**Aplicación AUR**

López Sebastian, Ocanto Pablo 2, Bogarin Horacio3

1 Universidad Tecnológica Nacional, Facultad Regional La Plata, [sebastianlopezz90@gmail.com](mailto:%20sebastianlopezz90@gmail.com)

2 Universidad Tecnológica Nacional, Facultad Regional La Plata, [pomalianni@gmail.com](mailto:pomalianni@gmail.com)

3 Universidad Tecnológica Nacional, Facultad Regional La Plata, [bogarin1983@gmail.com](mailto:bogarin1983@gmail.com)

**Abstract.** El presente documento trata sobre el desarrollo de una aplicación de ejecución remota de comandos a través del protocolo Telnet. Este protocolo permite viajar a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. Para que la conexión funcione la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones.

**Keywords.** C, Sockets, Comandos remotos, administración, cliente-Servidor

1 Motivación del proyecto

El protocolo Telnet es un protocolo estándar que permite conectar terminales y aplicaciones entre sí. Proporciona reglas básicas que permiten vincular a un cliente con un intérprete de comandos (servidor).

El acceso a otra computadora se realiza en modo terminal (sin ningún tipo de interfaz gráfica) y permite solucionar fallos a distancia, de modo tal que se pueda tratar el problema sin necesidad de estar físicamente junto al equipo en cuestión. Telnet también posibilita la consulta remota de datos o el inicio de una sesión con una máquina UNIX (en este caso, múltiples usuarios pueden abrir sesión de manera simultánea y trabajar con la misma computadora).

El protocolo Telnet se aplica en una conexión TCP para enviar datos en formato ASCII codificados en 8 bits, entre los cuales se encuentran secuencias de verificación Telnet.

Por lo tanto, brinda un sistema de comunicación orientado bidireccional (semidúplex) codificado en 8 bits y fácil de implementar.

Una conexión mediante Telnet permite acceder a cualquiera de los servicios que la máquina remota ofrezca a sus terminales locales. De esta manera se puede abrir una sesión o acceder a otros servicios especiales. Normalmente para ello sólo debemos conocer el nombre de la máquina remota y tener cuenta en ella.

La comunicación entre cliente y servidor se maneja con órdenes internas, que no son accesibles por los usuarios. Todas las órdenes internas de Telnet consisten en secuencias de 2 ó 3 bytes, dependiendo del tipo de orden.

Debido a lo comentado anteriormente podemos implementar nuestra aplicación de ejecución de comandos remota a través del uso de este protocolo.

2 Estado del arte

Telnet fue el primer protocolo de aplicación demostrado en la ARPAnet, en 1969. La primera RFC de Telnet en ser definida específicamente fue la RFC 97. El desarrollo de Telnet continuó durante toda la década de 1970, con un buen número de diferentes RFC que se dedicaron a la revisión del protocolo y a las discusiones de los temas relacionados con ella. Tomó muchos años para perfeccionar Telnet y resolver todas las dificultades que se asociaron con su desarrollo. [1]

La versión final del protocolo Telnet fue publicado como RFC 854 en mayo de 1983. A través de los años otros RFC se han publicado para aclarar el uso del protocolo y abordar diversas cuestiones tales como la autenticación. También hay una serie de otros RFC que definen las opciones de Telnet.

2.1 Telnet en la Actualidad

Hoy en día este protocolo también se usa para acceder a los BBS, que inicialmente eran accesibles únicamente con un módem a través de la línea telefónica. Para acceder a un BBS mediante telnet es necesario un cliente que dé soporte a gráficos ANSI y protocolos de transferencia de ficheros. Los gráficos ANSI son muy usados entre los BBS. Con los protocolos de transferencia de ficheros (el más común y el que mejor funciona es el ZModem) podrás enviar y recibir ficheros del BBS, ya sean programas o juegos o ya sea el correo del BBS (correo local, de FidoNet u otras redes). [2]

3 Tecnologías

La aplicación de administración remota de usuarios fue desarrollada en lenguaje C, usando la librería de BSD Sockets, y las librerías de arpa/inet.h. Dicho servidor se montó en una máquina virtual conectada a un enlace con IP fija que permitió que todos los miembros del equipo se conecten y trabajen en simultáneo.

Para armar el entorno de trabajo fue necesario consultar documentación de los

Programas y aplicar algunos conocimientos ya adquiridos previamente por los integrantes.

Para la codificación de la aplicación se acudió a la documentación de la cátedra, a foros y

distintas páginas de la web que con ejemplos nos ha facilitado la familiarización del lenguajes C y de las librerías utilizadas.

4 Desarrollo del proyecto

El objetivo de este proyecto es desarrollar una aplicación que permita enviar comandos a un servidor y estos sean interpretados y procesados, que se ejecute en cualquier distribución de Linux utilizando el lenguaje C y las librerías BSD Sokets.

Se desarrollaron dos aplicaciones una servidor y otra cliente, que cumple con una series

de pautas que les da la funcionalidad a cada una de las aplicaciones.

