UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Proyecto de Redes de Computadores I: “Acceso remoto aplicado al protocolo SSH”

Integrantes:
Rodrigo Muñoz 201421028-0
Steev González 201421009-4
Profesor: Agustín González
Asignatura: ELO322 - “Redes de Computadores I”

Introducción

En este trabajo presentaremos el protocolo SSH, su utilidad y características, enfocándonos en una de sus prestaciones más importantes como lo es el acceso a equipos remotos, abarcando aspectos de seguridad, credenciales y encriptación, llevando estas utilidades a la práctica, mostrando experiencias de la aplicación de este protocolo y exponiendo una realizada por nosotros.

¿Qué es SSH?

Es un protocolo creado para obtener acceso remoto total a terminales desde otro terminal y está regido por una arquitectura cliente-servidor, cuya principal característica es servir como una alternativa muy segura a la hora de facilitar la manipulación y las comunicaciones entre dos sistemas terminales que se encuentre físicamente distanciados. Su gran confiabilidad reside en que certifica los hosts y encripta los datos a transmitir, por lo cual, en la eventualidad de que un usuario malicioso esté interceptando nuestra información, no le será de utilidad ya que se encontrará codificada bajo un sólido sistema de encriptación que revisaremos más adelante.

La primera versión de SSH, conocida como SSH1, consistía en un protocolo y programa libres y fueron creados por un finlandés llamado Tatu Ylönen en 1995, pero su licencia fue cambiando y terminó apareciendo la compañía SSH Communications Security, que lo ofrecía gratuitamente para uso doméstico y académico, pero exigía el pago a otras empresas. A principios de 1999 se empezó a escribir una versión que se convertiría en la implementación libre por excelencia, la de OpenBSD (sistema operativo libre tipo Unix), llamada OpenSSH o también SSH2.

Características: ¿Por qué usarlo?

- **Conexión**: Para empezar, SSH es un protocolo orientado a la conexión por lo cual para entablar un enlace entre ambos terminales se emplea un protocolo afín como TCP. Al establecer una conexión, SSH cuenta con un sistema de "keys" o credenciales para confirmar la identidad de los ordenadores involucrados, es decir, podrá determinar si es que se trata del mismo servidor al cual se conectó anteriormente o de algún tipo de amenaza de seguridad.

- **Seguridad**: Sin duda el punto fuerte de SSH respecto a otros protocolos similares, pero menos seguros como telnet, rsh o rcp. Como ya indicamos cuenta con un robusto sistema de encriptación, tanto para la autenticación de credenciales como para el tráfico de datos, el cual a su vez cuenta con varios protocolos para este fin, por lo tanto la codificación no sigue un algoritmo fijo en cada ejecución, sin embargo todos se basan en una encriptación de 128 bits. Para obtener el número de posibles combinaciones en el cifrado de 128 bits, multiplica "2" por sí mismo 128 veces. El cifrado de 128 bits tiene más de $3{,}4 \times 10^{37}$ combinaciones. Así, usando una computadora que realice mil millones de cálculos por segundo, le tomaría a un hacker más de 1 sextillón de años para descomponer una clave cifrada de 128 bits.

Aplicaciones

Acceso remoto

Como se mencionó anteriormente, SSH sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos y también puede redirigir el tráfico de X, un programa para obtener interfaz gráfica en sistemas UNIX, para poder ejecutar programas gráficos si tenemos disponible un Servidor X.

Tunneling

En esencia el tunneling es el proceso de colocación de cada paquete de información que se envía dentro de otro paquete que hace de encapsulado.

El protocolo SSH se utiliza con frecuencia para *tunelizar* tráfico confidencial sobre Internet de una manera segura. Por ejemplo, un servidor de ficheros puede compartir archivos usando el protocolo HTTP, cuyos datos no viajan cifrados. Esto permitiría que una tercera parte, que tuviera acceso a la conexión pudiera examinar a conciencia el contenido de cada fichero transmitido.

