Índice de contenido

Información de Derechos reservados de esta publicación.

Autor del manual: Javier Eduardo Duarte

Tema 4. Seguridad con openSSH

4.1 El Protocolo SSH

El protocolo SSH (Secure Shell) es una herramienta que nos permite conectarnos a equipos remotos (Servidores en Producción) así mismo, nos da la capacidad de llevar a cabo tareas administrativas dentro del mismo como, activar o apagar servicios,

Además de la conexión a otros equipos, SSH nos permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

Una clave RSA (Sistema Criptografico con Clave Publica) es un algoritmo que genera un par de llaves de autenticacion, la publica y la privada. La publica se distribuye en forma autenticada y la privada que generalmente es guardada en secreto por el propietario.

El protocolo SSH (Secure Shell) esta implementado bajo el estándar TCP/IP, el cual a su vez se encuentra dividido en 5 secciones:

1. Nivel Físico
2. Nivel De Enlace
3. Nivel de Internet
4. Nivel de Transporte
5. Nivel de Aplicación

por lo que el protocolo SSH esta ubicado en la quinta capa del modelo TCP/IP, nos referimos a la capa de aplicación

La capa de aplicación es el nivel que los programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar.

De manera predeterminada, el protocolo SSH atiende peticiones por el puerto 22

En este capitulo haremos uso de OpenSSH la cual es la alternativa libre y abierta al programa propietario SSH

4.2 Acerca de OpenSSH

OpenSSH (Open Secure Shell) es un conjunto de aplicaciones que permiten realizar comunicaciones cifradas a través de una red, usando como base al protocolo SSH. Este proyecto es liderado actualmente por Theo de Raadt quien actualmente es fundador y líder de proyectos como OpenBSD.

Los desarrolladores de OpenSSH aseguran que este es más seguro que el original, lo cual es debido a la conocida reputación de los desarrolladores de OpenBSD por crear código limpio y perfectamente auditado, lo que contribuye a que sea más seguro. Su seguridad también es atribuible al hecho de que su código fuente se distribuya libremente con una licencia BSD. Aunque todo el código fuente del SSH original también está disponible, existen restricciones con respecto a su uso y distribución, lo que convierte a OpenSSH en un proyecto mucho más atractivo a la hora de atraer nuevos desarrolladores.

Además de la conexión a otros equipos, openSSH nos permite copiar datos de forma segura mediante la implementación de dos herramientas proporcionadas por openSSH, estas son:

- SCP
- SFTP

Estas herramientas en realidad tienen la misma función de copiado solo se diferencian en la forma en como son aplicadas tema del cual hablaremos mas adelante.

4.3 Instalando OpenSSH

A partir de este punto empezaremos a descargar los paquetes necesarios para el perfecto funcionamiento de openSSH, de esta manera si usted se encuentra trabajando bajo algún ambiente gráfico, sea KDE o GNOME le pedimos abra una terminal de BASH, por otra parte si usted se encuentra trabajando bajo linea de comandos no tendrá que hacer nada.

Los paquetes a descargar son los siguientes:

- openssh
- openssh-clients
- openssh-server

La forma en que se instalaran estos paquetes sera tecleando en consola lo siguiente:

```
[root@ localhost ] # yum install -y openssh openssh-clients openssh-server
```

Una vez finalizado el proceso de instalación pasaremos con las configuraciones propias de openSSH, nos referimos a los ficheros de configuración

4.4 Archivos de configuración de OpenSSH

OpenSSH dispone de dos conjuntos diferentes de ficheros de configuración: uno completamente dedicado al cliente (ssh, scp y sftp) y otro orientado completamente al servidor.

4.4.1 Archivos de configuración del lado del servidor

La ubicación de los ficheros de configuración referentes al servidor openSSH se encuentran en la siguiente ruta:

```
/etc/ssh/
```

Dentro del directorio podemos encontrar los siguientes ficheros de configuración:

moduli	Contiene grupos Diffie-Hellman usados para el intercambio de la clave Diffie-Hellman que es imprescindible para la construcción de una capa de transporte seguro. Cuando se intercambian las claves al inicio de una sesión SSH, se crea un valor secreto y compartido que no puede ser determinado por ninguna de las partes individualmente. Este valor se usa para proporcionar la autenticación del host.

ssh_config	El archivo de configuración del sistema cliente SSH por defecto. Este archivo se sobrescribe si hay alguno ya presente en el directorio principal del usuario
sshd_config	El archivo de configuración para el demonio sshd
ssh_host_dsa_key	La clave privada DSA usada por el demonio sshd
ssh_host_dsa_key.pub	La clave pública DSA usada por el demonio sshd
ssh_host_key	La clave privada RSA usada por el demonio sshd para la versión 1 del protocolo SSH.
ssh_host_key.pub	La clave pública RSA usada por el demonio sshd para la versión 1 del protocolo SSH.
ssh_host_rsa_key	La clave privada RSA usada por el demonio sshd para la versión 2 del protocolo SSH.
ssh_host_rsa_key.pub	La clave pública RSA usada por el demonio sshd para la versión 2 del protocolo SSH.

