CONFIGURACIÓN Y PROTECCIÓN DEL SERVICIO OPENSSH

RH124

Capitulo 9

Descripción general	
Meta	Configurar acceso seguro a la línea de comandos en sistemas remotos con OpenSSH
Objetivos	• Inicie sesión en un sistema remoto usando ssh para ejecutar comandos desde el aviso de shell. • Configure ssh para permitir inicios de sesión seguros sin contraseña mediante el uso de un archivo de clave de autenticación privada. • Personalice la configuración de sshd para limitar los inicios de sesión directos como root o para deshabilitar la autenticación con contraseña.
Secciones	• Acceso a la línea de comandos remota con SSH (y práctica) • Configuración de autenticación con clave de SSH (y práctica) • Personalización de la configuración del servicio SSH (y práctica)
Trabajo de laboratorio	• Configuración y protección del servicio OpenSSH

Acceso a la línea de comandos remota con SSH

RH124

Objetivo

Tras finalizar esta sección, los estudiantes deberían poder iniciar sesión en un sistema remoto usando ssh para ejecutar comandos desde su aviso de shell.

¿Qué es OpenSSH secure shell (SSH)?

El término OpenSSH hace referencia a la implementación del software **Secure Shell** que se utiliza en el sistema. OpenSSH **Secure Shell**, **ssh**, se utiliza para ejecutar una shell en un sistema remoto de manera segura. Si tiene una cuenta de usuario en un sistema Linux remoto que proporciona los servicios SSH, **ssh** es el comando normalmente usado para iniciar sesión de manera remota en ese sistema. El comando **ssh** también se puede usar para ejecutar un comando individual en un sistema remoto.

Ejemplos de Secure Shell

Aquí le mostramos algunos ejemplos de la sintaxis del comando **ssh** para el inicio de sesión remoto y la ejecución remota:

- Cree una shell interactiva remota como el usuario actual y, luego, vuelva a su shell anterior cuando termine con el comando **exit**.

```
[student@host ~]$ ssh remotehost
student@remotehost's password:
[student@remotehost ~]$ exit
Connection to remotehost closed.
[student@host ~]$
```

- Conéctese a una shell remota como un usuario diferente (**remoteuser**) en un host seleccionado (**remotehost**):

```
[student@host ~]$ ssh remoteuser@remotehost
remoteuser@remotehost's password:
[remoteuser@remotehost ~]$
```

- Ejecute un único comando (**hostname**) en un host remoto (**remotehost**) y como usuario remoto (**remoteuser**) de manera que regrese la salida a la pantalla local:

```
[student@host ~]$ ssh remoteuser@remotehost hostname
remoteuser@remotehost's password:
remotehost.example.com
[student@host ~]$
```

El comando **w** muestra una lista de usuarios actualmente con sesión activa en el equipo. Esto es especialmente útil para mostrar qué usuarios están con sesión activa con **ssh**, desde qué ubicaciones remotas y qué están haciendo.

```
[student@host ~]$ w -f
```

```
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1     :0                Wed08    2days  1:52m  0.07s pam: gdm-passwo
root     tty6     -                 12:33    4:14m 16.27s 15.74s -bash
student  pts/0    :0.0              Wed08    5:11   1.63s  1.60s /usr/bin/gnome-
student  pts/1    :0.0              Wed08   43:44  14.48s 13.81s vim hello.c
student  pts/3    :0.0              Wed14    0.00s  0.06s  0.06s w
visitor  pts/6    server2.example.  09:22    3:14   0.02s  0.02s -bash
```

En este ejemplo, el usuario *student* inició sesión en la consola virtual 1 (**tty1**) mediante el inicio de sesión gráfico (**:0**) aproximadamente a las 8:00 del miércoles. El usuario *student* actualmente tiene tres pseudoterminales abiertos (**pts/0**, **pts/1**, and **pts/3**) iniciados mediante el entorno gráfico; estos son, casi con certeza, ventanas de terminales. En una ventana, *student* está editando **hello.c**. El usuario *root* inició sesión en la consola virtual 6, hoy a las 12:33. El usuario *visitor* inició sesión en el pseudoterminal 6 hoy a las 09:22 desde el host server2.example.com (observe que el nombre ha sido truncado), probablemente con **ssh**, y ha estado inactivo en su aviso de shell durante 3 minutos y 14 segundos.

Llaves SSH del host

SSH asegura la comunicación a través del cifrado con llave pública. Cuando un cliente **ssh** se conecta a un servidor SSH, antes de que el cliente inicie sesión, el servidor le envía una copia de su *llave pública*. Esto se utiliza con el fin de establecer el cifrado seguro para el canal de comunicación y autenticar el servidor para el cliente.