Para poder montar el sistema de archivo de forma segura, se establece una conexión mediante un túnel SSH que encamina todo el tráfico HTTP al servidor de archivos dentro de una conexión cifrada SSH. Aunque el protocolo HTTP sigue siendo inseguro, al viajar dentro de una conexión cifrada se impide el acceso al mismo.

Por ejemplo, para conectar con un servidor web de forma segura, utilizando SSH, haríamos que el cliente WEB, en vez de conectarse al servidor directamente, se conecte a un cliente SSH. El cliente SSH se conectaría con el servidor tunelizado, el cual a su vez se conectaría con el servidor WEB final. Lo atractivo de este sistema es que hemos añadido una capa de cifrado sin necesidad de alterar ni el cliente ni el servidor web.

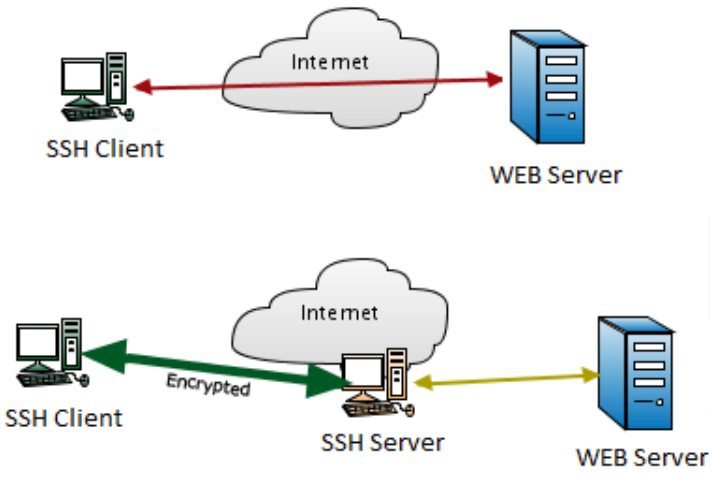

Figura 1: Ejemplo de tunneling utilizando un proxy y SSH.

Conocimientos en práctica

En la parte práctica de nuestro informe, como grupo quisimos mostrar cómo podemos usar SSH para no solo operar un computador a distancia, sino también, que podemos utilizar distintos periféricos a partir de un computador por medio de SSH.

Para esto, montamos una pequeña estación meteorológica en un Raspberry Pi, un pequeño computador, con un servidor SSH implementado e imprime por consola los datos obtenidos por el sensor. Para esto se utilizaron los GPIO (General Purpose Input Output) y como periféricos se utilizó un sensor de temperatura y humedad ambiental y un LED RGB a modo de indicador de funcionamiento, al inicio el LED prende de color verde, indicando que no está realizando una medición. Para obtener acceso a Internet, el Raspberry Pi fue conectado a un router por medio de un cable UTP para una conexión Ethernet y le fue asignada una IP estática.

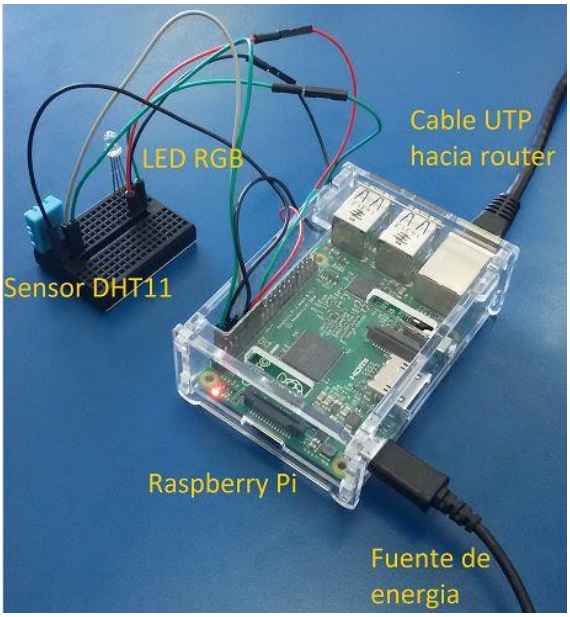

Figura 2: Raspberry Pi conectado a sus periféricos, sensor y LED.

