SSH - Secure Shell

Muy breves conceptos respecto del cifrado asimétrico

Antes de abordar SSH debemos **comprender los conceptos básicos del cifrado asimétrico** que es una de las herramientas fundamentales que le dan a SSH el ser lo que es y reemplazar lo que hasta ese momento se conocía como el estándar de facto: **telnet**.

SSH utiliza el cifrado asimétrico para **proteger el canal de comunicación del cliente al servidor y del servidor al cliente** de la siguiente manera:

E**l cliente/servidor crea dos claves,** una pública y otra privada en un canal seguro con las siguientes condiciones:

1. Con la pública yo solo puedo cifrar y con la privada solo puedo descifrar.
2. Cada clave pública se corresponde con sólo una privada y viceversa.
3. A partir de la clave pública no se puede deducir la clave privada.
4. La clave privada **JAMÁS se divulga** o comparte; pero la clave pública se entrega a quien sea que me la solicite.

Cuando se está estableciendo la conexión, tanto cliente como servidor van a intercambiar sus claves públicas. De esta forma, siempre que el **cliente quiera enviar un mensaje al servidor lo cifra con la clave pública del server**, que solo podrá ser **descifrado con la privada** que solo está en poder del servido y ya sea que el servidor responda o envíe un mensaje al **cliente, lo cifrará con la clave pública del cliente quien lo descifrará con su propia clave privada**. Es por eso que únicamente los extremos son los que podrán decodificar los mensajes.

El protocolo SSH

Acerca de OpenSSH

El protocolo SSH (Secure Shell) es un protocolo de la capa de aplicación el cual no solo nos permitirá conectarnos de forma segura a equipos remotos, sino que dará seguridad a todo tráfico que sea encapsulado por él. Pero por ahora nos **limitaremos a la conexión a equipos remotos** básicamente **gestionar claves RSA y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.** De manera predeterminada, el protocolo SSH atiende peticiones por el **puerto 22**.

Instalación de OpenSSH

Debemos separar esto en dos partes bien definidas. Por un lado el cliente que:

- **En Windows:** Podremos descargar **putty de la siguiente URL:** https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- En Linux: Es extraño que no venga el cliente preinstalada; pero si así fuera el caso:

```
$ sudo apt install openssh-client
```

PAra instalar el **demonio SSH, que se instala en el servidor al cual deseamos conectarnos** ejecutaremos el siguiente comando:

```
$ sudo apt install openssh-server
```

Archivos de configuración de OpenSSH

OpenSSH dispone de dos conjuntos diferentes de archivos de configuración. Por un lado **los archivos de configuración del servidor que definen el comportamiento del daemon** y por otro lado **los archivos de configuración del lado del cliente** que definen las opciones las cuales utilizará el cliente siempre que se conecte.

Archivos de configuración del servidor

La ubicación de los archivos referentes al servidor los podremos ver en la ruta **`/etc/ssh/.`**
Dentro del directorio podemos encontrar los siguientes archivos de configuración:

- **moduli:** El intercambio de claves mencionados anteriormente se hace por un canal seguro. Este canal seguro es establecido utilizando el protocolo Diffie-Hellman. Este archivo contiene información para establecer dicho canal.
- **ssh_config:** Este es el archivo de configuración **del cliente por defecto, si no es que hay uno ya dentro del usuario.**
- **sshd_config:** El archivo de configuración para el demonio sshd.
- **ssh_host_*key**: Diferentes claves privadas.
- **ssh_host_key*.pub:** Las claves públicas asociadas a las claves privadas.