4.4.2 Archivos de configuración del lado del cliente

La ubicación de los ficheros referentes al cliente se encuentran almacenados en el directorio de trabajo de cada usuario:

Ejemplo: "/home/usuario/"

Dentro del directorio podemos encontrar los siguientes ficheros de configuración:

authorized_keys	Este archivo contiene una lista de claves públicas autorizadas. Cuando un cliente se conecta al servidor, el servidor autentica al cliente chequeando su clave pública firmada almacenada dentro de este archivo.
id_dsa	Contiene la clave privada DSA del usuario.
id_dsa.pub	La clave pública DSA del usuario
id_rsa	La clave RSA privada usada por ssh para la versión 2 del protocolo SSH.
id_rsa.pub	La clave pública RSA usada por ssh para la versión 2 del protocolo SSH.
identity	La clave privada RSA usada por ssh para la versión 1 del protocolo SSH.

identity.pub	La clave pública RSA usada por ssh para la versión 1 del protocolo SSH.
known_hosts	Este archivo contiene las claves de host DSA de los servidores SSH a los cuales el usuario ha accedido. Este archivo es muy importante para asegurar que el cliente SSH está conectado al servidor SSH correcto

4.4.3 Configuración de fichero sshd_config

La función que desempeñan los ficheros de configuración de openSSH son de vital importancia para la seguridad de nuestro servidor , ya que si no se llegaran a configurar apropiadamente estos ficheros la vulnerabilidad de nuestro servidor seria demasiado sensible a ataques informáticos, es por ello que le enseñaremos la manera apropiada en la que deberá ser configurado este vital fichero.

4.4.3.1 Blindando el fichero sshd_config

Este fichero lo podrá localizar en en la siguiente ruta

```
/etc/ssh/
```

El siguiente paso sera abrir el fichero con la ayuda del editor de textos VI

```
[root@ localhost #] vi /etc/ssh/sshd_config
```

A partir de este punto comenzaremos a blindar SSH

4.4.3.2 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.

Para modificar esta opción y las siguientes que iremos mencionando editaremos el fichero de configuración **shd_config**, que por defecto se encuentra en el directorio **/etc/ssh/**.

Se recomienda usar un puerto cualquiera por encima del 1024, así que usted puede elegir el que quiera. En este ejemplo usaremos el 34765, por lo que tendrá que editar el parámetro ***Port*** del fichero de configuración el cual deberá quedar así:

```
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

Port 34567
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

4.4.3.3 Desactivando el Protocolo 1

Hay dos versiones de ssh en cuanto a su protocolo de comunicación, estas son:

- Versión 1
- Versión 2.

La versión 1 de openSSH hace uso de varios algoritmos de cifrado de datos mas sin embargo, algunos de estos algoritmos han dejado de ser mantenidos por sus creadores y por lo tanto presenta serios huecos de seguridad que potencialmente permite a un intruso insertar datos en el canal de comunicación. Para evitar el uso del protocolo 1 y sus posibles ataques a este, basta con indicar que solo admita comunicaciones de ssh basadas en el protocolo 2, por lo que tendrá que editar el parámetro **Protocol** del fichero de configuración el cual deberá quedar así:

```
# Disable legacy (protocol version 1) support in the server for new
# installations. In future the default will change to require explicit
# activation of protocol 1
Protocol 2
```

4.4.3.4 Deshabilitando el acceso a root

Este es quizá el parámetro mas importante de seguridad que podemos indicar para blindar nuestro servidor. Prácticamente la mayoría de sistemas operativos Linux crean por defecto al usuario root , es por ello que la mayoría de los ataques informáticos se concentran en atacar al equipo a través de la cuenta de root y mucho mas si la cuenta tiene asignada una contraseña débil

Una manera de deshabilitar el logeo al sistema a través de la cuenta de root es poner en '**no**' la variable **PermitRootLogin** , con esto el usuario root no tendrá permiso de acceder mediante ssh y por lo tanto cualquier intento de ataque directo a root será inútil. Con esto siempre tendremos que ingresar como un usuario normal y ya estando adentro entonces mediante un **su –** cambiarnos a la cuenta de root.

Para llevar a cabo estos cambios tendrá que editar el parámetro **PermitRootLogin** del fichero de configuración el cual deberá quedar de la siguiente manera:

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
```

4.4.3.5 Definiendo un número máximo de intentos de conexión

Muchos de los ataques llevados a cabo por piratas informáticos se basan en fuerza bruta, estableciendo un número máximo de intentos de conexión lograremos que sus intentos por entrar a nuestro servidor sean disuadidos.

Para llevar a cabo estos cambios tendrá que editar el parámetro **MaxAuthTries** del fichero de configuración el cual deberá quedar de la siguiente manera:

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 2
```

El número **2** indica la cantidad de veces que podemos equivocarnos al ingresar el usuario y/o contraseña, en este caso después de dos intentos, se perderá o cerrará la conexión. Claro, es totalmente posible volver a intentarlo, pero con solo dos intentos por vez.

4.4.3.6 Activando el modo estricto

La opción **StrictModes** debe activarse para que, por ejemplo, los usuarios que establecen permisos de escritura para todos en sus ficheros y directorios no se lleven una desagradable noticia cuando otro usuario los modifique, de esta manera se protege la información de los usuarios.

Para llevar a cabo estos cambios tendrá que editar el parámetro **StrictModes** del fichero de configuración el cual deberá quedar de la siguiente manera:

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
StrictModes yes
MaxAuthTries 2
```

4.4.3.7 Impidiendo la conexión al servidor gráfico

Si nuestro servidor no tienen entorno gráfico instalado, o no queremos que los usuarios se conecten a él, definiremos esta opción en el fichero de configuración:

Para llevar a cabo estos cambios tendrá que editar el parámetro **X11Forwarding** del fichero de configuración el cual deberá quedar de la siguiente manera:

```
# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding yes
```

4.4.3.8 Limitando el tiempo para autenticarse con SSH

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 aun, que alguien este intentando mediante un script varias veces el adivinar un usuario y contraseña. Si somos el único usuario del sistema considero que con 20 o 30 segundos es mas que suficiente.

