Gestión de acceso mediante shell seguro en Oracle® Solaris 11.2

Referencia: E53963-02
Septiembre de 2014

ORACLE®

Contenido

Uso de esta documentación

Gestión de acceso mediante shell seguro en Oracle® Solaris 11.2 explica cómo administrar y utilizar la función de shell seguro para el acceso remoto seguro.

- **Descripción general**: describe los conceptos y las tareas acerca del uso del shell seguro en Oracle Solaris.
- **Destinatarios**: los administradores del sistema que deben implementar la seguridad en la empresa.
- **Conocimiento requerido**: estar familiarizado con terminología y conceptos de seguridad.

Biblioteca de documentación del producto

En la biblioteca de documentación (http://www.oracle.com/pls/topic/lookup?ctx=E56339), se incluye información de última hora y problemas conocidos para este producto.

Acceso a My Oracle Support

Los clientes de Oracle disponen de asistencia a través de Internet en el portal My Oracle Support. Para obtener más información, visite http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info o, si tiene alguna discapacidad auditiva, visite http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs.

Comentarios

Envíenos comentarios acerca de esta documentación mediante http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CAPÍTULO 1

Uso de shell seguro (tareas)

La función Shell seguro de Oracle Solaris proporciona acceso seguro a un host remoto por medio de una red no segura. El shell proporciona comandos para el inicio de sesión remoto, la visualización remota de ventanas y la transferencia de archivos remota. En este capítulo, se tratan los siguientes temas:

Para obtener información de referencia, consulte Capítulo 2, Referencia de shell seguro.

Shell seguro (descripción general)

Shell seguro es el protocolo predeterminado de acceso remoto en un sistema Oracle Solaris recientemente instalado. Shell seguro en Oracle Solaris se crea a partir del Kit de herramientas de código abierto, OpenSSL, que implementa la capa de sockets seguros y la seguridad de capa de transporte.

Dos versiones distintas del kit de herramientas están disponibles en Oracle Solaris.

- La versión 1.0.0 es la versión predeterminada donde se ejecuta Shell seguro.
- La versión 0.9.8 implementa FIPS-140FIPS 140,un estándar de seguridad informática del Gobierno de los EE. UU. para módulos de cifrado.

  Para obtener información acerca de cómo usar Shell seguro en el modo FIPS 140, consulte "Shell seguro y FIPS 140" [11].

En Shell seguro, la autenticación la proporciona el uso de contraseñas, claves públicas o ambas. Todo el tráfico de la red está cifrado. Por lo tanto, Shell seguro impide que un posible intruso pueda leer una comunicación interceptada. Shell seguro también impide que un adversario falsifique el sistema.

Shell seguro también puede utilizarse como una red privada virtual a petición. Una VPN puede reenviar tráfico de sistemas de ventanas X o puede conectar números de puerto individuales entre los equipos locales y remotos mediante un enlace de red cifrado.

Con Shell seguro, puede realizar estas acciones:

- Iniciar sesión en otro host de forma segura por medio de una red no segura.
- Copiar archivos de forma segura entre los dos hosts.
- Ejecutar comandos de forma segura en el host remoto.

En el lado del servidor, Shell seguro admite la versión 2 (v2) del protocolo de Shell seguro. En el cliente, además de v2, se admite la versión 1 (v1).

Autenticación de shell seguro

Shell seguro proporciona métodos de clave pública y contraseña para autenticar la conexión al host remoto. La autenticación de clave pública es un mecanismo de autenticación más potente que la autenticación de contraseña, porque la clave privada nunca viaja por medio de la red.

Los métodos de autenticación se prueban en el siguiente orden: cuando la configuración no satisface un método de autenticación, se prueba el siguiente método.

- **GSS-API**: utiliza credenciales para mecanismos GSS-API, como `mech_krb5` (Kerberos V) y `mech_dh` (AUTH_DH), para autenticar clientes y servidores. Para obtener más información sobre GSS-API, consulte "Introduction to GSS-API" de "Developer's Guide to Oracle Solaris 11 Security ".
- **Autenticación basada en host**: utiliza claves de host y archivos `rhosts`. Utiliza las claves de host públicas y privadas RSA y DSA del cliente para autenticar el cliente. Utiliza los archivos `rhosts` para autorizar clientes a usuarios.
- **Autenticación de clave pública**: autentica a los usuarios con sus claves públicas y privadas RSA y DSA.
- **Autenticación de contraseña**: utiliza PAM para autenticar usuarios. El método de autenticación de teclado en v2 permite la solicitud arbitraria por PAM. Para obtener más información, consulte la sección `SECURITY` en la página del comando man `sshd`(1M).

En la siguiente tabla, se muestran los requisitos para autenticar a un usuario que está intentando iniciar sesión en un host remoto. El usuario está en el host local, el cliente. El host remoto, el servidor, está ejecutando el daemon `sshd`. En la tabla, se muestran los métodos de autenticación de Shell seguro y los requisitos de host.

