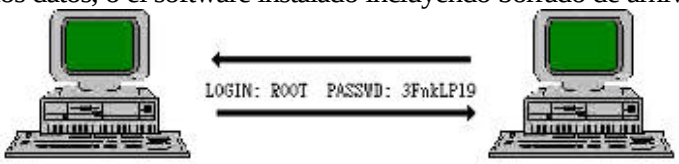
Manual de Uso Secure Shell
Departamento de Seguridad en Computo

¿Por qué SSH?

Cuando se realiza una conexión a un servidor remoto usando por ejemplo el comando **telnet** o **ftp**, el **login (usuario)** y **password (contraseña)** son transmitidos en la red de forma clara o plana, lo cual representa un gran riesgo si llega a existir sobre la red un programa que capture la información, basándose en el modo promiscuo de las redes ethernet (comúnmente llamado sniffer), con el cual se puede obtener el **login** y el **password** de un usuario legitimo del sistema, con lo que el atacante podría acceder al sistema y modificar los datos, o el software instalado incluyendo borrado de arhivos.



Este tipo de problemas ha llevado al diseño de herramientas que permitan evitar estas situaciones siendo el caso de **Secure Shell (ssh)**, desarrollado por **Tatu Ylonen** en la Universidad Tecnológica de Helsinki en Finlandia y **OpenSSH**, que nace del proyecto de un sistema operativo orientado con la filosofía de la seguridad en mente como lo es **OpenBSD**.

Secure Shell y **OpenSSH** permiten realizar la comunicación y transferencia de información de forma cifrada proporcionando fuerte autenticación sobre el medio inseguro. Este tipo de conexión se muestra en la ilustración siguiente:

El protocolo de **Secure Shell**, retomando lo anteriormente mencionado en resumen se pudieran dar las siguientes razones para su uso en la empresa o en una organización:

- ?? El protocolo transfiere datos de forma encriptada del punto emisor al receptor.
- ?? El protocolo dispone de múltiples algoritmos para encriptar la información.
- ?? El transmitir información encriptada de un punto a otro la hace inmune a ser espiada.
- ?? El protocolo usa llaves públicas que cambia según haya la necesidad, por lo que el esquema de encripción no será el mismo.

Es importante aclarar que la conexión segura entre un punto a otro debe involucrar a que el cliente seguro se está conectando a un servidor seguro. Si se conecta a un extremo inseguro y de ahi se establece la conexión segura no funcionará adecuadamente el protocolo, haciéndolo que pueda ser espiado.

¿ De que Previene Secure Shell?

Debido a la promiscuidad de la interfaz ethernet, se genera una problemática sobre los siguientes servicios de red usados en la actualidad, tales como:

- telnet
- ftp
- http
- rsh
- rlogin
- rexec

Ello nos representa un problema importante, ya que, incluso en un entorno de red cerrado, debe existir como mínimo un medio seguro para poder desplazar archivos, hacer copia de archivos, establecer permisos, ejecutar archivos, scrips, etc, a través de medios seguros. Por ello para evitar que determinadas personas capturen el trafico diario de la red, es conveniente instalar el **Secure Shel** (**ssh**).

Entre los ataques más comunes que nos previenen **Secure Shell** están:

- Sniffering(Captura de trafico)
- IP Spoofing
- MACpoofing
- DNS Spoofing
- Telnet Hickjacking
- ARP Spoofing
- IP Routing Spoofing
- ICMP Spoofing

Instalación de Secure Shell cliente/servidor para Unix

Desafortunadamente los protocolos de **Secure Shell** (**ssh1** y **ssh2**) no son compatibles uno con otro, por lo tanto, sí deseamos que exista compatibilidad debemos de instalar primero **Secure Shell** protocolo **ssh1** y posteriormente **Secure Shell** protocolo **ssh2**.

Otra opción para mantener la compatibilidad con los dos protocolos sin problema alguno es instalar **OpenSSH**.

Secure Shell con protocolo **ssh1** y **ssh2** pueden instalarse exactamente igual tal como se muestra a continuación.