Para llevar a cabo estos cambios tendrá que editar el parámetro **LoginGraceTime** del fichero de configuración el cual deberá quedar de la siguiente manera:

```
# Authentication:

LoginGraceTime 30
PermitRootLogin no
StrictModes yes
MaxAuthTries 2
```

4.5 Iniciar, detener o reiniciar el servidor openSSH

Llegado a este punto usted ya deberá contar con las configuraciones de seguridad apropiadas, por lo que solo faltaría iniciar el servicio de SSH.

Para iniciar el servicio de SSH tendrá que teclear en consola y como **root** lo siguiente:

```
[root@ localhost ]# /etc/init.d/sshd start
```

Igualmente existen opciones ya sea para reiniciar, detener, recargar o conocer el status en el que se encuentra el servicio. Estas opciones pueden ser consultadas en la siguiente tabla:

start	**Inicia el servicio**
stop	**Detiene el servicio**
restart	**Reinicia el servicio.-La diferencia con reload radica en que al ejecutar un restart este mata todos los procesos relacionado con el servicio y los vuelve a generar de nueva cuenta**
reload	**Recarga el servicio.-La diferencia con restart radica en que al ejecutar un reload este solamente carga las actualizaciones hechas al fichero de configuración del servicio sin necesidad de matar los procesos relacionados con el mismo, por lo que podría entenderse que hace el cambio en caliente.**
condrestart	**Reinicio Condicional.- Solamente se inicia si el servicio se encuentra ejecutándose.**
status	**Da a conocer el estado en el que se encuentra el servicio**

4.6 Anexando el servicio de SSH al arranque del servidor

Para añadir el servicio de SSH al arranque del servidor solo tendrá que teclear en consola y como **root** lo siguiente:

```
[root@ localhost ]# chkconfig sshd on
```

Esto es útil cuando por motivos ajenos a usted se reinicia el servidor, de esta manera cuando el equipo arranque, automáticamente levantara el servicio de SSH sin necesidad de levantarlo manualmente después.

4.7 Aprendiendo a utilizar openSSH

En esta parte del capitulo le enseñaremos a:

- Conectarse a un equipo remotamente a través de SSH
- Copiar archivos o carpetas desde un equipo remoto
- Enviar archivos o carpetas a un equipo remoto

4.7.1 Conectándose a un equipo remoto a través de SSH

Para establecer una conexión con un servidor SSH remoto desde Centos haremos uso del Bash, o también conocido como Terminal.

La sintaxis para llevar a cabo esta operación es la siguiente:

```
[root@ localhost ]# ssh usuarioRemoto@ipDelServidorRemoto
```

En caso de haber establecido un puerto de escucha distinto al puerto 22, solo deberá especificar el puerto por el cual requiere autenticarse al servidor. Ejemplo

```
[root@ localhost ]# ssh -p[puertoDeEscucha] usuarioRemoto@ipDelServidorRemoto
```

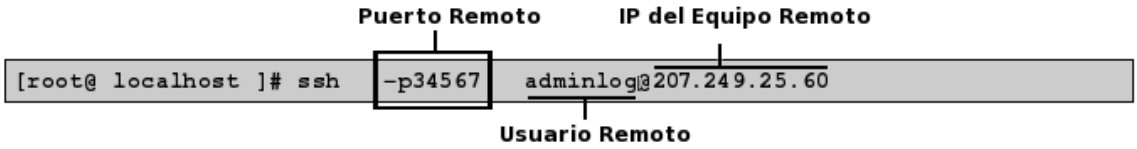
Ejemplo 1: La empresa Factor Integración para la cual trabajamos, nos ha pedido reiniciar el servicio de apache , para ello nos ha proporcionado los siguientes datos:

- IP del servidor remoto -> **207.249.25.60**
- Nombre del usuario remoto -> **adminlog**
- Puerto de autenticacion -> **34765**

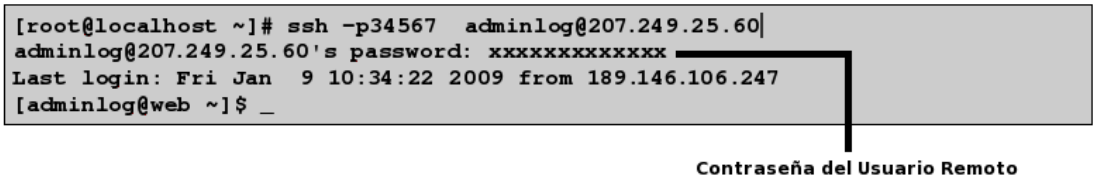
Solución:

1.-Para conectarnos al servidor remoto habrá que especificar el puerto de escucha, el usuario remoto y la IP del servidor remoto

(Recuerde que no esta permitido conectarse como root desde SSH)



2.-El siguiente paso sera teclear la contraseña del usuario remoto



3.-Una vez dentro del servidor remoto nos logearemos ahora si como **"root"**

```
[adminlog@ web ]# su -
Contraseña: xxxxxxxxxxxxxxxxx
[root@ web ]#
```

4.-Por ultimo, solo bastara reiniciar el servidor de apache

```
[root@ web ]# /etc/init.d/httpd restart
```

5.-Para salir del SSH solo basta teclear "exit"

```
[root@ web ]# exit
Connection to 207.249.25.60 closed.
[root@ localhost ~] # _
```

4.7.2 Copiar u obtener archivos o carpetas desde un equipo remoto

Para copiar archivos, ficheros o carpetas desde un equipo remoto hacia nuestro equipo existen dos maneras:

- Mediante el uso del comando SCP
- Mediante el uso del comando SFTP

4.7.2.1 Copiando ficheros a través de SCP (Shell Secure Copy)

Es un medio de transferencia segura de archivos entre un equipo local y uno remoto haciendo uso del protocolo Open Secure Shell (openSSH).