TABLA 1-1 Métodos de autenticación para shell seguro

Método de autenticación	Requisitos de host local (cliente)	Requisitos de host remoto (servidor)
GSS-API	Credenciales de iniciador para el mecanismo GSS	Credenciales de aceptador para el mecanismo GSS Para obtener más información, consulte

Método de autenticación	Requisitos de host local (cliente)	Requisitos de host remoto (servidor)
		"Adquirir credenciales GSS en shell seguro" [34].

Basado en host	Cuenta de usuario
	
	Clave privada de host local en `/etc/ssh/ssh_host_rsa_key` o `/etc/ssh/ssh_host_dsa_key`
	
	`HostbasedAuthentication yes` en `/etc/ssh/ssh_config`
		Cuenta de usuario
		
		Clave pública de host local en `/etc/ssh/known_hosts` o `~/.ssh/known_hosts`
		
		`HostbasedAuthentication yes` en `/etc/ssh/sshd_config`
		
		`IgnoreRhosts no` en `/etc/ssh/sshd_config`
		
		Entrada de host local en `/etc/ssh/shosts.equiv`, `/etc/hosts.equiv`, `~/.rhosts` o `~/.shosts`

Basado en contraseña	Cuenta de usuario	Cuenta de usuario Admite PAM.
`.rhosts` con RSA (v1) en el servidor solamente	Cuenta de usuario Clave pública de host local en `/etc/ssh/ssh_host_rsa1_key`	Cuenta de usuario Clave pública de host local en `/etc/ssh/ssh_known_hosts` o `~/.ssh/known_hosts` `IgnoreRhosts no` en `/etc/ssh/sshd_config` Entrada de host local en `/etc/ssh/shosts.equiv`, `/etc/hosts.equiv`, `~/.shosts` o `~/.rhosts`
Clave pública RSA o DSA	Cuenta de usuario Clave privada en `~/.ssh/id_rsa` o `~/.ssh/id_dsa` Clave pública del usuario en `~/.ssh/id_rsa.pub` o `~/.ssh/id_dsa.pub`	Cuenta de usuario Clave pública del usuario en `~/.ssh/authorized_keys`

Shell seguro y el proyecto OpenSSH

El shell seguro es una bifurcación del proyecto OpenSSH (http://www.openssh.com). Las correcciones de seguridad para las vulnerabilidades que se detectan en versiones posteriores de OpenSSH se integran en el shell seguro, ya que son funciones y correcciones de errores individuales. A partir de septiembre de 2012, la versión de shell seguro de Oracle Solaris es 2.0. El comando `ssh -V` muestra el número de versión.

Las siguientes funciones se han implementado para el protocolo v2 en esta versión de shell seguro:

- Palabra clave `ForceCommand`: fuerza la ejecución de los comandos especificados independientemente de lo que escriba el usuario en la línea de comandos. Esta palabra clave es muy útil dentro de un bloque `Match`. Esta opción de configuración `sshd_config` es similar a la opción `command="..."` en `$HOME/.ssh/authorized_keys`.
- Protección de frase de contraseña `AES-128`: las claves privadas generadas por el comando `ssh-keygen` están protegidas con el algoritmo `AES-128`. Este algoritmo protege las claves recientemente generadas y las claves que se volvieron a cifrar, por ejemplo, cuando se cambia la frase de contraseña.
- Opción `-u` para el comando `sftp-server`: permite al usuario establecer una `umask` explícita en archivos y directorios. Esta opción sustituye la `umask` predeterminada del usuario. Para ver un ejemplo, consulte la descripción de `Subsystem` en la página del comando man `sshd_config`(4).
- Palabras clave adicionales para bloques `Match`: `AuthorizedKeysFile`, `ForceCommand` y `HostbasedUsesNameFromPacketOnly` se admiten dentro de bloques `Match`. De manera predeterminada, el valor de `AuthorizedKeysFile` es `$HOME/.ssh/authorized_keys` y de `HostbasedUsesNameFromPacketOnly` es `no`. Para utilizar bloques `Match`, consulte Cómo crear excepciones de host y usuario para valores predeterminados del shell seguro [16].

Los ingenieros de Oracle Solaris proporcionan correcciones de errores para el proyecto OpenSSH. Además, han integrado las siguientes funciones de Oracle Solaris en la bifurcación de shell seguro:

- PAM: el shell seguro utiliza PAM. La opción de configuración `UsePAM` de OpenSSH no se admite.
- Separación de privilegios: el shell seguro no utiliza el código de separación de privilegios del proyecto OpenSSH. El shell seguro separa el procesamiento de auditoría, conservación de registros y restablecimiento de claves del procesamiento de protocolos de sesión.