La primera vez que un usuario utiliza **ssh** para conectarse a un servidor en particular, el comando **ssh** almacena la llave pública del servidor en el archivo **~/.ssh/known_hosts** del usuario. Cada vez que el usuario se conecte nuevamente, el cliente se asegura de obtener la misma clave pública desde el servidor comparando la entrada del servidor en el archivo **~/.ssh/known_hosts** con la clave pública que envió el servidor. Si las claves *no* coinciden, el cliente supone que el tráfico de red sufre un secuestro o que el servidor está en riesgo, e interrumpe la conexión.

Esto significa que, si se cambia una clave pública del servidor (porque la clave se perdió debido a una falla en el disco duro o porque fue reemplazada por alguna razón legítima), los usuarios deberán actualizar los archivos **~/.ssh/known_hosts** para eliminar la entrada anterior y, así, poder entrar.

- Las identificaciones del host se almacenan en **~/.ssh/known_hosts** en su sistema cliente local:

```
$ cat ~/.ssh/known_hosts
remotehost,192.168.0.101 ssh-rsa AAAAB3Nzac...
```

- Las claves de host se almacenan en **/etc/ssh/ssh_host_key*** en el servidor SSH.

```
$ ls /etc/ssh/*key*
ssh_host_dsa_key          ssh_host_key          ssh_host_rsa_key
ssh_host_dsa_key.pub      ssh_host_key.pub      ssh_host_rsa_key.pub
```

nota

Un enfoque aún mejor consiste en añadir entradas haciendo coincidir los archivos **ssh_host_*key.pub** de un servidor con los del usuario **~/.ssh/known_hosts** o los de todo el sistema **/etc/ssh/ssh_known_hosts** anticipadamente cuando cambian las claves públicas. Consulte **ssh-copy-id**(1) para conocer una manera avanzada de administrar las claves ssh.

Referencias

Es posible encontrar información adicional en el capítulo sobre el uso de la utilidad ssh en la *Guía del administrador del sistema Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

> https://access.redhat.com/documentation/

Páginas del manual: **ssh**(1), **w**(1), **hostname**(1)

Práctica: Acceso remoto a la línea de comandos

Configuración de autenticación basada en llaves SSH

RH124

Objetivo

Tras finalizar esta sección, los estudiantes deberían poder configurar SSH para permitir inicios de sesión seguros sin contraseñas mediante el uso de un archivo de llave de autenticación privada.

Autenticación mediante llave SSH

Los usuarios pueden autenticar los inicios de sesión **ssh** sin una contraseña si utilizan *autenticación mediante llave pública*.**ssh** permite que los usuarios realicen la autenticación usando un esquema de llave privada y pública. Esto significa que se generan dos llaves: una privada y una pública. El archivo de llave privada se utiliza como credencial de autenticación y, al igual que una contraseña, debe ser secreta y segura. La llave pública se copia en los sistemas en los que el usuario desea iniciar sesión y se utiliza para verificar la llave privada. No es necesario que la llave pública sea secreta. Un servidor SSH que tiene llave pública puede emitir una pregunta que solo un sistema que guarde su llave privada podrá responder. En consecuencia, usted puede realizar la autenticación con la presencia de su llave. Esto le permite acceder a los sistemas sin que sea necesario escribir siempre una contraseña y, aun así, la acción sigue siendo segura.

nota

Durante la generación de claves, tiene la opción de especificar una frase de contraseña, la cual será necesaria para acceder a su clave privada. En caso de robo de la clave privada, resultará muy difícil para cualquiera que no sea el emisor usarla si está protegida con una frase de contraseña. Esto le da tiempo para crear un nuevo par de claves y quitar todas las referencias relacionadas con las anteriores, antes de que un intruso que haya decodificado la clave privada pueda utilizarla.

Siempre es recomendable proteger la clave privada con una frase contraseña, ya que la clave le permite acceder a otras máquinas. Sin embargo, esto significa que deberá escribir su frase de contraseña cada vez que utilice la clave, de manera que el proceso de autenticación deja de ser sin contraseña. Esto puede evitarse utilizando **ssh-agent**, al que se le puede dar la frase de contraseña una vez al comienzo de la sesión (mediante **ssh-add**), de modo que la pueda proporcionar cuando sea necesario mientras mantenga la sesión iniciada.

Para obtener información adicional sobre el comando **ssh-agent**, consulte la Guía de administración de Red Hat System, capítulo 8.2.4.2.: Configuración de ssh-agent.