La diferencia en utilizar SCP (Shell) y SFTP (Security File Transfer Protocol) para copiar archivos, carpetas o ficheros radica en que para SCP tenemos que conocer exactamente donde se encuentra el recurso que queremos copiar, de otra forma nunca lo descargara, en cambio SFTP nos deja navegar entre las carpetas lo cual hace mas sencillo la ubicación del recurso que deseamos copiar. La única desventaja que presenta **SCP** es que únicamente permite la transferencia de archivos (descarga y subida de ficheros).

La sintaxis de SCP para llevar a cabo esta operación es la siguiente:

```
[root@ localhost ]#scp usuarioRemoto@ipDelServidorRemoto:rutaDelRecursoRemoto
```

En caso de haber establecido un puerto de escucha distinto al puerto 22, solo deberá especificar el puerto por el cual requiere autenticarse al servidor.

Ejemplo:

```
[root@ localhost ]#scp -P[puertoDeEscucha]
usuarioRemoto@ipDelServidorRemoto:rutaDelRecursoRemoto
```

Aunado a esto, para descargar una carpeta tendrá que seguir la siguiente sintaxis:

```
[root@ localhost ]#scp -P[puertoDeEscucha] -r
usuarioRemoto@ipDelServidorRemoto:rutaDelDirectorioRemoto
```

Ejemplo 2: La misma empresa, Factor Integración, nos ha pedido copiar la carpeta de inventarios de la empresa la cual esta hospedada en un servidor remoto , para ello nos ha proporcionado los siguientes datos:

- IP del servidor remoto -> **207.249.25.60**
- Nombre del usuario remoto -> **adminlog**
- Puerto de autenticacion -> **34567**
- Ruta del Recurso Remoto **-> /tmp/Conta**

Solución:

Para poder hacer la copia desde servidor remoto habrá que especificar el puerto de escucha, el usuario remoto, la IP del servidor remoto y la ruta (**sin errores**) del recurso remoto.

Lo anterior nos copiará la carpeta **"/tmp/Conta"** remota en el directorio actual **"."** naturalmente siempre que usuario tenga permisos sobre la carpeta y su cuenta esté entre las de los que pueden hacer ssh.

La opción **"-r"** significa recursivo, es decir, copia la carpeta y todo su contenido, incluidas las subcarpetas y el contenido de éstas.

4.7.2.2 Copiando ficheros a través de SFTP (Security File Transfer Protocol)

El protocolo de transferencia de archivos SFTP es un protocolo que proporciona la transferencia de archivos y la funcionalidad de manipulación de los mismos Se utiliza normalmente con SSH a fin de asegurar la transferencia de archivos.

En comparación de capacidades con el anterior protocolo SCP, que únicamente permite la transferencia de archivos, el protocolo SFTP permite una serie de operaciones sobre archivos, ficheros, o carpetas remotos, en pocas palabras, nos permite navegar directamente en el servidor remoto con el fin de localizar el recurso que deseamos descargar.

La sintaxis de SFTP para llevar a cabo esta operación es la siguiente:

```
[root@ localhost]# sftp usuarioRemoto@ipDelServidorRemoto
```

En caso de haber establecido un puerto de escucha distinto al puerto 22, solo deberá especificar el puerto por el cual requiere autenticarse al servidor. Ejemplo.

```
[root@ localhost]# sftp -o Port=[PuertoDeEscucha]
usuarioRemoto@ipDelServidorRemoto
```

El siguiente paso sera autenticarnos con la contraseña del usuario remoto

```
Connecting to IPDelServidorRemoto...
usuarioRemoto@IPDelServidorRemoto's password:  xxxxxxxxxxxxxxxxxxxx
```

Una vez dentro del servidor solo bastara ejecutar el comando **"get"** para descargar algún fichero o archivo.

```
[root@ localhost]# sftp -o Port=[PuertoDeEscucha]
usuarioRemoto@ipDelServidorRemoto
Connecting to IPDelServidorRemoto...
usuarioRemoto@IPDelServidorRemoto's password:  xxxxxxxxxxxxxxxxxxxx
sftp> get recursoRemoto
```

La siguiente tabla explica mas a detalle los comandos que pueden ser utilizados con **SFTP:**

cd [rutaRemota]	**Cambia de directorio dentro del servidor remoto**
lcd [rutaLocal]	**Cambia de directorio en el equipo local**
chgrp [grp] [rutaRemota]	**Cambia el grupo de trabajo de un fichero remoto. El [grp] tiene que ser un Group ID**
chmod [opciones] [rutaRemota]	**Cambia los permisos de Lectura, Escritura o de Ejecución a un fichero remoto**
chown [own] [rutaRemota]	**Cambia el grupo de trabajo de un fichero remoto. El [own] tiene que ser un User ID**
get [rutaRemota] [rutaLocal]	**Copia un recurso remoto en un equipo local**

lmkdir [rutaLocal]	**Crea una carpeta en el equipo local**
lpwd	**Imprime la ruta local en la cual estamos trabajando**
mkdir [rutaRemota]	**Crea una carpeta en el equipo remoto**
put [rutaLocal] [rutaRemota]	**Sube un fichero o archivo desde una ruta local hasta una ruta remota**
pwd	**Imprime la ruta remota en la cual estamos trabajando**
exit	**Salimos de SFTP**
rename [rutaLocal] [rutaRemota]	**Renombra un un fichero remoto**
rmdir [rutaRemota]	**Borra una carpeta remota**
rm [rutaRemota]	**Borra un fichero remoto**