  El código de separación de privilegios del shell seguro siempre está activado y no se puede desactivar. La opción `UsePrivilegeSeparation` de OpenSSH no se admite.
- Configuración regional: el shell seguro admite completamente la negociación de idiomas, como se define en RFC 4253, *Protocolo de transferencia de shell seguro*. Después de que el usuario inicia sesión, el perfil del shell de inicio de sesión del usuario puede sustituir la configuración regional negociada del shell seguro.
- Auditoría: el shell seguro está totalmente integrado en el servicio de auditoría de Oracle Solaris. Para obtener información acerca del servicio de auditoría, consulte “Gestión de auditoría en Oracle Solaris 11.2 ”.
- Compatibilidad con GSS-API: la GSS-API se puede utilizar para la autenticación de usuario *y* para el intercambio de claves inicial. La GSS-API se define en RFC4462, *Generic Security Service Application Program Interface* (Interfaz de programa de aplicación de servicios de seguridad genéricos).
- Comandos de proxy: el shell seguro proporciona comandos de proxy para protocolos SOCKS5 y HTTP. Para ver un ejemplo, consulte Cómo configurar conexiones de shell seguro predeterminadas a hosts fuera de un firewall [29].

En las versiones de Oracle Solaris, el shell seguro resincroniza el indicador de compatibilidad `SSH_OLD_FORWARD_ADDR` del proyecto OpenSSH.

Shell seguro y FIPS 140

Shell seguro es consumidor del módulo FIPS 140 de OpenSSL. Oracle Solaris ofrece una opción de FIPS 140 para el lado del servidor y el lado del cliente. Para cumplir con los requisitos de FIPS 140, los administradores deben configurar y utilizar las opciones de FIPS 140.

El modo FIPS, donde Shell seguro utiliza el modo FIPS 140 de OpenSSL, no es el predeterminado. Como administrador, debe activar explícitamente el shell seguro para ejecutar en el modo FIPS 140. Puede invocar el modo FIPS 140 con el comando `ssh -o "UseFIPS140 yes"` *remote-host*. Como alternativa, se puede definir una palabra clave en los archivos de configuración.

Brevemente, la implementación consta de los siguientes elementos:

- Los siguientes cifrados aprobados por FIPS 140 están disponibles en el servidor y en el cliente: `aes128-cbc`, `aes192-cbc` y `aes256-cbc`.

  `3des-cbc` está disponible de manera predeterminada en el cliente, pero no está en la lista de cifrado del servidor debido a posibles riesgos de seguridad.
- Están disponibles los siguientes códigos de autenticación de mensajes (MAC) aprobados por FIPS 140:
  - `hmac-sha1`, `hmac-sha1-96`
  - `hmac-sha2-256`, `hmac-sha2-256-96`
  - `hmac-sha2-512`, `hmac-sha2-512-96`
- Se admiten cuatro configuraciones de servidor-cliente:
  - Ningún modo FIPS 140 en el cliente o el servidor
  - Modo FIPS 140 en el cliente y en el servidor
  - Modo FIPS 140 en el servidor, pero no en el cliente
  - Ningún modo FIPS 140 en el servidor, pero sí en el cliente
- El comando `ssh-keygen` tiene una opción para generar la clave privada del usuario en el formato PKCS nº 8 que requieren los clientes Shell seguro en modo FIPS. Para obtener más información, consulte la página del comando man `ssh-keygen`(1).

Para obtener más información acerca de FIPS 140, consulte "Using a FIPS 140 Enabled System in Oracle Solaris 11.2 ". Consulte también las páginas del comando man `sshd`(1M), `sshd_config`(4), `ssh`(1) y `ssh_config`(4).

Cuando utiliza una tarjeta Sun Crypto Accelerator 6000 para operaciones Shell seguro, Shell seguro se ejecuta con compatibilidad de FIPS 140 en el nivel 3. El hardware de nivel 3 está

certificado para evitar la alteración física, utilizar autenticación basada en identidad y aislar las interfaces que gestionan parámetros de seguridad críticos de otras interfaces de hardware.

Configuración de shell seguro (tareas)

Shell seguro se configura durante la instalación. Para cambiar los valores predeterminados, se requiere la intervención del administrador. Las siguientes tareas muestran cómo cambiar algunos de los valores predeterminados.

Configuración de shell seguro (mapa de tareas)

En el siguiente mapa de tareas, se indican procedimientos para configurar Shell seguro. Para utilizar Shell seguro, consulte “Uso de shell seguro (tareas)” [19].

Tarea	Descripción	Para obtener instrucciones
Configurar autenticación basada en host.	Configura la autenticación basada en host en el cliente y el servidor.	Cómo configurar la autenticación basada en host para el shell seguro [12]
Aumentar tamaño del búfer para controlar la latencia de conexión.	Genera el valor de la propiedad TCP `recv_buf` para ancho de banda alto y redes de latencia alta.	“Cambio del tamaño de la memoria intermedia de recepción de TCP” de “Administración de redes TCP/IP, IPMP y túneles IP en Oracle Solaris 11.2 ”
Configurar reenvío del puerto.	Permite a los usuarios utilizar el reenvío del puerto.	Cómo configurar el reenvío del puerto en el shell seguro [16]
Configurar excepciones para valores predeterminados del sistema Shell seguro.	Para usuarios, hosts, grupos y direcciones, especifica valores de Shell seguro diferentes de los valores predeterminados del sistema.	Cómo crear excepciones de host y usuario para valores predeterminados del shell seguro [16]
Aislar un entorno `root` para transferencias `sftp`.	Proporciona un directorio protegido para las transferencias de archivos.	Cómo crear un directorio aislado para archivos `sftp` [18]

▼ Cómo configurar la autenticación basada en host para el shell seguro

El siguiente procedimiento configura un sistema de clave pública en el que la clave pública del cliente se utiliza para la autenticación en el servidor. El usuario también debe crear un par de clave pública y clave privada.

