Protocolo SSH

Ampliación de Sistemas Operativos

José Raúl López Medina

2004/2005

Contenido

0. Introducción

- **Permite a los usuarios registrarse en sistemas de host remotamente a través de la shell**
- **Encripta la sesión de registro no permite que alguien pueda obtener contraseñas no encriptadas**
- **Reemplaza a métodos menos seguros como** telnet**,** rsh **y** rcp

1. Características de SSH

- **Tipos de protección:**
  - **El cliente puede verificar que se está conectando a un mismo servidor**
  - **Información de autenticación encriptada con 128 bits**
  - **Datos enviados y recibidos encriptados con 128 bits**
  - **Posibilidad de enviar aplicaciones lanzadas desde el intérprete de comandos (reenvío por X11)**

1.1. ¿Por qué usar SSH?

- **Existen ciertas amenazas:**
  - **Intercepción de la comunicación entre dos sistemas:** un tercero en algún lugar de la red entre entidades en comunicación hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada.
  - **Personificación de un determinado host:** un sistema interceptor finge ser el receptor a quien está destinado un mensaje. Si funciona la estrategia, el cliente no se da cuenta del engaño y continúa la comunicación con el interceptor como si su mensaje hubiese llegado a su destino satisfactoriamente.

2. Versiones del protocolo SSH

- **Existen dos variedades en la actualidad:**
  - **SSH v.1:** vulnerable a un agujero de seguridad que permite, a un intruso, insertar datos en la corriente de comunicación
  - **SSH v.2:** carece de dicho agujero de seguridad (OpenSSH)

3. Secuencia de eventos de una conexión SSH

1. **Handshake encriptado para que el cliente pueda verificar la comunicación con el servidor correcto**
2. **Encriptación de la capa de transporte entre cliente y servidor mediante código simétrico**
3. **Autenticación del cliente ante el servidor**
4. **Interactuación del cliente con la máquina remota sobre la conexión encriptada**

3.1. Capa de transporte

- **Facilita una comunicación segura entre los dos hosts en el momento y después de la autenticación.**
- **Maneja la encriptación y decodificación de datos y proporciona protección de integridad de los paquetes de datos mientras son enviados y recibidos.**
- **Comprime los datos, acelerando la transmisión de información**
- **Al contactar un cliente a un servidor se producen los siguientes pasos:**
  - Intercambio de claves
  - Determinación del algoritmo de encriptación de la clave pública
  - Determinación del algoritmo de encriptación simétrica
  - Determinación del algoritmo de autenticación de mensajes
  - Determinación del algoritmo de hash que hay que usar
- **El servidor se identifica con una** clave de host **única.**
- **Después del intercambio de claves se crea un valor hash para el intercambio y un valor compartido secreto.**
- **Después de transmitir una cierta cantidad de datos con un determinado algoritmo y clave se produce otro intercambio de claves que genera otro conjunto de valores hash y un nuevo valor secreto compartido.**

3.1. Capa de transporte

```
[root@localhost /]# ssh a3144@serdis.dis.ulpgc.es
The authenticity of host 'serdis.dis.ulpgc.es (193.145.147.54)' can't be
established.
DSA key fingerprint is 47:57:1a:ea:75:d0:71:5c:24:3c:e7:9b:66:24:ff:41.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'serdis.dis.ulpgc.es,193.145.147.54' (DSA) to the
list of known hosts.
a3144@serdis.dis.ulpgc.es's password:
Last login: Wed Dec 29 2004 13:29:06
No mail.
bash: /bin/mail: Permission denied
bash$
```

3.2. Autenticación

- **Servidor informa al cliente de los métodos de autenticación soportados (firmas privadas codificadas con claves, inserción de contraseña,…) → Cliente se autenticará con cualquiera de los métodos**
- **Clientes y servidores SSH se pueden configurar para conceder varios tipos de autenticación**
- **Servidor decide qué métodos de encriptación soportará en base a su pauta de seguridad → Cliente puede elegir el orden en que intentará utilizar los métodos entre las distintas opciones**

3.3. Canales

- **Múltiples canales son abiertos mediante multiplexación**
- **Cada canal maneja la conexión para diferentes sesiones de terminal y de X11**
- **Tanto clientes como servidores pueden crear canales nuevos**
- **Soportan el** control de flujo **que les permite enviar y recibir datos ordenadamente. Los datos no se envían a través del canal sino hasta que el host haya recibido un mensaje avisando que el canal está abierto y puede recibirlos**
- **El cliente y el servidor negocian las características de cada canal automáticamente. Otorga una gran flexibilidad en el manejo de diferentes tipos de conexiones remotas sin tener que cambiar la infraestructura básica del protocolo**