Para realizar la conexión SSH utilizaremos un software cliente llamado PuTTY. Nos conectamos al Raspberry Pi ingresando la IP asignada: 192.168.2.25 y al puerto determinado por el servidor SSH, el puerto 22 (puerto por defecto de SSH).

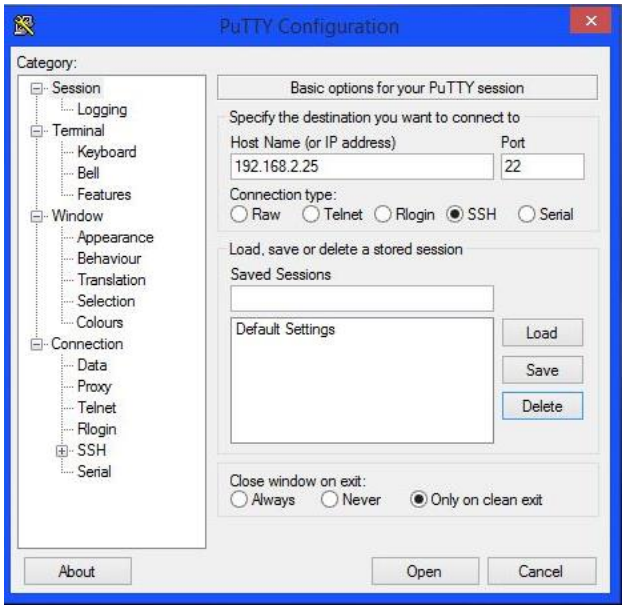

Figura 3: Ventana de PuTTY, con los datos ingresados y listos para realizar la conexión.

Una vez conectados, buscamos entre los directorios y ficheros de la Raspberry Pi nuestro código de Python para realizar las mediciones de temperatura y humedad ambiental por medio del comando: “sudo python est_rpi.py”. Luego el LED de verde cambiará a color rojo, indicando que está realizando una medición.

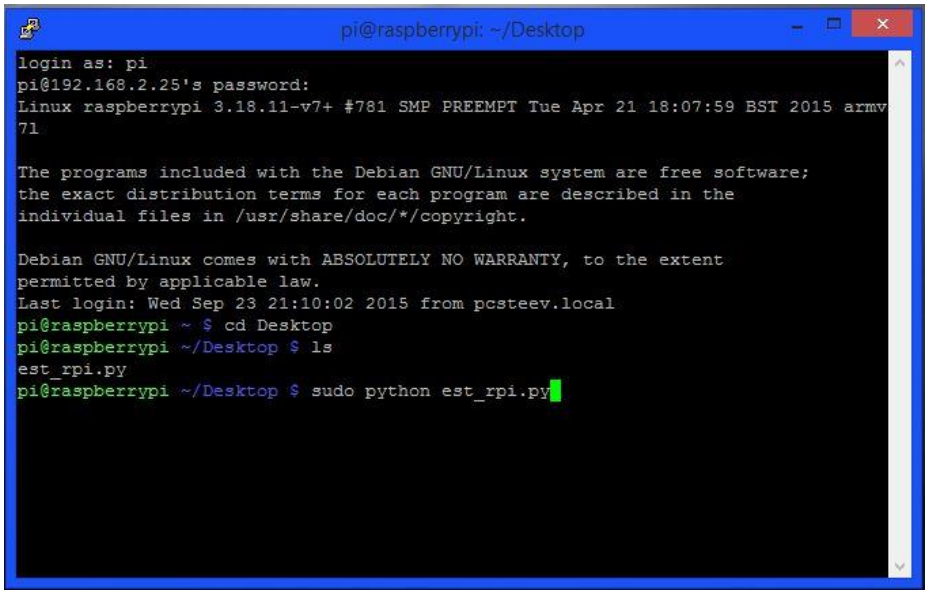

Figura 4: Búsqueda entre los directorios y ejecución del código de Python.