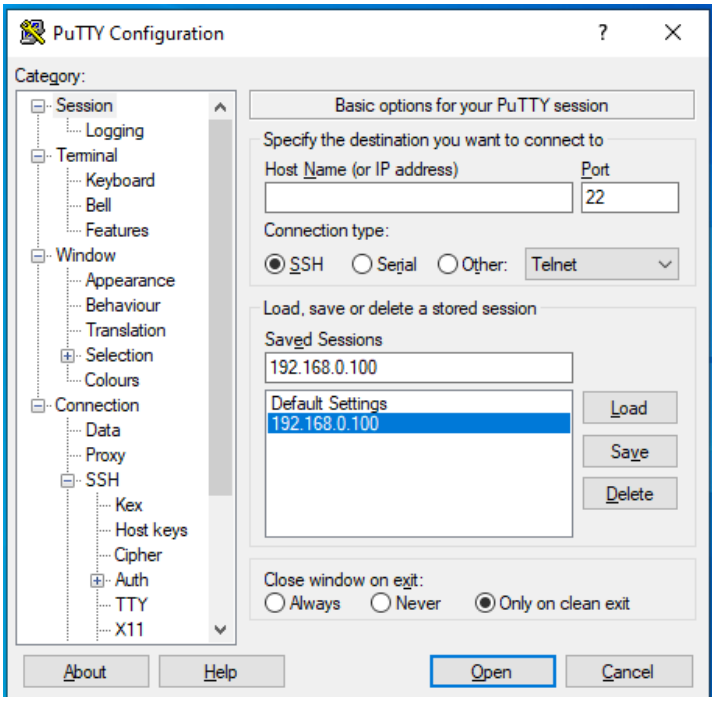
Archivos de configuración del lado del cliente

Estos se encuentran dentro de la carpeta **home del usuario, subcarpeta .ssh archivo config.** Esto es **`~/.ssh/config`** o dentro de Windows podemos especificarlo nosotros.
En Linux, si no existiese, como el usuario en cuestión **debemos ejecutar el siguiente comando que nos creará la carpeta junto con la clave pública y privada** que utilizará nuestro usuario por defecto, o cada vez que queramos loguearnos como el usuario:

```
$ ssh-keygen -b 2048 -t rsa -N ''
```

Donde:

- **-b**: tamaño de la clave.
- **-t**: tipo de clave.
- **-N**: passphrase. Es la passphrase que necesitamos para desbloquear la clave. No la requeriremos en este ejemplo.



En los sistemas Linux, veamos que hay dentro de la carpeta creada anteriormente:

```
$ ls ~/.ssh/
id_rsa  id_rsa.pub  authorized_keys
```

- **authorized_keys:** Este archivo lo vamos a encontrar dentro de **los servidores** y contiene una lista de claves públicas **de los clientes autorizados**. Esto es, si la clave pública del cliente se encuentra aqui dentro podremos autenticarnos si especificamos el usuario en e cual se encuentra dicho archivo.
- **id_rsa:** Contiene la clave privada DSA del usuario.
- **id_rsa.pub** La clave pública DSA del usuario.
- **known_hosts:** A medida que nos vayamos conectando a diferentes servidores, el fingerprint de dichos servers se almacena aquí asociado al hostname.

Archivo “sshd_config”

El archivo de configuración **`/etc/ssh/sshd_config`** de OpenSSH es de vital importancia porque definen el comportamiento de nuestro servidor.

Parámetros del archivo sshd_config

A continuación, dentro de nuestro server, podremos ubicarlo en la carpeta **`/etc/ssh/sshd_config`** y ver el contenido con el siguiente comando:

```
$ cat /etc/ssh/sshd_config
```

Ahora veamos algunos parámetros que podemos configurar:

Cambiando el puerto por defecto

SSH tiene asignado por defecto el puerto 22, esto es algo que conocen todos nuestros posibles atacantes, por lo que es una buena idea cambiarlo. Probablemente durante el curso lo dejaremos por defecto en el puerto 22, pero sepan que existe dicha posibilidad.

```
Port 2022
```

Deshabilitar el acceso de root

Este es quizá el parámetro más importante de seguridad. No permitiremos que nuestros usuarios accedan utilizando a root de forma directa. Y de paso permitiremos únicamente un solo intento antes de desconectar.

```
PermitRootLogin no
MaxAuthTries 1
```