4. Archivos de configuración de OpenSSH (I)

- **Tiene dos conjuntos diferentes de archivos de configuración:**
  - Archivos para programas cliente (ssh, scp y sftp)
  - Archivos para el demonio del servidor (sshd)

- **Directorio /etc/ssh/:**
  - **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:** Archivo de configuración del sistema cliente SSH por defecto que se sobreescribe si hay alguno ya presente en el directorio principal del usuario (~/.ssh/config)
  - **sshd_config:** Archivo de configuración para el demonio sshd
  - **ssh_host_dsa_key:** Clave privada DSA usada por el demonio sshd
  - **ssh_host_dsa_key.pub:** Clave pública DSA usada por el demonio sshd
  - **ssh_host_key:** Clave privada RSA usada por el demonio sshd para SSH v1
  - **ssh_host_key.pub:** Clave pública RSA usada por el demonio sshd para SSH v1
  - **ssh_host_rsa_key:** Clave privada RSA usada por el demonio sshd para SSH v2
  - **ssh_host_rsa_key.pub:** Clave pública RSA usada por el demonio sshd para SSH v2

4. Archivos de configuración de OpenSSH (II)

- **Directorio principal del usuario ~/.ssh/:**
  - **authorized_keys:** Lista de claves públicas "autorizadas". Cuando un cliente se conecta al servidor, el servidor valida al cliente chequeando su clave pública firmada almacenada dentro de este archivo
  - **id_dsa:** Clave privada DSA del usuario
  - **id_dsa.pub:** Clave pública DSA del usuario
  - **id_rsa:** Clave RSA privada usada por ssh para SSH v2
  - **id_rsa.pub:** La clave pública RSA usada por ssh para SSH v2
  - **identity:** La clave privada RSA usada por ssh para SSH v1
  - **identity.pub:** La clave pública RSA usada por ssh para SSH v1
  - **known_hosts:** Claves de host DSA de los servidores SSH accedidos por el usuario. Este archivo es muy importante para asegurarse de que el cliente SHH está conectado al servidor SSH correcto

4. Archivos de configuración OpenSSH

```
[root@localhost ~]# cd .ssh/
[root@localhost .ssh]# ls
known_hosts
[root@localhost .ssh]# less known_hosts
```

serdis.dis.ulpgc.es,193.145.147.54 ssh-dss
AAAAB3NzaC1kc3MAAACBAJlMtCEDUdfR4GqZAR7Iw07qsuTkB
5eS5/vFuciLOYraFGIaLHeG3uZWXA0WBaUYtNEBrc/TrBmi4jRv3d
hU7e79cxYIEuJMGInVlEsSRl7eRKmy97GPUnYMA54y7dlB8DhyI8
n01Vz3yGKniX7rrb8m36rtPNpQsDvFaY5gZU6/AAAAFQC0c4i5QR
+B5H9LLs8/uZCCx31lvwAAAIEAkEe04FNS+TITDEUjBIi+4ADVM
zJ1XGqgFitQh/ZmO1b2lYuvs5mJRzqBaS2SUVP+PPoe/vSiMfsALD7
rSv4fk+Y8W0DnUf+Ddap40G8sibRhly4EIs6mrUqtNlpsSS6LEzrC07
S1dY47+eiBcKxdhIfwceYFCda+CJuoqZ/NOb8AAACAdZz5EGhpnr
XCr3ssRbFyCdxdefI3JcEk7sYT7nuDrlnrDuqO6n66mXVqmfnHIRFx
PDqp4fni74PF1oXUZtJtLd+Dq4YfDvhrs+GZ7ABbcw8/DnBXhY+oR
bBWfH/hVQ9EFR2dopeh05CwPKP7jUNTTERlVv+RWHDz9AYSM
VfdfZA=

5. Más que un Shell seguro

- **Con un ancho de banda apropiado las sesiones X11 se pueden dirigir por un canal SSH**
- **Con reenvío TCP/IP se pueden asignar conexiones de puerto entre sistemas que previamente eran inseguras a canales SSH específicos**

5.1. Reenvío por X11

- **Para abrir una sesión X11 por medio de una conexión SSH basta ejecutar un programa X en una máquina local**

- **Ejemplo:**
  - **Crear una sesión segura e interactiva con up2date:**
    - **ssh sistema_remoto**
    - **up2date &**

5.2. Reenvío del puerto

- **Permite asegurar los protocolos TCP/IP a través del reenvío de puertos.**
- **Funciona mediante el mapeado de un puerto local en el cliente a un puerto remoto del servidor**
- **Creación de un canal de reenvío de puerto TCP/IP que escucha conexiones del host local:**
  - **ssh -L *local-port*:*remote-hostname*:*remote-port* *username@hostname***