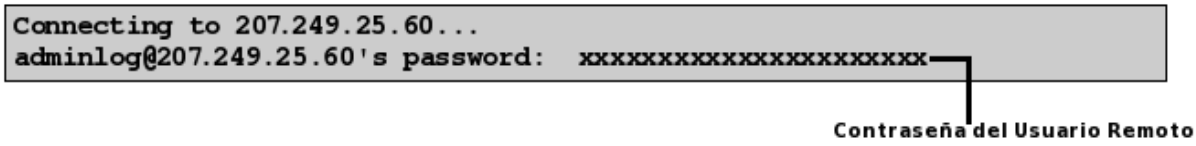
Ejemplo 3: La misma empresa, Factor Integración, nos ha pedido copiar el fichero **inventarioEnero.odt** que se encuentra dentro la ruta **"/tmp/Conta/"** , para ello nos ha proporcionado los siguientes datos:

- IP del servidor remoto -> **207.249.25.60**
- Nombre del usuario remoto -> **adminlog**
- Puerto de autenticacion -> **34567**
- Ruta del Recurso Remoto **-> /tmp/Conta**

Solución:

1.-Para poder traer la copia desde servidor remoto hacia nuestro equipo habrá que especificar el puerto de escucha, el usuario remoto y la IP del servidor remoto

2.- Nos pedirá autenticarnos con la contraseña del usuario remoto, en este caso la contraseña del usuario **"adminlog"**



3.-Una vez autenticados con el servidor nos dará acceso a través de SFTP

```
sftp>
```

4.- Nos moveremos entre directorios con la ayuda del comando **"cd"** hasta estar ubicados en **"/tmp/Conta"**

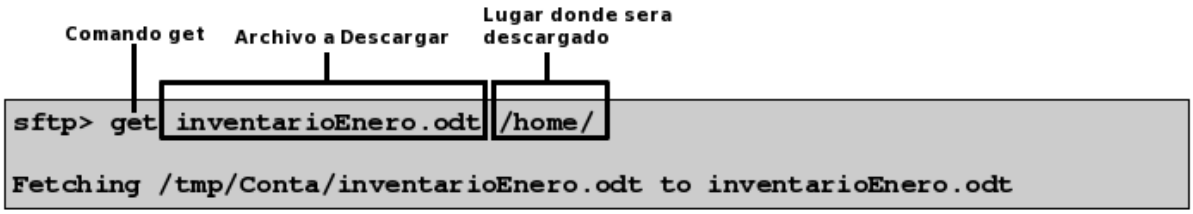
```
sftp> cd /tmp/Conta
```

5.-Dentro de la carpeta **"Conta"** aplicar el comando **"dir"** para visualizar el contenido de la misma

```
sftp>dir

inventarioEnero.odt  inventarioFebrero.odt  inventarioMarzo.odt
```

6.-Con la ayuda del comando **"get"** descargaremos el fichero nombrado **"inventarioEnero.odt"** dentro de la carpeta **"home"** de nuestro sistema



4.7.3 Subir o enviar archivos o carpetas a un equipo remoto

Para subir archivos, ficheros o carpetas desde nuestro equipo hacia un equipo remoto existen dos maneras:

- Mediante el uso del comando SCP
- Mediante el uso del comando SFTP

4.7.3.1 Enviando ficheros a través de SCP (Shell Secure Copy)

La sintaxis de SCP para llevar a cabo esta operación es la siguiente:

```
[root@ localhost ]#scp rutaDelRecursoLocal
usuarioRemoto@ipDelServidorRemoto:rutaRemota
```

En caso de haber establecido un puerto de escucha distinto al puerto 22, solo deberá especificar el puerto por el cual requiere autenticarse al servidor. Ejemplo

```
[root@ localhost ]#scp -P[puertoDeEscucha] rutaDelRecursoLocal
usuarioRemoto@ipDelServidorRemoto:rutaRemota
```

Aunado a esto, para subir una carpeta tendrá que seguir la siguiente sintaxis:

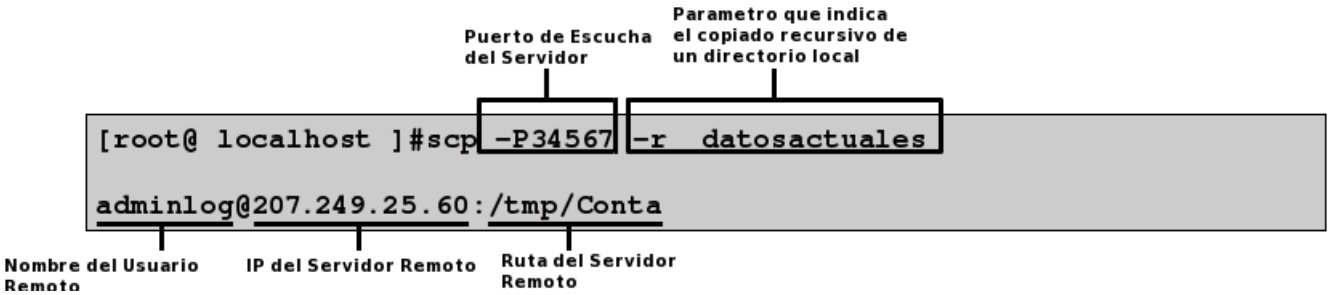
```
[root@ localhost ]#scp -P[puertoDeEscucha] -r directorioLocal
usuarioRemoto@ipDelServidorRemoto:rutaRemota
```

Ejemplo 4: Se nos ha pedido subir una actualización referente a la pagina web de la empresa , para ello nos ha proporcionado los siguientes datos:

- IP del servidor remoto -> **207.249.25.60**
- Nombre del usuario remoto -> **adminlog**
- Puerto de autenticacion -> **34567**
- Ruta del Servidor a donde se tiene que subir la información **-> /tmp/Conta**

Solución:

