF5 |
| AYT | Are You There (¿Estás ahí?) | 0xF6 |
| EC | Erase Character (Borrar carácter) | 0xF7 |
| EL | Erase Line (Borrar línea) | 0xF8 |
| GA | Go Ahead (Adelante) | 0xF9 |

Los comandos son introducidos normalmente en el flujo de datos, pero en ciertas ocasiones, como puede ser el comando Interrupt Process, que debe ejecutarse inmediatamente, el cliente aprovecha la característica que posee TCP de poder etiquetar determinados segmentos de la transmisión como urgentes, para que sean procesados en primer lugar[5].

## *El cliente de telnet.*

El cliente telnet permite la emulación de diferentes tipos de terminal, por lo que se permite acceder a ordenadores con sistemas operativos Linux/UNIX, sistemas VAX/VMS o grandes ordenadores de IBM. Si se ejecuta telnet desde un sistema multiusuario, generalmente se manejará una sencilla interfaz de usuario de texto. El cliente de telnet, que se encuentra en */usr/bin/telnet*, se ejecuta como[6]:

```
> telnet [nombre del ordenador] [puerto]
```

[4] Si una terminal utiliza un modelo de terminal en el que el valor 0xFF se corresponde con un carácter valido, como por ejemplo en ASCII extendido, el carácter 0xFF enviado debe ser duplicado para informar al servidor de que no se trata de un comando sino de un carácter.

[5] Para ello utiliza el campo puntero de datos urgentes situados en los dos últimos bytes de la cabecera TCP cuando esta no lleva ninguna opción.

[6] A menudo, la emulación de las terminales IBM 3270 se suministra en un programa llamado tn3270, cuyo uso es similar al de telnet.

Un ejemplo de conexión desde nuestro ordenador a *glup.irobot.uv.es* es el siguiente:

```
> telnet glup.irobot.uv.es
Trying 147.156.222.65...
Connected to glup.irobot.uv.es (147.156.222.65).
Escape character is '^]'.
Fedora release 19 (Schrödinger's Cat)
Kernel 3.10.11-200.fc19.x86_64 on an x86_64 (1)
login: usuario
Password: **********
Last login: Wed Oct 10 13:56:45 from glup.irobot.uv.es
```

Donde, al no especificar el puerto, se toma por defecto el puerto 23.

El cliente de telnet puede ser ejecutado sin ningún parámetro, en cuyo caso el cliente permanece a la espera de un comando. En dicho estado, podemos obtener información de los comandos de nuestro cliente telnet tecleando "?", obteniendo como salida:

```
> telnet
telnet> ?
Commands may be abbreviated.  Commands are:

close        close current connection
logout       forcibly logout remote user and close the connection
display      display operating parameters
mode         try to enter line or character mode ('mode ?' for more)
open         connect to a site
quit         exit telnet
send         transmit special characters ('send ?' for more)
set          set operating parameters ('set ?' for more)
unset        unset operating parameters ('unset ?' for more)
status       print status information
toggle       toggle operating parameters ('toggle ?' for more)
slc          change state of special charaters ('slc ?' for more)
z            suspend telnet
!            invoke a subshell
environ      change environment variables ('environ ?' for more)
?            print help information
telnet>
```

Además de las opciones anteriores, en toda sesión de telnet existe la posibilidad de interaccionar con el cliente de telnet para modificar las características de la sesión activa, para ejecutar algún comando en nuestro ordenador local mediante una shell, etc. Para ello, todo cliente telnet tiene una secuencia de control de teclado[7], que se indica en los mensajes de conexión (Escape character is...) y que generalmente es Ctrl-], que permite la salida al modo de comando del cliente de telnet. Por ejemplo, en el ejemplo de conexión mostrado con anterioridad, la secuencia de control se indica en la línea:

```
Escape character is '^]'.
```

Así, si estamos en una sesión de telnet e introducimos la secuencia de control, podemos, por ejemplo, observar el estado de la sesión actual, tal y como muestra el siguiente ejemplo:

[7] La secuencia de control de teclado puede ser definida por el usuario.

```
telnet> status
Connected to glup.irobot.uv.es (147.156.222.65).
Operating in single character mode
Catching signals locally
Remote character echo
Local flow control
Escape character is '^]'.
```

Pudiendo volver a la conexión pulsando “Enter”.