4.1 Funcionalidades

**Aplicación Servidor:**

Hace el rol principal, tiene un registro de usuarios, los cuales a través del cliente tienen que enviarle las credenciales al servidor para que este los valide y le permita ejecutar comandos. Luego recibe y ejecuta los comandos que les llegan y envía los resultados a los clientes

**Aplicación Cliente:**

Posee el rol de enviar datos al servidor, en primera instancia envía los datos de usuario y contraseña para que el servidor valide y luego envía los comandos y espera respuesta del servidor.

4.2 Flujo de ejecución inicial

La aplicación cliente se conecta con el servidor, si la conexión esta correcta este le envía los datos de usuarios y contraseña que previamente son ingresados por el usuario, el servidor recibe estos datos y valida en su base de usuarios si son válidos tanto el usuario como la contraseña, en caso de que sean válidos el servidor le envía el ok y el cliente va a tomar los datos del comando del cliente y se lo envía al servidor, este lo procesa y le envía la respuesta al cliente.

4.3 Configuración

Las aplicaciones proveen un menú para la conexión por lo tanto solamente a la hora de la llamada hay que hacerla de la siguiente manera

./Servidor

./Cliente localhost

5 Descripción del Desarrollo

5.1 Definición del protocolo

El protocolo está basado en un conjunto de mensajes de cabecera que anuncian las acciones a tomar y registros que se usan para la transmisión propia de los paquetes de datos.

**Registro CABECERA:**

Componentes del protocolo

* + - **Header:** Se emplea para marcar la longitud total del mensaje.
    - **Body:** Define el cuerpo del mensaje de transmisión.

**Los valores que puede trasmitir son los siguientes:**

* SOLICITUD\_SIGN\_IN: Indica el Pedido de solicitud al servidor para poder logearse algún usuario en particular.
* ACK\_SIGN\_IN: Confirma el pedido de login de una usuario.
* LOGGIN\_USER: Pide acceso al sistema mediante usuario y password.
* USER\_VALID: Confirma que el usuario tiene acceso al sistema.
* USER\_ERROR: Indica los Tipos de errores de login que pueden ser por usuario invalido, password inválido o los dos.
* SOLICITUD\_COMANDO: Envió de distintos tipos de comandos (en nuestro caso sería ls|pwd|cd ) hacia el servidor.
* RESPUESTA\_COMANDO: Muestra el tipo de comando y la ejecución del mismo.
* EXIT\_COMANDO: indica para Salir de la aplicación.

5.2 Definición de la lógica de la aplicación

**Lógica del Hilo Servidor**

Una vez iniciado el servidor, este permanece en espera de que los distintos cli entes de

Registren. El proceso de autenticación, consiste en el envió de datos del cliente al servidor y este los valida con un archivo de texto local en este que contiene los datos de usuario.

El servidor tiene que interpretar los mensajes y ejecutarlos, una vez realizado le envía a los clientes el resultado del mismo.

**Mensajes de la Aplicación Servidor:**

* **Server Iniciado**
* **Nuevo Cliente**
* **Enviar tus datos**
* **Usuario Loggeado con éxito**
* **No es un usuario Valido**

**Lógica del Hilo Cliente**

Los clientes una vez que se inician buscan conectarse con la dirección ip que le solicitan al usuario. Una vez que se realiza la conexión el cliente le solicita al usuario que cargue el mensaje a enviar lo envía y espera respuesta de parte del servidor.

**Mensajes de la Aplicación Cliente** Puede realizar las siguientes operaciones ejecutando los comandos:

<ls|pwd|cd> <dir>:

HELP Ayuda

SALIR - Sale del programa

Se ha presionado Ctrl-c

Debe ejecutar la dirección de ip

Error en connect

Server me quiero conectar

SALIR

Se ah desconectado del servidor

6 Resultados obtenidos

El principal logro que se considera es aprender a usar el leguaje C y sockets sobre Linux, la aplicación generada en este trabajo tiene aún múltiples mejoras que plantear que se fueron descubriendo a medida que se profundizó en las tecnologías y la experiencia de su uso.

7 Conclusiones

La intercomunicación de las redes y el surgimiento de internet, han dado un espacio propicio para la interacción y colaboración entre usuarios, cosa que hoy día ha tomado en las sociedades un rol protagonista. Gran cantidad de los temas que se contemplan gira en torno al uso de estas tecnologías. En este trabajo nos enfocamos en buscar una técnica de transmisión de comandos que sea

Indagar sobre una técnica de transmisión de datos entre distintos clientes y sortear los distintos escoyos que se presenten ante este desafío.

Del presente trabajo surgió un conjunto de aplicaciones que buscan facilita la

Transmisión de archivos entre usuario y nos permitió adentrarnos en las tecnologías y los métodos requeridos para lograrlo.

Referencias

[1 http://www.tcpipguide.com/free/t\_TelnetOverviewHistoryandStandards-3.htm

[2] https://es.wikipedia.org/wiki/Telnet#Telnet\_en\_la\_actualidad [3]<https://netbeans.org/kb/docs/ide/subversion.html> [4]<http://subversion.apache.org/quick-start> [5][http://www.lopst.com/2008/06/16/instalar-un-servidor-de-subversion-svn-en-ubuntu- o-debian/](http://www.lopst.com/2008/06/16/instalar-un-servidor-de-subversion-svn-en-ubuntu-o-debian/)

[6 ] <http://c.conclase.net/librerias>