Para subir este directorio al servidor remoto habrá que especificar la ruta del directorio local,el puerto de escucha, el usuario remoto, la IP del servidor remoto y la ruta (**sin errores**) a donde se quiere enviar el directorio



Luego de haber hecho esto nos pedirá autenticarnos con la contraseña del usuario remoto

```
adminlog@207.249.25.60's password:xxxxxxxxxxxx
```

Al finalizar nos mostrara un ventana mostrando el progreso de cada copia hecha al servidor remoto como la que se muestra a continuación.

```
Actualizacion1.html       100%                0.0KB/s       05:00
actualizacion2.html       100%                0.0KB/s       07:00
actualizacion3.html       100%                0.0KB/s       15:00
actualizacion4.html       100%                0.0KB/s       15:00
actualizacion5.html       100%                0.0KB/s       25:00
actualizacion6.html       100%                0.0KB/s       30:00
actualizacion7.html       100%                0.0KB/s       31:00
actualizacion8.html       100%                0.0KB/s       40:00
```

4.7.3.2 Enviando ficheros a través de SFTP (Security File Transfer Protocol)

La sintaxis de SFTP para llevar a cabo esta operación es la siguiente:

```
[root@ localhost]# sftp usuarioRemoto@ipDelServidorRemoto
```

En caso de haber establecido un puerto de escucha distinto al puerto 22, solo deberá especificar el puerto por el cual requiere autenticarse al servidor. Ejemplo.

```
[root@ localhost]# sftp -o Port=[PuertoDeEscucha]
usuarioRemoto@ipDelServidorRemoto
```

El siguiente paso sera autenticarnos con la contraseña del usuario remoto

```
Connecting to IPDelServidorRemoto...
usuarioRemoto@IPDelServidorRemoto's password:  xxxxxxxxxxxxxxxxxxxx
```

Una vez dentro del servidor solo bastara ejecutar el comando **"put"** para descargar algún fichero o archivo.

```
[root@ localhost]# sftp -o Port=[PuertoDeEscucha]
usuarioRemoto@ipDelServidorRemoto
Connecting to IPDelServidorRemoto...
usuarioRemoto@IPDelServidorRemoto's password:  xxxxxxxxxxxxxxxxxxxx
sftp> put recursoLocal
```

La siguiente tabla explica mas a detalle los comandos que pueden ser utilizados con **SFTP:**

cd [rutaRemota]	**Cambia de directorio dentro del servidor remoto**
lcd [rutaLocal]	**Cambia de directorio en el equipo local**
chgrp [grp] [rutaRemota]	**Cambia el grupo de trabajo de un fichero remoto. El [grp] tiene que ser un Group ID**
chmod [opciones] [rutaRemota]	**Cambia los permisos de Lectura, Escritura o de Ejecución a un fichero remoto**
chown [own] [rutaRemota]	**Cambia el grupo de trabajo de un fichero remoto. El [own] tiene que ser un User ID**
get [rutaRemota] [rutaLocal]	**Copia un recurso remoto en un equipo local**
lmkdir [rutaLocal]	**Crea una carpeta en el equipo local**
lpwd	**Imprime la ruta local en la cual estamos trabajando**
mkdir [rutaRemota]	**Crea una carpeta en el equipo remoto**
put [rutaLocal] [rutaRemota]	**Sube un fichero o archivo desde una ruta local hasta una ruta remota**
pwd	**Imprime la ruta remota en la cual estamos trabajando**
exit	**Salimos de SFTP**

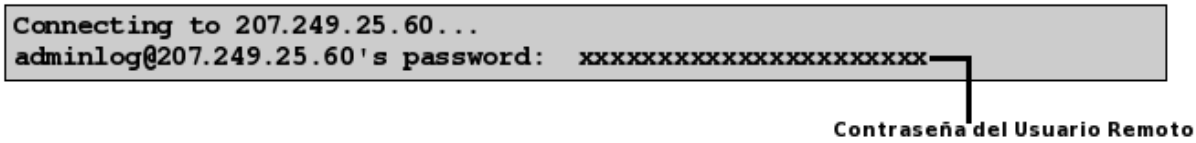
rename [rutaLocal] [rutaRemota]	**Renombra un un fichero remoto**
rmdir [rutaRemota]	**Borra una carpeta remota**
rm [rutaRemota]	**Borra un fichero remoto**

Ejemplo 5: Se nos ha pedido subir una actualización referente a la pagina web de la empresa , pero esta vez sera usando **SFTP**, para ello nos ha proporcionado los siguientes datos:

- IP del servidor remoto -> **207.249.25.60**
- Nombre del usuario remoto -> **adminlog**
- Puerto de autenticacion -> **34567**
- Ruta del Servidor a donde se tiene que subir la información **-> /tmp/Conta**
- Solución:

  1.- Para subir este directorio al servidor remoto habrá que especificar el puerto de escucha, el usuario remoto y la IP del servidor remoto

2.- Nos pedirá autenticarnos con la contraseña del usuario remoto, en este caso la contraseña del usuario **"adminlog"**



3.-Una vez autenticados con el servidor nos dará acceso a través de SFTP

```
sftp>
```

4.- Nos moveremos entre directorios con la ayuda del comando **"cd"** hasta estar ubicados en **"/tmp/Conta"**

```
sftp> cd /tmp/Conta
```

5.-Dentro de la carpeta **"Conta"** aplicar el comando "lpwd" para verificar la ruta en la cual estamos ubicados localmente

```
sftp> lpwd
Local working directory: /home/juanito
```

6.-Si no se encuentra ubicado en el directorio de trabajo indicado cámbiese de directorio mediante el comando "lcd"

```
sftp> lcd /home/juanito/datosActualizados
lcd /home/juanito/datosActualizados
```

7.- Cuando este ubicado en el directorio de trabajo que contiene la información que desea subir al servidor remoto teclee lo siguiente:

```
sftp> put datosactuales
```

El comando **"put"** tiene la funcionalidad de subir archivos desde una maquina local hasta un equipo remoto.