## Acceso a un puerto concreto mediante un cliente de telnet.

Como hemos visto, un cliente de telnet si no se especifica un puerto, establece una conexión con el puerto TCP 23, donde se encuentra en escucha el servidor de telnet, obteniéndose como respuesta un cursor para introducir el usuario y la contraseña.

Sin embargo, y dado que telnet se diseño como herramienta general de comunicaciones entre aplicaciones, incluye la posibilidad de conectar el cliente a cualquier puerto. Para ello, debemos añadir detrás del nombre del ordenador el puerto TCP al que queremos conectarnos. Por ejemplo, en el diálogo que aparece a continuación se produce una conexión a un servidor de entrega final de correo que funciona en el puerto 110.

```
> telnet post.uv.es 110
Trying 147.156.0.253...
Connected to post.uv.es (147.156.0.253).
Escape character is '^]'.
+OK post2.uv.es Cyrus POP3 Murder v2.3.16 server ready
QUIT
+OK
Connection closed by foreign host.
```

La posibilidad de acceder con telnet a cualquier puerto ha demostrado ser muy útil, pues permite comprobar el funcionamiento de la mayoría de servicios cliente/servidor implementados en Internet. Sin embargo, esa utilidad se ha convertido en una potencial fuente de problemas de seguridad cuando algún usuario lo utiliza para, a través de un programa servidor deficiente que se ejecuta en algún puerto sin restricciones, acceder a un ordenador remoto sin tener permisos para ello.

## *El servidor de telnet.*

El servidor de telnet es un programa que se encuentra en */usr/sbin/in.telnetd* y que soporta la emulación del estándar NVT, así como otros tipos de terminales.

La forma clásica de ejecutar un servidor de telnet es que este se ejecutará en el arranque y se encontrará a la escucha del puerto TCP 23, esperando solicitudes de conexión al citado puerto. Sin embargo en ordenadores con sistema operativo Linux, el servidor de telnet es lanzado por el servidor xinetd[8].

[8] En temas posteriores se analizará el funcionamiento y la configuración del servidor xinetd.

Sea lanzado de una u otra forma, el servidor de telnet atiende las conexiones solicitadas por los clientes de telnet, solicitando el nombre del usuario y su clave de acceso en el ordenador, permitiendo, una vez ha sido verificada su identidad, el acceso a los recursos de dicho usuario como si se encontrara en una terminal conectada físicamente al ordenador.

Sin embargo, y por motivos de seguridad, la mayoría de implementaciones de telnet no permiten al usuario "root" acceder al ordenador mediante telnet, excepto para un conjunto concreto de terminales, conocidas como "terminales seguras", cuyo listado se realiza en el fichero */etc/securetty*. Un ejemplo de dicho fichero es el siguiente:

```
console
vc/1
...
vc/11
tty1
...
tty11
hvc0
hvc1
hvsi0
hvsi1
hsvi2
xvc0
```

En dicho ejemplo, se puede comprobar como el acceso de root solo esta permitido para clientes de telnet que se ejecutan en terminales conectadas directamente al ordenador, no existiendo ninguna terminal remota (pts/0, pts/1, etc.) desde la que el usuario root tenga acceso.

El funcionamiento del servidor de telnet es relativamente sencillo. En primer lugar, el servidor de telnet negocia con el cliente el tipo de terminal a emular, mostrando a continuación el contenido del fichero */etc/issue.net*, que es un mensaje de acceso al sistema. Una vez mostrado ese mensaje, el servidor de telnet ejecuta el programa[9] */bin/login*, el cual solicita el usuario y la contraseña de acceso al sistema, muestra el contenido del fichero */etc/motd*, que es el mensaje del día (Message Of The Day) y que suele contener avisos a los usuarios del sistema, y ejecuta por último la shell que se encuentra predeterminada para el usuario.