En el procedimiento, los términos *cliente* y *host local* hacen referencia al sistema en el que un usuario introduce el comando `ssh`. Los términos *servidor* y *host remoto* hacen referencia al sistema al que el cliente está intentando acceder.

Antes de empezar Debe asumir el rol `root`. Para obtener más información, consulte "Uso de sus derechos administrativos asignados" de "Protección de los usuarios y los procesos en Oracle Solaris 11.2 ".

1. **En el cliente, active la autenticación basada en host.**

   En el archivo de configuración del cliente, `/etc/ssh/ssh_config`, escriba la siguiente entrada:

   ```
   HostbasedAuthentication yes
   ```

   Para ver la sintaxis del archivo, consulte la página del comando man `ssh_config`(4).

2. **En el servidor, active la autenticación basada en host.**

   En el archivo de configuración del servidor, `/etc/ssh/sshd_config`, escriba la misma entrada:

   ```
   HostbasedAuthentication yes
   ```

   Para ver la sintaxis del archivo, consulte la página del comando man `sshd_config`(4).

3. **En el servidor, configure un archivo que permita que el cliente se reconozca como un host de confianza.**

   Para obtener más información, consulte la sección `FILES` de la página del comando man `sshd`(1M).

   - **Si usted realiza la configuración, agregue el cliente como una entrada al archivo `/etc/ssh/shosts.equiv` del servidor.**

     client-host

   - **Si los usuarios realizan la configuración, deben agregar una entrada para el cliente a su archivo `~/.shosts` en el servidor.**

     client-host

4. **En el servidor, asegúrese de que el daemon `sshd` pueda acceder a la lista de hosts de confianza.**

   Establezca `IgnoreRhosts` en `no` en el archivo `/etc/ssh/sshd_config`.

   ```
   ## sshd_config
   IgnoreRhosts no
   ```

5. **Asegúrese de que los usuarios de Shell seguro en su sitio tengan cuentas en ambos hosts.**

6. **Coloque la clave pública del cliente en el servidor con uno de los siguientes métodos:**

   - **Modifique el archivo `sshd_config` en el servidor y luego indique a sus usuarios que agreguen las claves de host públicas del cliente a sus archivos `~/.ssh/known_hosts`.**

     ```
     ## sshd_config
     IgnoreUserKnownHosts no
     ```

     Para obtener instrucciones para el usuario, consulte Cómo generar un par de clave pública y clave privada para utilizar con el shell seguro [20].

   - **Copie la clave pública del cliente en el servidor.**

     Las claves de host se almacenan en el directorio `/etc/ssh`. Las claves suelen ser generadas por el daemon `sshd` al iniciar por primera vez.

     a. **Agregue la clave al archivo `/etc/ssh/ssh_known_hosts` en el servidor.**

        En el cliente, escriba el siguiente comando en una línea sin barra diagonal inversa.

        ```
        # cat /etc/ssh/ssh_host_dsa_key.pub | ssh RemoteHost \
        'cat >> /etc/ssh/ssh_known_hosts && echo "Host key copied"'
        ```

Nota - Si faltan las claves de host del servidor, con el shell seguro se genera un mensaje de error similar al siguiente:

```
Client and server could not agree on a key exchange algorithm:
client "diffie-hellman-group-exchange-sha256,diffie-hellman-group-
exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1",
server "gss-group1-sha1-toWM5Slw5Ew8Mqkay+al2g==". Make sure host keys
are present and accessible by the server process. See sshd_config(4)
description of "HostKey" option.
```

---

     b. **Cuando se le pida, proporcione la contraseña de inicio de sesión.**

        Cuando el archivo se copia, se muestra el mensaje “Host key copied” (clave de host copiada).