Impidiendo forwardear ventanas

Si nuestro servidor tienen entorno gráfico instalado o no, o no queremos que los usuarios se conecten a él, definiremos esta opción en el archivo de configuración. No habilitemos nada que no sea necesario

```
X11Forwarding yes
```

Limitando el tiempo de autenticación

El número indica la cantidad de segundos en que la pantalla de login estará disponible para que el usuario capture su nombre de usuario y contraseña. Si no lo hace, el login se cerrará, evitando así dejar por tiempo indeterminado pantallas de login sin que nadie las use, o peor aún, que alguien esté intentando mediante un script varias veces adivinar un usuario y contraseña. Daremos únicamente 10 segundos.

```
LoginGraceTime 30
PermitRootLogin no
StrictModes yes
MaxAuthTries 2
```

Iniciando el servicio SSH

Muy probablemente ya se encuentre configurado el inicio automático del demonio de SSH dentro de su sistema Linux. Podremos revisarlo con el siguiente comando:

```
$ sudo systemctl status sshd
```

```
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor
preset: enabled)
     Active: active (running) since Wed 2022-10-12 15:50:22 -03;
54min ago
```

Sino podremos iniciarlo ejecutando el siguiente comando:

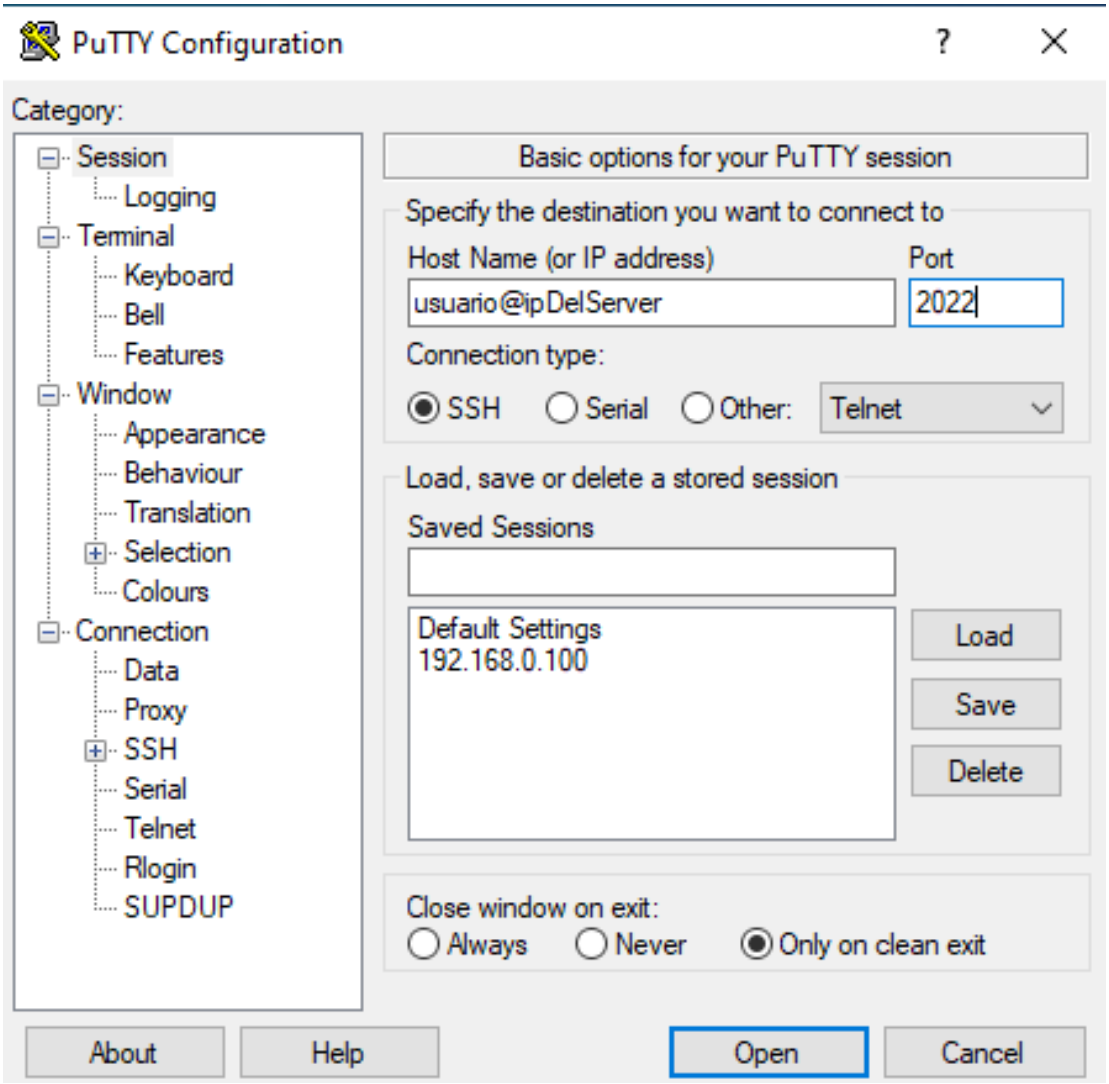
```
$ sudo systemctl start sshd
```

Conectándonos a un equipo utilizando SSH

Para establecer una conexión con un servidor SSH remoto desde haremos uso de una **terminal en el caso de Linux** o **Mac y** de **putty en el caso de Windows**. La sintaxis para llevar a cabo esta operación es una de las siguientes:

```
$ ssh usuario@ipDelServidor
$ ssh -l usuario ipDelServidor
$ ssh -l usuario ipDelServido -p 2022
```

O lo que es lo mismo en máquinas **Windows**



Fingerprint SSH

Algo que notaremos la primera vez que nos conectemos a un servidor será que nos dice algo de un **fingerprint**. ¿Qué es eso?