Mientras el contenido del fichero */etc/motd* es un simple texto, el contenido del fichero */etc/issue.net* puede contener secuencias de caracteres que modifican el mensaje a mostrar. Estas secuencias, que están formadas por el carácter \ (ó el carácter %) seguido de una letra, son:

| Secuencia | Descripción |
|---|---|
| \l | Muestra el identificador de la terminal. |
| \h ó \n | Muestra el nombre del ordenador. |
| \D ó \o | Muestra el nombre del dominio NIS. |
| \d ó \t | Muestra el día y hora del sistema |
| \s | Muestra el nombre del sistema operativo. |

[9] Existe la posibilidad de modificar el programa que ejecuta el servidor de telnet mediante la opción –L.

| Secuencia | Descripción |
|---|---|
| \m | Muestra el tipo de hardware del ordenador. |
| \r | Muestra la revisión del sistema operativo. |
| \v | Muestra la versión del sistema operativo. |
| \\ | Muestra el símbolo %. |

Un ejemplo del fichero */etc/issue.net* es el siguiente:

```
Fedora release 19 (Schrödinger's Cat)
Kernel \r on an \m (\l)
```

## *Apéndice A: Negociación de emulación de terminal en NVT.*

Las características de emulación de terminal, una vez establecida la conexión mediante NVT, se establecen intercambiando comandos que negocian opciones telnet, que como vimos son:

| Petición de negociación | Valor hexadecimal |
|---|---|
| WILL | FB |
| WON'T | FC |
| DO | FD |
| DON'T | FE |
| SB (Inicio de subnegociación) | FA |
| SE (Fin de subnegociación) | F0 |

Cualquiera de los extremos puede pedir a su corresponsal que realice (DO) una opción en particular, como por ejemplo, "hacer eco de caracteres individuales". El otro extremo puede aceptar o rehusar. Cualquiera de los extremos puede ofrecerse para realizar (WILL) alguna opción. De nuevo, el otro extremo puede aceptar o no. Esta característica de funcionamiento establece la existencia de cinco intercambios básicos de petición/respuesta que suelen ocurrir durante la negociación de opciones:

| Opción | Descripción | Respuesta | Descripción |
|---|---|---|---|
| DO | Pide que se acepte una opción. | WILL | La opción es aceptada y ejecutada. |
| DO | Pide que se acepte una opción. | WON'T | La opción es rechazada y el estado no cambia. |
| WILL | Indica la posibilidad de ejecutar una opción. | DO | La posibilidad es aceptada y ejecutada. |
| WILL | Indica la posibilidad de ejecutar una opción. | DON'T | La posibilidad es rechazada en el instante actual y el estado no cambia. |
| WILL | Indica la posibilidad de ejecutar una opción. | WON'T | La posibilidad es rechazada para siempre y el estado no cambia. |

Al inicio de una conexión, existen un gran número de peticiones de opciones que son intercambiadas entre el cliente y el servidor. Algunas de estas opciones señalan el comienzo de subnegociaciones, en las cuales se intercambia información adicional. Un ejemplo de subnegociación es la que se establece cuando el cliente envía el comando WILL TERMINAL TYPE, indicando al servidor que esta dispuesto a mandarle los tipos de terminal que puede emular, y el servidor acepta que se le envíe ahora dicha información mediante DO TERMINAL TYPE. En ese momento, se establece una subnegociación, en la que el servidor pide al cliente que indique uno de los tipos de terminal que puede emular y el cliente responde. El servidor puede repetir la petición

hasta que, o bien el cliente proporciona un tipo que es aceptable para el servidor, o bien la lista de tipos disponibles por el cliente se termina.