8.- Por ultimo teclee la palabra exit para salir del **"SFTP"**

```
sftp> exit
[root@localhost ]#
```

4.8 Evitar que nos pida autenticacion el servidor SSH

Siempre que intentemos conectarnos a un equipo remoto con SSH nos va a pedir la contraseña de acceso para asegurarse de que tenemos acceso al mismo. Hay una forma de evitar que nos pase eso siempre. Para ello hemos de generar un par de llaves RSA y DSA las cuales sirven como claves de autenticacion entre los dos equipos remotos.

4.8.1 RSA

Es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye en forma autenticada, y otra privada, la cual es guardada en secreto por su propietario.

Los mensajes enviados usando el algoritmo RSA se representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes mayores que 10100 elegidos al azar para conformar la clave de descifrado.

4.8.2 DSA (Digital Signature Algorithm)

Es un estándar del Gobierno Federal de los Estados Unidos de América para firmas digitales. Fue un Algoritmo propuesto por el Instituto Nacional de Normas y Tecnología de los Estados Unidos para su uso en su Estándar de Firma Digital. Este algoritmo como su nombre lo indica, sirve para firmar y no para cifrar información. Una desventaja de este algoritmo es que requiere mucho más tiempo de cómputo que RSA.

El proceso para generar estas claves es el siguiente:

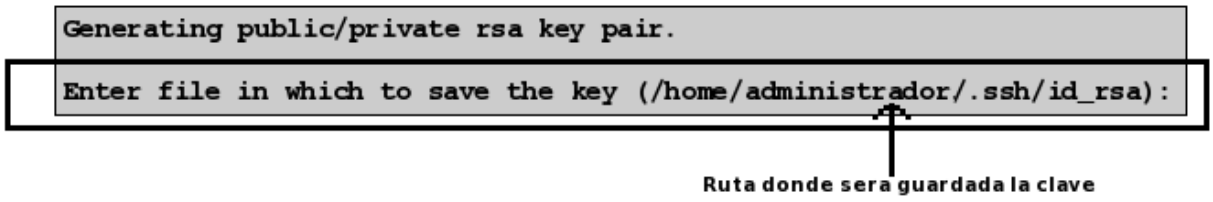
4.8.3 Generación de claves RSA

1.-Teclee el siguiente comando desde una terminal BASH

[NOTA: El comando debe ejecutarse en el equipo cliente]

```
[root@ localhost ]# ssh-keygen -t rsa
```

2.-Al haber tecleado el comando este nos preguntara si queremos guardar esa clave en otra ubicación, por defecto seleccionaremos la que nos da por defecto



3.- Al haber aceptado nos pedirá introducir una contraseña y confirmarla nuevamente

```
Enter passphrase (empty for no passphrase):xxxxxxxxxxxxx
Enter same passphrase again: xxxxxxxxxxxxxx
```

4.- Finalmente nos creara dos tipos de clave:

- Una Publica, la cual sera almacenada en la ruta:

```
/home/administrador/.ssh/id_rsa.pub
```

- Una Privada, la cual sera almacenada en la ruta:

```
/home/administrador/.ssh/id_rsa
```

Tras haber terminado de generar las claves nos tendrá que aparecer algo similar a esto

```
Your identification has been saved in /home/administrador/.ssh/id_rsa.
Your public key has been saved in /home/administrador/.ssh/id_rsa.pub.
The key fingerprint is:
c8:d1:10:62:52:1d:97:5d:7d:5a:d3:84:b5:24:48:3d administrador@localdomain
```

5.- El siguiente paso sera cambiar los permisos de ejecución del siguiente directorio

```
/home/administrador/.ssh
```

lo cual se hará de la siguiente manera:

```
[root@ localhost ]# chmod 755 /home/administrador/.ssh
```

6.- Lo siguiente sera copiar el contenido del fichero

```
/home/administrador/.ssh/id_rsa.pub
```

al fichero

```
/home/usuarioRemoto/.ssh/authorized_keys
```

del equipo remoto. Si este no existe no se preocupe, generelo con el uso del comando **"touch"** y pegue dentro de este el contenido del fichero

En caso de que el fichero

```
authorized_keys
```

exista solo pegue el contenido del fichero

```
id_rsa.pub
```

al fichero

```
authorized_keys
```

7.- El siguiente paso sera cambiar los permisos de ejecución del siguiente directorio remoto

```
/home/usuarioRemoto/.ssh/authorized_keys
```

lo cual se hará de la siguiente manera:

```
[root@ localhost ]# chmod 644 /home/usuarioRemoto/.ssh/authorized_keys
```

Con esto habremos concluido la generación de la clave RSA, ahora solo nos falta generar la clave **DSA**

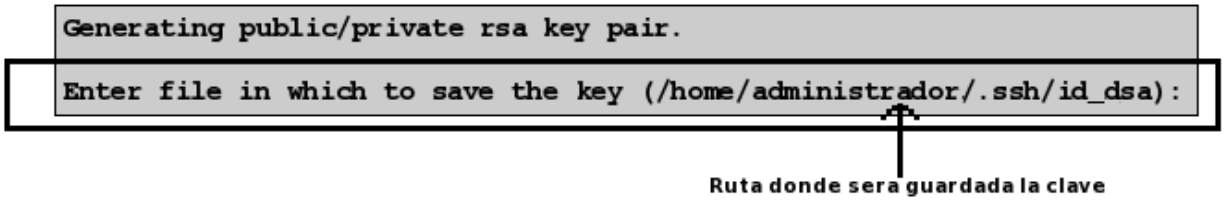
4.8.4 Generación de claves DSA

1.-Teclee el siguiente comando desde una terminal BASH

[NOTA: El comando debe ejecutarse en el equipo cliente]

```
[root@ localhost ]# ssh-keygen -t dsa
```