        Cada línea en el archivo `/etc/ssh/ssh_known_hosts` consta de campos que están separados por espacios:

        hostnames algorithm-name publickey comment

     c. **Edite el archivo `/etc/ssh/ssh_known_hosts` y agregue *RemoteHost* como el primer campo en la entrada copiada.**

        ```
        ## /etc/ssh/ssh_known_hosts File
        ```

RemoteHost *<copied entry>*

ejemplo 1-1 Configuración de autenticación basada en host

En el siguiente ejemplo, cada host está configurado como servidor y como cliente. Un usuario en cualquiera de los hosts puede iniciar una conexión `ssh` al otro host. La siguiente configuración hace que cada host sea un servidor y un cliente:

- En cada host, los archivos de configuración de Shell seguro contienen las siguientes entradas:

```
## /etc/ssh/ssh_config
HostBasedAuthentication yes
#
## /etc/ssh/sshd_config
HostBasedAuthentication yes
IgnoreRhosts no
```

- En cada host, el archivo `shosts.equiv` contiene una entrada para el otro host:

```
## /etc/ssh/shosts.equiv on machine2
machine1

## /etc/ssh/shosts.equiv on machine1
machine2
```

- La clave pública de cada host está en el archivo `/etc/ssh/ssh_known_hosts` del otro host:

```
## /etc/ssh/ssh_known_hosts on machine2
… machine1

## /etc/ssh/ssh_known_hosts on machine1
… machine2
```

- Los usuarios tienen una cuenta en ambos hosts. Por ejemplo, la siguiente información aparecerá para el usuario John Doe:

```
## /etc/passwd on machine1
jdoe:x:3111:10:J Doe:/home/jdoe:/bin/sh

## /etc/passwd on machine2
jdoe:x:3111:10:J Doe:/home/jdoe:/bin/sh
```

Cómo configurar el reenvío del puerto en el shell seguro

El reenvío del puerto permite que un puerto local sea reenviado a un host remoto. Efectivamente, un socket se asigna para escuchar el puerto en el lado local. De forma similar, un puerto se puede especificar en el lado remoto.

Nota - El reenvío del puerto de Shell seguro debe utilizar conexiones TCP. Shell seguro no admite conexiones UDP para el reenvío del puerto.

Antes de empezar Debe asumir el rol `root`. Para obtener más información, consulte "Uso de sus derechos administrativos asignados" de "Protección de los usuarios y los procesos en Oracle Solaris 11.2 ".

1. **Configure un valor de Shell seguro en el servidor remoto para permitir el reenvío del puerto.**

   Cambie el valor de `AllowTcpForwarding` a `yes` en el archivo `/etc/ssh/sshd_config`.

   ```
   # Port forwarding
   AllowTcpForwarding yes
   ```

2. **Reinicie el servicio de Shell seguro.**

   ```
   remoteHost# svcadm restart network/ssh:default
   ```

   Para obtener información acerca de la gestión de servicios persistentes, consulte Capítulo 1, "Introducción a la Utilidad de gestión de servicios" de "Gestión de los servicios del sistema en Oracle Solaris 11.2 " y la página del comando man `svcadm`(1M).

3. **Verifique que el reenvío del puerto se pueda utilizar.**

   ```
   remoteHost# /usr/bin/pgrep -lf sshd
   1296 ssh -L 2001:remoteHost:23 remoteHost
   ```

Cómo crear excepciones de host y usuario para valores predeterminados del shell seguro

Este procedimiento agrega un bloque `Match` condicional después de la sección global del archivo `/etc/ssh/sshd_config`. Los pares de valores de palabras clave a continuación del bloque `Match` especifican excepciones para el usuario, grupo, host o dirección que se especifica como coincidencia.

Antes de empezar Debe convertirse en un administrador con la autorización `solaris.admin.edit/etc/ssh/sshd_config` asignada. De manera predeterminada, el rol `root` tiene esta autorización. Para obtener más información, consulte "Uso de sus derechos administrativos asignados" de "Protección de los usuarios y los procesos en Oracle Solaris 11.2 ".

1. **Abra el archivo `/etc/ssh/sshd_config` para modificarlo.**

   ```
   # pfedit /etc/ssh/sshd_config
   ```

2. **Configure un usuario, grupo, host o dirección para que utilice una configuración de Shell seguro diferente de la configuración predeterminada.**

   Coloque los bloques `Match` después de la configuración global.

   Nota - La sección global del archivo puede no siempre mostrar la configuración predeterminada. Para obtener los valores predeterminados, consulte la página del comando man `sshd_config`(4).

   Por ejemplo, es posible que tenga usuarios que no deberían poder utilizar el reenvío TCP. En el siguiente ejemplo, cualquier usuario en el grupo `public` y cualquier nombre de usuario que comienza con `test` no puede utilizar el reenvío TCP:

   ```
   ## sshd_config file
   ## Global settings

   # Example (reflects default settings):
   #
   # Host *
   #   ForwardAgent no
   #   ForwardX11 no
   #   PubkeyAuthentication yes
   #   PasswordAuthentication yes
   #   FallBackToRsh no
   #   UseRsh no
   #   BatchMode no
   #   CheckHostIP yes
   #   StrictHostKeyChecking ask
   #   EscapeChar ~
   Match Group public
   AllowTcpForwarding no
   Match User test*
   AllowTcpForwarding no
   ```

   Para obtener más información sobre la sintaxis del bloque `Match`, consulte la página del comando man `sshd_config`(4).