Después de obtener el programa de **Secure Shell** procedemos a desempacarlo:

```
$ gunzip ssh-x.x.x.tar.gz
$ tar -xvf ssh-x.x.x.tar
```

En este punto obtendremos un directorio ssh-x.x.x sobre la ruta donde desempacamos.

A continuación se citan los pasos necesarios configurar y compilar **Secure Shell**. Estos pasos pueden realizarse sin ser root.

a) Configuración del entorno de compilación.

A diferencia de otras herramientas ssh no requiere de editar el archivo Makefile, la configuración la realizamos a través de los parámetros que le pasemos al script llamado configure.

Dentro del directorio ssh-x.x.x se encuentra el script llamado configure, el cual tiene los siguientes argumentos validos:

--prefix=PREFIX	Donde se instalarán los binarios por default /usr/local.
--exec_prefix=PREFIX	Donde se instalarán los ejecutables por default es el mismo que la variable --prefix.
--with-rsh=PATH	Permitirá comandos rsh utilizando la estructura de ssh.
--without-idea	No incluir IDEA
--with-tis=PATH	Soporte a mecanismo de autenticación Tis authsrv.
--with-etcdir=PATH	Ruta sobre la que obtendrá información sobre el sistema por default /etc.
--with-libwrap=[PATH]	Usa libwrap (tcp_wrappers) y inetd.
--with-socks4[=PATH]	Incluye soporte para SOCKS (Cruce de firewall).
--with-socks5[=PATH]	Incluye soporte para SOCKS5.
--enable-warnigs	Habilita la bandera -Wall al compilador gcc.

Sí nuestra intención es lograr que tcp-wrapper lleve un control sobre los accesos realizados con ssh, se necesitará la bandera --with-libwrap, además, sí se pretende realizar una autenticación de la máquina para ejecutar comandos rsh, se requiere entonces la bandera --with-rsh.

Entonces ejecutamos el script tomando en cuenta que tenemos instalado TCP-Wrappers y con soporte para el uso de comandos rsh:

$ ./configure --with-libwrap=/path/libwrap.a --wit-rsh=/path/rsh.

Esto genera los cambios necesarios en los archivos de código fuente para su correcta compilación.

Para obtener la ruta del comando rsh utilizar el comando whereis.

*b) Compilar los binarios de **Secure Shell** (se requiere el compilados GCC). Para esto basta con ejecutar:*
$ make

Nota importante: Para poder ejecutar la instalación es necesario estar dentro de la cuenta de root.

c) *Instalación de Secure Shell en el sistema y obtener las llaves del host.*

$ make install

Los archivos de configuración de Secure Shell (ssh_host_key y sshd_config) quedan localizados en el directorio "/etc", los programas clientes (ssh y scp) quedan en "/usr/local/bin". Finalmente el programa servidor o demonio de Secure Shell (sshd) queda localizado en "/usr/local/sbin".

d) Ya que se realizó la instalación en el equipo procedemos a configurar el sistema para poder ejecutar el demonio del servidor de ssh y permitir accesos por el puerto por default de Secure Shell (port 22).
Editar el archivo /etc/inetd.conf e incluir la siguiente línea:
(Sí no tiene habilitado TCP-Wrapper)

```
ssh  tcp  root  nowait  /usr/local/sbin/sshd  /usr/local/sbin/sshd
                                                -i
```

(Sí se esta usando TCP-Wrapper)

```
ssh  tcp  root  nowait  /usr/local/etc/tcpd  /usr/local/sbin/sshd
                                               -i
```

Editar el archivo /etc/services y habilitar el puerto para Secure Shell usando la siguiente línea:

```
ssh      22/udp      Secure Shell
```

Por ultimo reiniciar el demonio de inetd.

Obtener el número de proceso del demonio inetd.
$ ps -fea | grep inetd

Enviar la señal HUP al proceso del inetd.
$ kill -HUP procesid

En este punto el sistema debe responder a las peticiones de conexión por **Secure Shell**