Una vez que se hayan generado las claves SSH, se guardarán de modo predeterminado en el directorio **.ssh/** de su directorio principal. Los permisos deben ser 600 en la clave privada y 644 en la clave pública.

Para poder usar la autenticación mediante claves, la clave pública debe copiarse en el sistema de destino. Esto puede realizarse con **ssh-copy-id**.

```
[student@desktopX ~]$ ssh-copy-id root@desktopY
```

Al copiar la clave en otro sistema mediante **ssh-copy-id**, este copiará el archivo **~/.ssh/id_rsa.pub** de forma predeterminada.

Demostración de claves SSH

- Utilice **ssh-keygen** para crear un par de claves públicas y privadas.

```
[student@desktopX ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): Enter
Created directory '/home/student/.ssh'.
Enter passphrase (empty for no passphrase): redhat
Enter same passphrase again: redhat
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
a4:49:cf:fb:ac:ab:c8:ce:45:33:f2:ad:69:7b:d2:5a student@desktopX.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|      . .        |
|     . *         |
|    . * S        |
|     + + .       |
|      o.E        |
|   o oo+oo       |
|   .=.**ooo      |
+-----------------+
```

- Utilice **ssh-copy-id** para copiar la clave pública en la ubicación correcta en un sistema remoto. Por ejemplo:

```
[student@desktopX ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub root@serverX.example.com
```

Referencias

Es posible encontrar información adicional en el capítulo sobre el uso de autenticación mediante claves en la *Guía del administrador del sistema Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

> https://access.redhat.com/documentation/

Páginas del manual: **ssh-keygen**(1), **ssh-copy-id**(1), **ssh-agent**(1), **ssh-add**(1)

Práctica: Uso de la autenticación mediante claves SSH

Personalización de la configuración del servicio SSH

RH124

Objetivo

Tras finalizar esta sección, los estudiantes deberían poder personalizar la configuración de sshd para restringir los inicios de sesión directos como root o para inhabilitar la autenticación con contraseña.

Archivo de configuración del servidor OpenSSH

Si bien la configuración del servidor OpenSSH no suele requerir modificación, hay medidas adicionales de seguridad disponibles.

Pueden modificarse varios aspectos del servidor OpenSSH en el archivo de configuración **`/etc/ssh/sshd_config`**.

Prohibir al usuario root el inicio de sesión con SSH

Desde el punto de vista de la seguridad, es aconsejable prohibir al usuario root que inicie sesión en el sistema en forma directa con **ssh**.

- El nombre de usuario root existe en cada sistema Linux de manera predeterminada; por lo tanto, un posible atacante solo tiene que adivinar la contraseña en lugar de la combinación de nombre de usuario y contraseña válidos.

- El usuario root tiene privilegios sin restricciones.

El servidor OpenSSH tiene un parámetro de archivo de configuración interno para prohibir el inicio de sesión en el sistema como usuario root, que es comentado de manera predeterminada en el archivo **/etc/ssh/sshd_config**:

```
#PermitRootLogin yes
```

Si se habilita la opción anterior en el archivo de configuración **/etc/ssh/sshd_config** de la siguiente manera, el usuario root no podrá iniciar sesión en el sistema con el comando **ssh** después de que se haya reiniciado el servicio sshd:

```
PermitRootLogin no
```

El servicio sshd tiene que reiniciarse para que puedan implementarse los cambios:

```
[root@serverX ~]# systemctl sshd
```

Prohibir la autenticación de contraseña con SSH

El solo hecho de permitir el inicio de sesión mediante clave a la línea de comando remota tiene varias ventajas:

- Las claves SSH son más extensas que una contraseña estándar y este detalle aporta seguridad.
- El inicio del acceso a la shell remota implica menos esfuerzo después de la configuración inicial.

Existe una opción en el archivo de configuración **`/etc/ssh/sshd_config`** que activa una autenticación por contraseña de manera predeterminada:

```
PasswordAuthentication yes
```

Para evitar la autenticación de contraseña, la opción **`PasswordAuthentication`** tiene que configurarse en **`no`** y es necesario reiniciar el servicio sshd:

```
PasswordAuthentication no
```

Recuerde que cada vez que cambie el archivo **`/etc/ssh/sshd_config`**, debe reiniciarse el servicio sshd:

```
[root@serverX ~]# systemctl reload sshd
```

Referencias

Páginas del manual: **ssh**(1), **sshd_config**(5)

Práctica: Restricción de inicios de sesión en SSH