La idea de SSH es que nadie nos engañe al momento de conectarnos a un servidor, entonces la advertencia que nos muestra es que es la primera vez que nos conectamos a ese servidor y que el fingerprint o huella es la siguiente.

La huella está **asociada a la clave pública del servidor**. O sea como cada servidor tiene su clave pública, **cada fingerprint será diferente.**
Como es la primera vez que nos conectamos, nos está avisando que no tiene ese fingerprint.
Una vez que se **acepte el fingerprint**, el mismo se **almacenará en la carpeta: `/home/usuario/.ssh/known_hosts.`** A partir de ese momento, cada vez que nos conectemos a esa URL/IP, siempre deberá tener el mismo fingerprint, lo que implicará que siempre tendrá la misma clave pública, lo que implica que tendrá la misma clave privada y como esta no se divulga implicará que es siempre el mismo server.

Si alguien quisiera **hacerse pasar** por el servidor al que nos queremos conectar para robar tráfico, cuando el **cliente detecta que el fingerprint es distinto a la IP** que tenía almacenada en el **known_hosts** nos **denegará el acceso.**

Pero un ejemplo más que común es cuando s**e le asigna la IP a otro servidor** o cuando **reinstalamos ese servidor** y no respaldamos las claves. En ambos casos nos dará la misma alerta. Si este es el caso, deberemos ingresar a la línea known_hosts que nos indica y eliminar esa línea, o **si es posible ejecutar el siguiente comando**:

```
$ ssh-keygen -R hostname
```

Un ejemplo de dicho error es el siguiente:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
5c:9b:16:56:a6:cd:11:10:3a:cd:1b:a2:91:cd:e5:1c.
Please contact your system administrator.
Add correct host key in /home/user/.ssh/known_hosts to get rid of
this message.
Offending key in /home/user/.ssh/known_hosts:1
RSA host key for ras.mydomain.com has changed and you have requested
strict checking.
Host key verification failed.
```