Curso avanzado de Linux

Rafael Varela Pet

Unidad de Sistemas
Área de Tecnologías de la Información y Comunicaciones
Universidad de Santiago de Compostela

Curso avanzado Linux

Administración remota con SSH

SSH - Introducción

- SSH: Secure SHell
- Dos versiones del protocolo: SSH1 y SSH2
- SSH1 es inseguro, sólo se debería usar en circunstancias especiales
- OpenSSH:
  - Implementación libre de SSH
  - Dos ramas de desarrollo. Una para OpenBSD y otra "portable" para diversos S.O.

OpenSSH

- Comandos básicos: ssh, scp, sftp
- Sustituyen a rlogin/telnet, rcp y ftp
- Servidor: sshd
- Otras utilidades:
  - ssh-add, sftp-server, ssh-keygen, ssh-agent y ssh-keyscan

OpenSSH. Autentificación

- keyboard-interactive: método genérico para casos en que se necesiten datos suministrados por el usuario. Métodos soportados:
  - password
  - PAM (*Pluggable Authentication Modules*)
- Autentificación con clave pública: cada usuario genera un par de claves pública/privada. Se autentican depositando su clave pública en el servidor

SSH – Clave pública

- Primero es necesario generar nuestro par de claves con el comando `ssh-keygen`
- Depositamos la clave pública en el servidor remoto, en `~/.ssh/authorized_keys`
- Disponemos de un script para automatizarlo: `ssh-copy-id`
- Desactivar autentificación con usuario/contraseña en `/etc/ssh/sshd_config`

SSH – Clave pública

- Una clave privada sin proteger puede representar un problema de seguridad
- Si cae en manos ajenas otro puede autenticarse sin necesidad de usuario/contraseña
- Más seguro usar claves cifradas:
  - Especificando una clave al llamar a `ssh-keygen`
  - Usando posteriormente `ssh-keygen -p`

OpenSSH - Agente

- **ssh-agent**: gestiona las claves privadas cuando usamos autentificación de clave pública
- **ssh-add**: incorpora identidades al agente ssh
- Inicio automático en la sesion X-Window:
  - Añadir use-ssh-agent a /etc/X11/Xsession.options

OpenSSH - Túneles X-Window

- Aplicaciones X a través del túnel SSH
- Habilitar en el servidor, en `/etc/ssh/sshd_config`
- Habilitarlo al iniciar la conexión:

```
ssh -X usuario@host
```

OpenSSH - Uso no interactivo

- Podemos lanzar comandos sin invocar una shell:

```
ssh usuario@host df -h
```

- Restricciones en ~/.ssh/authorized_keys:

```
from="host1,host2",command="/bin/df -k" KEY
user@host
```

OpenSSH. Túneles locales

- Ejemplo:

```
ssh -L2143:pop.usc.es:143 usuario@servidor
```

- Redirige el puerto local 2143 al 143 en pop.usc.es a través de **servidor**

OpenSSH -Túneles remotos

- Ejemplo:

```
ssh -R8080:www.usc.es:80 usuario@servidor
```

- Abre el puerto 8080 en **servidor**. Las conexiones a ese puerto se redirigen al puerto 80 de www.usc.es a través de la máquina local

OpenSSH - Túneles

- Al establecer el túnel también obtenemos una shell en el servidor remoto
- La opción -N permite no ejecutar nada en el servidor SSH. Ejemplo:

```
ssh -N -R8080:www.usc.es:80 usuario@servidor
```

- La opción -f hace que el proceso ssh se ejecute en segundo plano. Ejemplo:

```
ssh -f -N -R8080:www.usc.es:80 usuario@servidor
```

OpenSSH – Proxy SOCKS

- Podemos obtener más flexibilidad haciendo que OpenSSH funcione como proxy SOCKS
- Ejemplo:

```
ssh -D1080 usuario@servidor
```

VPN basada en SSH

- Permite unir dos redes de forma segura
- La opción `PermitTunnel` en `sshd_config` establece:
  - Si el servidor SSH admite esta funcionalidad
  - Qué tipo de tráfico (nivel 2 o 3)
- No es demasiado eficiente. Adecuado para montajes temporales

VPN basada en SSH

- Utilidad:
  - No hay que “tunelizar” puertos independientes
  - Permite canalizar tráfico no orientado a conexión: ICMP, UDP
- Emplea el pseudodispositivo de red `tun`
- Preparativos:

```
# aptitude install uml-utilities
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

VPN basada en SSH

- Ejemplo. Conexión punto a punto :
  - En el cliente:

```
# tunctl -t tun0
# ssh -f -w 0:1 servidor_ssh true
# ifconfig tun0 192.168.19.1 \
   pointopoint 192.168.19.2 \
   netmask 255.255.255.0
```

  - En el servidor:

```
# tunctl -t tun1
# ifconfig tun1 192.168.19.2 \
   pointopoint 192.168.19.1 \
   netmask 255.255.255.0
```

Resolución problemas

- Opción -v muestra información sobre la conexión
- Repitiendo la opción (hasta 3 veces) obtenemos más datos
  - Ejemplo:

```
ssh -v -v usuario@host
```