▼ Cómo crear un directorio aislado para archivos `sftp`

Este procedimiento configura un directorio `sftponly` creado específicamente para transferencias `sftp`. Los usuarios no pueden ver los archivos o directorios fuera del directorio de transferencia.

Antes de empezar Debe asumir el rol `root`. Para obtener más información, consulte "Uso de sus derechos administrativos asignados" de "Protección de los usuarios y los procesos en Oracle Solaris 11.2 ".

1. **En el servidor Shell seguro cree el directorio aislado como un entorno `chroot`.**

```
# groupadd sftp
# useradd -m -G sftp -s /bin/false sftponly
# chown root:root /export/home/sftponly
# mkdir /export/home/sftponly/WWW
# chown sftponly:staff /export/home/sftponly/WWW
```

   En esta configuración, `/export/home/sftonly` es el directorio `chroot` al que sólo tiene acceso la cuenta `root`. El usuario tiene permiso de escritura en el subdirectorio `sftponly/WWW`.

2. **Todavía en el servidor, configure un bloque de coincidencia para el grupo `sftp`.**
   En el archivo `/etc/ssh/sshd_config`, busque la entrada `sftp subsystem` y modifique el archivo de la siguiente forma:

```
# pfedit /etc/ssh/sshd_config
...
# sftp subsystem
#Subsystem      sftp    /usr/lib/ssh/sftp-server
Subsystem       sftp    internal-sftp
...
## Match Group for Subsystem
## At end of file, to follow all global options
Match Group sftp
ChrootDirectory %h
ForceCommand internal-sftp
AllowTcpForwarding no
```

   Puede usar las siguientes variables para especificar la ruta `chroot`:

   - `%h`: especifica el directorio raíz.
   - `%u`: especifica el nombre de usuario del usuario autenticado.
   - `%%`: omite el signo `%`.

3. **En el cliente, compruebe que la configuración funcione correctamente.**
   Los archivos en el entorno `chroot` pueden ser diferentes.

```
root@client:~# ssh sftponly@server
This service allows sftp connections only.
Connection to server closed.       No shell access, sftp is enforced.
root@client:~# sftp sftponly@server
sftp> pwd       sftp access granted
Remote working directory: /      chroot directory looks like root directory
sftp> ls
WWW             local.cshrc     local.login     local.profile
sftp> get local.cshrc
Fetching /local.cshrc to local.cshrc
/local.cshrc    100%  166     0.2KB/s   00:00     user can read contents
sftp> put /etc/motd
Uploading /etc/motd to /motd
Couldn't get handle: Permission denied     user cannot write to / directory
sftp> cd WWW
sftp> put /etc/motd
Uploading /etc/motd to /WWW/motd
/etc/motd     100%  118     0.1KB/s   00:00     user can write to WWW directory
sftp> ls -l
-rw-r--r--    1 101  10    118 Jul 20 09:07 motd     successful transfer
sftp>
```

Uso de shell seguro (tareas)

En esta sección, se proporcionan procedimientos para familiarizar a los usuarios con Shell seguro.

Uso de shell seguro (mapa de tareas)

En el siguiente mapa de tareas, se indican procedimientos de usuario para usar Shell seguro.

Tarea	Descripción	Para obtener instrucciones
Crear un par de clave pública y clave privada.	Permite el acceso a Shell seguro para sitios que requieren la autenticación de clave pública.	Cómo generar un par de clave pública y clave privada para utilizar con el shell seguro [20]
Cambiar la frase de contraseña.	Cambia la frase que autentica la clave privada.	Cómo cambiar la frase de contraseña de una clave privada de shell seguro [22]
Iniciar sesión con Shell seguro.	Proporciona comunicación de Shell seguro cifrada cuando se inicia sesión de manera remota.	Cómo iniciar sesión en un host remoto con shell seguro [22]
Iniciar sesión en Shell seguro sin que se le solicite una contraseña.	Permite iniciar sesión mediante un agente que proporciona la contraseña a Shell seguro.	Cómo reducir indicadores de contraseñas en el shell seguro [24]

Tarea	Descripción	Para obtener instrucciones
Conectarse a Shell seguro como `root`.	Permite iniciar sesión como `root` para los comandos `send` y `receive` de ZFS.	Cómo administrar ZFS con shell seguro de forma remota [25]
Utilizar el reenvío del puerto en Shell seguro.	Especifica un puerto local o un puerto remoto que se utilizará en una conexión de Shell seguro por TCP.	Cómo utilizar el reenvío del puerto en el shell seguro [27]
Copiar archivos con Shell seguro.	Copia archivos entre hosts de manera segura.	Cómo copiar archivos con shell seguro [28]
Conectarse de forma segura de un host dentro de un cortafuegos a un host fuera del cortafuegos.	Utiliza comandos de Shell seguro que son compatibles con HTTP o SOCKS5 para conectar hosts que están separados por un cortafuegos.	Cómo configurar conexiones de shell seguro predeterminadas a hosts fuera de un firewall [29]

▼ Cómo generar un par de clave pública y clave privada para utilizar con el shell seguro

Los usuarios deben generar un par de clave pública y clave privada cuando su sitio implementa la autenticación basada en host o la autenticación de clave pública de usuario. Para obtener opciones adicionales, consulte la página del comando man `ssh-keygen`(1).