2.-Al haber tecleado el comando este nos preguntara si queremos guardar esa clave en otra ubicación, por defecto seleccionaremos la que nos da por defecto



3.- Al haber aceptado nos pedirá introducir una contraseña y confirmarla nuevamente

```
Enter passphrase (empty for no passphrase):xxxxxxxxxxxxxx
Enter same passphrase again: xxxxxxxxxxxxxx
```

4.- Finalmente nos creara dos tipos de clave:

- Una Publica, la cual sera almacenada en la ruta:

```
/home/usuario/.ssh/id_dsa.pub
```

- Una Privada, la cual sera almacenada en la ruta:

```
/home/usuario/.ssh/id_dsa
```

Tras haber terminado de generar las claves nos tendrá que aparecer algo similar a esto

```
Your identification has been saved in /home/administrador .ssh/id_dsa.
Your public key has been saved in /home/administrador/.ssh/id_dsa.pub.
The key fingerprint is:
5d:7d:5a:d3:84:b5:24:48:3d:c8:d1:10:62:52:1d:97: administrador@localdomain
```

5.- El siguiente paso sera cambiar los permisos de ejecución del siguiente directorio

```
/home/administrador/.ssh
```

lo cual se hará de la siguiente manera:

```
[root@ localhost ]# chmod 755 /home/administrador/.ssh
```

6.- Lo siguiente sera copiar el contenido del fichero

```
/home/usuario/.ssh/id_dsa.pub
```

al fichero

```
/home/usuarioRemoto/.ssh/authorized_keys
```

del equipo remoto. Si este no existe no se preocupe, generelo con el uso del comando **"touch"** y pegue dentro de este el contenido del fichero

En caso de que el fichero

```
authorized_keys
```

exista solo pegue el contenido del fichero

```
id_dsa.pub
```

al fichero

```
authorized_keys
```

7.- El siguiente paso sera cambiar los permisos de ejecución del siguiente directorio remoto

```
/home/usuarioRemoto/.ssh/authorized_keys
```

lo cual se hará de la siguiente manera:

```
[root@ localhost ]# chmod 644 /home/usuarioRemoto/.ssh/authorized_keys
```

Con esto habremos concluido la generación de las dos claves y de esa manera ya no tendremos que autenticarnos cada vez que nos conectemos vía SSH hacia algún equipo remoto

4.9 Montando un sistema de ficheros remoto usando sshfs y fuse

Nosotros podemos acceder a un sistema de ficheros remoto usando **sshfs** en conjunto con la aplicación **"fuse"** el cual es un comando que nos permite montar un sistema de ficheros remotos cifrados mediante la implementacion del protocolo openSSH.

De esta manera nosotros podemos acceder a los archivos remotos como si estuvieran dentro de nuestra maquina, solo debemos recordar que la conexión entre las computadoras sera un tanto lenta.

4.9.1 Sobre sshfs

sshfs (Secure Shell File System) es un sistema de ficheros de Linux que tiene como funcionalidad montar sistemas de ficheros remotos en nuestro equipo mediante la implementacion del modulo del kernel **FUSE**

Los efectos prácticos de esto es que el usuario final puede interactuar amigablemente con archivos remotos estando en un servidor SSH,viéndolos como si estuvieran en su computadora local.

4.9.2 Sobre FUSE

El sistema de archivos en espacio de usuario FUSE (Filesystem in Userspace) es un modulo cargable del kernel de linux que permite a los usuarios crear sus propios sistemas de ficheros. Esto se logra mediante la ejecución del código del sistema de archivos en el espacio de usuario, mientras que el módulo FUSE sólo proporciona un puente a la interfaz del kernel real

4.10 Instalando sshfs y fuse

Los paquetes necesarios para la instalación serán:

- **sshfs**
- **fuse-utils**

1.- Teclee la siguiente instrucción desde una terminal BASH para instalar los paquetes antes descritos:

```
[root@ localhost ]# yum install -y fuse-utils sshfs
```

2.- El siguiente paso sera crear un punto de montaje para el sistema de ficheros remoto, esto lo haremos tecleando el siguiente comando:

```
[root@ localhost ]# mkdir  /mnt/carpetaRemota
```

3.- Cambie el propietario y grupo del directorio antes creado

```
[root@ localhost ]# chown suUsuario:suGrupo /mnt/carpetaRemota
```

4.- De de alta al grupo “fuse”

```
[root@ localhost ]# groupadd fuse
```

5.- Añadase al grupo de trabajo “fuse”

```
[root@ localhost ]# usermod -G fuse suUsuario
```

6.- Por ultimo solo deberá montar el sistema de ficheros remoto, para llevar a cabo esta operación deberá ejecutarlo como root del sistema

```
[root@ localhost ]# sshfs usurioRemoto@servidorRemoto:/directorioRemoto
    /mnt/carpetaRemota
```

7.- Si se tiene especificado algún puerto de escucha para el servidor de SSH solo deberá especificare mediante el parámetro “-P”

```
[root@ localhost ]# sshfs -P[puertoDeEscucha]
usurioRemoto@servidorRemoto:/directorioRemoto
    /mnt/carpetaRemota
```

8.-Por ultimo, si quiere verificar que se encuentra montado el directorio remoto abra un navegador de archivos como Nautilus o Konqueror y con al ayuda de ellos visualice el contenido de dicho directorio.