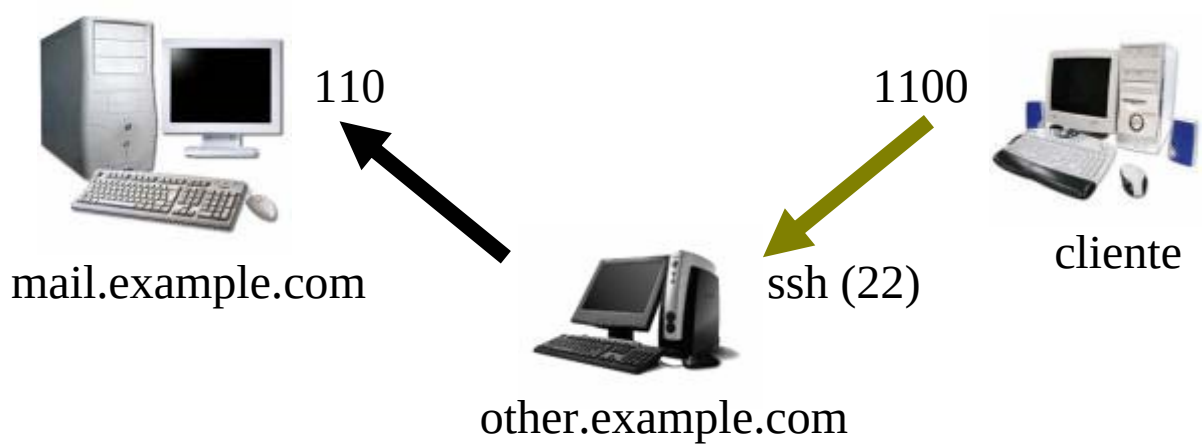
5.2. Reenvío del puerto

- **Comprobar el correo del servidor usando POP a través de una conexión encriptada:**
  - ssh -L 1100:mail.example.com:110 mail.example.com

Una vez que el canal de reenvío de puerto está entre la máquina cliente y el servidor de correo, puede direccionar su cliente de correo POP para usar el puerto 1100 en su host local para comprobar el nuevo correo. Cualquier petición enviada al puerto 1100 en el sistema cliente será dirigida de manera segura al servidor mail.example.com

5.2. Reenvío del puerto

- **Si** mail.example.com **no está ejecutando un servidor SSH, pero la otra máquina en la misma red si:**
  - ssh -L 1100:mail.example.com:110 other.example.com
- **En este ejemplo, se está reenviando su petición POP desde el puerto 1100 en la máquina cliente a través de una conexión SSH en el puerto 22 al servidor SSH,** other.example.com**. Luego,** other.example.com **se conecta al puerto 110 en** mail.example.com **para verificar nuevo correo. Observe que usando esta técnica, sólo la conexión entre el sistema cliente y el servidor SSH** other.example.com **es segura**



5.2. Reenvío del puerto

- **El reenvío del puerto se puede usar para obtener información segura a través de los firewalls de red.** Si el firewall está configurado para permitir el tráfico SSH a través del puerto estándar (22) pero bloquea el acceso a través de otros puertos, es posible todavía una conexión entre dos hosts usando los puertos bloqueados al redireccionar la comunicación sobre una conexión SSH establecida.

- El uso del reenvío de puerto permite a cualquier usuario en el sistema cliente conectarse a ese servicio. Un agresor puede también acceder los servicios reenviados**. Los administradores del sistema pueden deshabilitar esta funcionalidad en el servidor especificando** No **para la línea** AllowTcpForwarding **en** /etc/ssh/sshd_config **y reiniciando el servicio sshd.**

6. Requerir SSH para conexiones remotas

- **El uso de todos los protocolos de conexión inseguros deben ser prohibidos**

- **Servicios a deshabilitar:**
  - **telnet**
  - **rsh**
  - **ftp**
  - **rlogin**
  - **vsftpd**

- **Desactivar métodos de conexión inseguros:** chkconfig **o** ntsysv

OpenSSH

- **Implementación gratuita y de código libre de los protocolos SSH**
- **Soporta las versiones 1.3, 1.5 y 2 del protocolo SSH**
- **Automáticamente reenvía la variable DISPLAY a la máquina cliente**

Configurar un servidor OpenSSH

- **Paquetes *openssh-server* y *openssh.***
- **Archivo de configuración**
  /etc/ssh/sshd_config
- **Inicio y parada del servicio OpenSSH:**
  - /sbin/service sshd start
  - /sbin/service sshd stop

Problemas de reinstalación

- **Tras una reinstalación, aparecerá a los clientes el siguiente mensaje:**

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ 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.