La peticiones de opciones de negociación y subnegociación se codifican con tres bytes: un código de IAC (valor hexadecimal 0xFF), un octeto de petición y un código de opciones. Por ejemplo, la representación de la secuencia para WILL TERMINAL TYPE es: 0xFF 0xFB 0x18. En la tabla siguiente se muestran los números de código asociados a algunas de las opciones más utilizadas.

| Códigos de opción del comando | Valor hexadecimal |
|---|---|
| Transmit Binary (Transmitir en binario) | 00 |
| Echo (Eco) | 01 |
| Suppress Go Ahead (Suprime Adelante) | 03 |
| Status (Estado) | 05 |
| Timing Mark (Marca de tiempo) | 06 |
| Output Line Width (Ancho de línea de salida) | 08 |
| Output Page Size (Tamaño de página de salida) | 09 |
| Extended ASCII (ASCII extendido) | 11 |
| Data Entry Terminal (Terminal de entrada de datos) | 14 |
| Terminal Type (Tipo de terminal) | 18 |
| End of Record (Fin de registro) | 19 |
| Window Size (Tamaño de ventana) | 1F |
| Terminal Speed (Velocidad del terminal) | 20 |
| Remote Flow Control (Control de flujo remoto) | 21 |
| Linemode (Modo de línea) | 22 |
| Authentication (Autenticación) | 25 |
| Encryption (Cifrado) | 26 |
| Extended Options List (Lista extendida de opciones) | FF |

Se han escrito más de 30 RFC que detallan opciones para definir características especiales, incluyendo opciones como la capacidad de pedir al otro extremo los parámetros de la opción actual, la negociación del tamaño de la ventana donde se ejecuta la sesión de telnet, etc.

En el diálogo de ejemplo que aparece a continuación, se ejecuta telnet y se introduce “toggle options” para hacer que telnet nos muestre sus negociaciones. Entonces se usa “open” para iniciar una conexión. Los interlocutores negocian una emulación de una terminal ASCII de tipo XTERM seleccionando algunas características como:

- El servidor no enviará adelante (Go Ahead) porque la sesión será dúplex.
- Se utiliza una subnegociación del tipo de terminal, velocidad de transferencia, para indicar el tipo concreto de terminal ASCII a emular.
- El servidor hará eco de los caracteres del cliente.

Ninguna de las partes necesita esperar una respuesta a una petición de opción antes de enviar otra petición. Ni siquiera un negociador tiene que responder a las

opciones en el mismo orden en que las recibió. Como resultado, a veces para entender una serie de negociaciones hay que desenredarla[10].

```
> telnet
telnet> toggle options
Will show option processing.
telnet> open glup.irobot.uv.es
Trying 147.156.222.65...
Connected to glup.irobot.uv.es (147.156.222.65).
Escape character is '^]'.
SENT DO SUPPRESS GO AHEAD
SENT WILL TERMINAL TYPE
SENT WILL NAWS
SENT WILL TSPEED
SENT WILL LFLOW
SENT WILL LINEMODE
SENT WILL NEW-ENVIRON
SENT DO STATUS
SENT WILL XDISPLOC
RCVD DO TERMINAL TYPE
RCVD DO TSPEED
RCVD DO XDISPLOC
RCVD DO NEW_ENVIRON
RCVD WILL SUPPRESS GO AHEAD
RCVD DO NAWS
SENT IAC SB NAWS 0 80 (80) 0 24 (24)
RCVD DO LFLOW
RCVD DONT LINEMODE
RCVD WILL STATUS
RCVD IAC SB TERMINAL-SPEED SEND
SENT IAC SB TERMINAL-SPEED IS 38400,38400
RCVD IAC SB X-DISPLAY-LOCATION SEND
SENT IAC SB X-DISPLAY-LOCATION IS "amparo:0"
RCVD IAC SB NEW-ENVIRON SEND
SENT IAC SB NEW-ENVIRON IS VAR "DISPLAY" VALUE "amparo:0"
RCVD IAC SB TERMINAL-TYPE SEND
SENT IAC SB TERMINAL-TYPE IS "XTERM"
RCVD DO ECHO
SENT WONT ECHO
RCVD WILL ECHO
SENT DO ECHO
Fedora release 19 (Schrödinger's Cat)
Kernel 3.10.11-200.fc19.x86_64 on an x86_64 (1)
login:
```

[10] Una explicación mucha más detallada de las opciones de negociación puede encontrarse en las páginas de manual de telnet y telnetd.