Antes de empezar Pregunte al administrador del sistema si se ha configurado la autenticación basada en host.

1. Inicie el programa de generación de claves.

```
mySystem% ssh-keygen -t rsa
Generating public/private rsa key pair.
…
```

donde `-t` es el tipo de algoritmo, ya sea `rsa`, `dsa` o `rsa1`.

2. Especifique la ruta al archivo que contendrá la clave.

De manera predeterminada, el nombre de archivo `id_rsa`, que representa una clave v2 RSA, aparece entre paréntesis. Puede seleccionar este archivo presionando la tecla Return o proporcione un nombre de archivo alternativo.

```
Enter file in which to save the key (/home/username/.ssh/id_rsa):     <Press Return>
```

El nombre de archivo de la clave pública se crea automáticamente adjuntando la cadena `.pub` al nombre del archivo de clave privada.

3. Escriba una frase de contraseña para usar la clave.

Esta frase de contraseña se utiliza para cifrar la clave privada. Se *desaconseja* el uso de una entrada nula. Tenga en cuenta que la frase de contraseña no se muestra cuando la escribe.

```
Enter passphrase (empty for no passphrase):      <Type passphrase>
```

4. **Vuelva a escribir la frase de contraseña para confirmarla.**

```
Enter same passphrase again:      <Type passphrase>
Your identification has been saved in /home/username/.ssh/id_rsa.
Your public key has been saved in /home/username/.ssh/id_rsa.pub.
The key fingerprint is:
0e:fb:3d:57:71:73:bf:58:b8:eb:f3:a3:aa:df:e0:d1 username@my

System
```

5. **Compruebe que la ruta al archivo de claves sea correcta.**

```
% ls ~/.ssh
id_rsa
id_rsa.pub
```

En este punto, ha creado un par de clave pública y clave privada.

6. **Inicie sesión en el host remoto mediante la opción adecuada según su método de autenticación de la red.**

- **Si el administrador ha configurado la autenticación basada en host, es posible que necesite copiar la clave pública del host local en el host remoto.**
  Ahora puede iniciar sesión en el host remoto. Para obtener detalles, consulte Cómo iniciar sesión en un host remoto con shell seguro [22].

  a. **Escriba el siguiente comando en una línea sin barra diagonal inversa.**

```
% cat /etc/ssh/ssh_host_dsa_key.pub | ssh RemoteHost \
'cat >> ~./ssh/known_hosts && echo "Host key copied"'
```

  b. **Cuando se le pida, proporcione la contraseña de inicio de sesión.**

```
Enter password:      <Type password>
Host key copied
%
```

- **Si su sitio utiliza la autenticación de usuario con claves públicas, rellene el archivo `authorized_keys` en el host remoto.**

  a. **Copie la clave pública en el host remoto.**
  Escriba el siguiente comando en una línea sin barra diagonal inversa.

```
mySystem% cat $HOME/.ssh/id_rsa.pub | ssh myRemoteHost \
'cat >> .ssh/authorized_keys && echo "Key copied"'
```

  Cuando el archivo se copia, se muestra el mensaje “Key copied” (clave copiada).

b. Cuando se le pida, proporcione la contraseña de inicio de sesión.

```
Enter password:       Type login password
Key copied
mySystem%
```

7. (Opcional) Evite futuras solicitudes de frases de contraseña.

Consulte Cómo reducir indicadores de contraseñas en el shell seguro [24]. Para obtener más información, consulte las páginas del comando man `ssh-agent`(1) y `ssh-add`(1).

▼ Cómo cambiar la frase de contraseña de una clave privada de shell seguro

El siguiente comando cambia el mecanismo de autenticación para la clave privada, la frase de contraseña y no la clave privada real. Para obtener más información, consulte la página del comando man `ssh-keygen`(1).

- **Cambie la frase de contraseña.**

  Escriba el comando `ssh-keygen` con la opción `-p` y responda a las solicitudes.

```
mySystem% ssh-keygen -p
Enter file which contains the private key
      (/home/username/.ssh/id_rsa):     <Press Return>
Enter passphrase
      (empty for no passphrase):      <Type passphrase>
Enter same passphrase again:        <Type passphrase>
```

  Donde `-p` solicita cambiar la frase de contraseña de un archivo de clave privada.

▼ Cómo iniciar sesión en un host remoto con shell seguro

1. Inicie una sesión de Shell seguro.

Escriba el comando `ssh` y especifique el nombre del host remoto y de inicio de sesión.

```
mySystem% ssh myRemoteHost -l username
```

2. Si se le solicita, verifique la autenticidad de la clave del host remoto.

Puede aparecer una solicitud que cuestione la autenticidad del host remoto:

```
The authenticity of host 'myRemoteHost' can't be established.
RSA key fingerprint in md5 is: 04:9f:bd:fc:3d:3e:d2:e7:49:fd:6e:18:4f:9c:26
Are you sure you want to continue connecting(yes/no)?
```

Esta solicitud es normal para conexiones iniciales a hosts remotos.