- **Mantenimiento de las claves de host generadas para el sistema:**
  - /etc/ssh/ssh_host_key

Configuración de cliente OpenSSH
Uso del comando ssh

- **Conexión a una máquina remota:**
  - ssh ejemplo.servidor.es
- **Conexión a una máquina remota con otro usuario:**
  - ssh usuario@ejemplo.servidor.es
- **Ejecutar un comando en la máquina remota:**
  - ssh ejemplo.servidor.es comando
  - ssh serdis.dis.ulpgc.es ls ./

Configuración de cliente OpenSSH
Uso del comando scp

- **Permite transmitir ficheros entre máquinas sobre una conexión encriptada y segura**
- **Transferir un archivo local a un sistema remoto:**
  - scp archivo_local usuario@servidor:/archivo_remoto
- **Transferir un archivo remoto a un sistema local:**
  - scp usuario@servidor:/archivo_remoto /archivo_loal
- **Especificar múltiples archivos**
  - scp /dir_local/* usuario@servidor:/dir_remoto/

Configuración de cliente OpenSSH
Uso del comando sftp

- **Permite abrir una conexión segura interactiva de FTP**
- **Sintaxis general**
  - sftp usuario@ejemplo.sevidor.es
- **Sólo disponible a partir de la versión 2.5.0p1 de OpenSSH**

Configuración de cliente OpenSSH Generar pares de claves RSA v2

- **Sirve para no tener que introducir la contraseña cada vez que se conecte a una máquina remota**

- **Pasos para cada usuario:**

1. **Generar un par de claves RSA para trabajar con la versión 2 del protocolo:**
   - ssh-keygen –t rsa

   Aceptar la localización por defecto del archivo ~/.ssh/id_rsa
   Introducir un password distinto a la contraseña de la cuenta.
   La clave pública se escribe en ~/.ssh/id_rsa.pub
   La clave privada se haya en ~/.ssh/id_rsa

2. **Cambiar los permisos del directorio *.ssh:***
   - chmod 755 ~/.ssh

3. **Copiar el contenido de** ~/.ssh/id_rsa.pub **a** ~/.ssh/authorized_keys **en la máquina en la que se quiere conectar. Si el archivo no existe, se puede copiar el archivo** ~/.ssh/id_rsa_pub **en el archivo** ~/.ssh/authorized_keys **en la otra máquina**

Configuración de cliente OpenSSH Generar pares de claves DSA v2

1. **Para generar un par de claves DSA, se escribe el siguiente comando:**
   - ssh-keygen -t dsa

   Aceptar la localización por defecto del archivo ~/.ssh/id_dsa
   Introducir una palabra de paso diferente a la contraseña de la cuenta y confirmarla introduciéndola de nuevo
   La clave pública es escrita en ~/.ssh/id_dsa.pub
   La clave privada es escrita a ~/.ssh/ id_dsa

1. **Cambiar los permisos del directorio *.ssh* usando el comando**
   - chmod 755 ~/.ssh

2. **Copiar el contenido de** ~/.ssh/id_dsa.pub **a** ~/.ssh/authorized_keys **en la máquina en la cual quiere conectarse. Si el archivo** ~/.ssh/authorized_keys **no existe, puede copiar el archivo** ~/.ssh/id_dsa.pub **al archivo** ~/.ssh/authorized_keys **en la otra máquina.**

Configuración de cliente OpenSSH Pares de claves RSA v1.3 y v1.5

1. **Para generar un par de claves RSA se escribe el comando siguiente:**
   - ssh-keygen -t rsa1

   Aceptar la localización por defecto del archivo ~/.ssh/identity
   Introducir una palabra de paso diferente a la contraseña de la cuenta y confirmarla introduciéndola de nuevo
   La clave pública está escrita en ~/.ssh/identity.pub
   La clave privada está escrita en ~/. ssh/identity

1. **Cambiar los permisos del directorio *.ssh* y la clave con los comandos**
   - chmod 755 ~/.ssh
   - chmod 644 ~/.ssh/identity.pub

2. **Copiar los contenidos de** ~/.ssh/identity.pub **al archivo** ~/.ssh/authorized_keys **en la máquina a la cual se desea conectar. Si el archivo** ~/.ssh/authorized_keys **no existe, se puede copiarlo desde** ~/.ssh/identity.pub **al archivo** ~/.ssh/authorized_keys **en el equipo remoto.**