Una vez finalizada la medición, podemos ver en la consola la información recolectada por los sensores. Puede verse que con fecha el 23 de Septiembre del 2015, a las 21:15, habían 9 grados Celsius y un 75% de humedad relativa del aire.

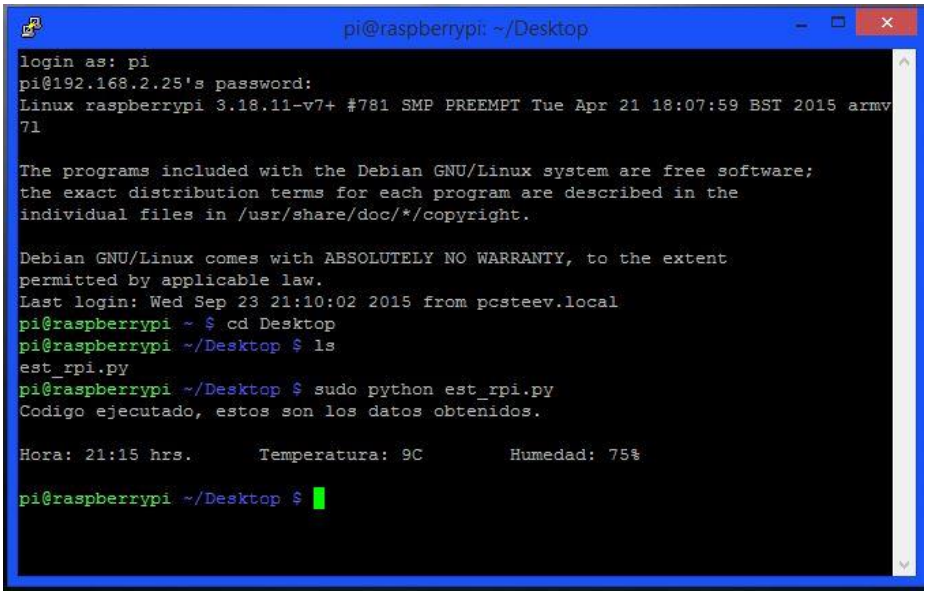

Figura 5: Datos impresos por el código de Python al medir con el sensor.

Como comentario al margen, si bien se conectó al Raspberry Pi, desde la misma red privada y utilizando una IP privada, también es posible conectarse desde una red externa manteniendo el puerto 22 y utilizando la IP publica del router que le fue configurado un port-fowarding.

Conclusión

Empezamos este trabajo con una parte teórica que cubrió los aspectos fundamentales de SSH en cuanto a sus orígenes, características y prestaciones, exponiendo lo útil y seguro que resulta ser como herramienta de encriptación y transferencia de datos, poniendo énfasis en sus funciones más importantes y poniéndolas a prueba en un breve proyecto, ilustrando sus pasos y haciendo en paralelo con lo expuesto en la teoría.

Respecto a la parte práctica, no sólo usamos una interfaz de software en el desarrollo del proyecto sino que también, utilizamos un hardware controlado por el computador lo que nos da más prestaciones a la hora de realizar proyectos mucho más complejos en áreas de sistemas automáticos o domótica , lo que combinado con la gran superficie que abarca Internet en el mundo, nos brinda una potente y siempre útil herramienta, por lo que esta experiencia nos sirvió como un buen acercamiento a métodos de control más avanzados.

Referencias

- http://www.openssh.com/
- http://docstore.mik.ua/orelly/networking_2ndEd/ssh/ch03_03.htm
- http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html
- https://es.wikipedia.org/wiki/T%C3%BAnel_(inform%C3%A1tica)
- https://es.wikipedia.org/wiki/Secure_Shell
- http://2xod.com/articles/ssh%20tunnelling/