- **Si no puede confirmar la autenticidad del host remoto, escriba `no` y póngase en contacto con el administrador del sistema.**

  ```
  Are you sure you want to continue connecting(yes/no)? no
  ```

  El administrador es responsable de actualizar el archivo `/etc/ssh/ssh_known_hosts` global. Un archivo `ssh_known_hosts` actualizado impide que esta solicitud aparezca.

- **Si confirma la autenticidad del host remoto, responda la solicitud y continúe con el siguiente paso.**

  ```
  Are you sure you want to continue connecting(yes/no)? yes
  ```

3. **Autentíquese en Shell seguro.**

   a. **Cuando se le solicite, escriba la frase de contraseña.**

   ```
   Enter passphrase for key '/home/username/.ssh/id_rsa':      <Type passphrase>
   ```

   b. **Cuando se le solicite, escriba la contraseña de su cuenta.**

   ```
   username@myRemoteHost's password:      <Type password>
   Last login: Wed Sep  7 09:07:49 2011 from myLocalHost
   Oracle Corporation      SunOS 5.11       September 2011
   myRemoteHost%
   ```

4. **Realice transacciones en el host remoto.**

   Los comandos que envía están cifrados. Ninguna respuesta que recibe está cifrada.

5. **Cierre la conexión de Shell seguro.**

   Cuando haya terminado, escriba salir (**`exit`**) o utilice el método habitual para salir de su shell.

   ```
   myRemoteHost% exit
   myRemoteHost% logout
   Connection to myRemoteHost closed
   mySystem%
   ```

ejemplo 1-2 Visualización de interfaz gráfica de usuario remoto en shell seguro

En este ejemplo, `jdoe` es el usuario inicial en ambos sistemas y se le asigna el perfil de derechos de instalación de software. `jdoe` desea utilizar la interfaz gráfica de usuario de Package

Manager en el sistema remoto. El valor predeterminado de la palabra clave `X11Forwarding` sigue siendo `yes`, y el paquete `xauth` se instala en el sistema remoto.

```
% ssh -l jdoe -X myRemoteHost
jdoe@myRemoteHost's password: password
Last login: Wed Sep  7 09:07:49 2011 from myLocalHost
Oracle Corporation      SunOS 5.11       September 2011
myRemoteHost% packagemanager &
```

Cómo reducir indicadores de contraseñas en el shell seguro

Si no desea escribir la frase de contraseña ni la contraseña para utilizar Shell seguro, puede utilizar el daemon del agente. Si tiene cuentas diferentes en hosts diferentes, agregue las claves que necesita para la sesión.

Puede iniciar el daemon del agente manualmente cuando sea necesario, como se describe en el siguiente procedimiento.

1. **Inicie el daemon del agente.**

   ```
   mySystem% eval `ssh-agent`
   Agent pid 9892
   ```

2. **Verifique que el daemon del agente se haya iniciado.**

   ```
   mySystem% pgrep ssh-agent
   9892
   ```

3. **Agregue la clave privada al daemon del agente.**

   ```
   mySystem% ssh-add
   Enter passphrase for /home/username/.ssh/id_rsa:      <Type passphrase>
   Identity added: /home/username/.ssh/id_rsa(/home/username/.ssh/id_rsa)
   mySystem%
   ```

4. **Inicie una sesión de Shell seguro.**

   ```
   mySystem% ssh myRemoteHost -l username
   ```

   No se le solicita una frase de contraseña.

ejemplo 1-3 Uso de opciones de `ssh-add`

En este ejemplo, `jdoe` agrega dos claves al daemon del agente. La opción `-l` se utiliza para enumerar todas las claves que se almacenan en el daemon. Al final de la sesión, la opción `-D` se usa para eliminar todas las claves del daemon del agente.

```
myLocalHost% ssh-agent
mySystem% ssh-add
Enter passphrase for /home/jdoe/.ssh/id_rsa:      <Type passphrase>
Identity added: /home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa)
mySystem% ssh-add /home/jdoe/.ssh/id_dsa
Enter passphrase for /home/jdoe/.ssh/id_dsa:      <Type passphrase>
Identity added:
/home/jdoe/.ssh/id_dsa(/home/jdoe/.ssh/id_dsa)

mySystem% ssh-add -l
md5 1024 0e:fb:3d:53:71:77:bf:57:b8:eb:f7:a7:aa:df:e0:d1
/home/jdoe/.ssh/id_rsa(RSA)
md5 1024 c1:d3:21:5e:40:60:c5:73:d8:87:09:3a:fa:5f:32:53
/home/jdoe/.ssh/id_dsa(DSA)
```

User conducts Oracle Solaris Secure Shell transactions

```
myLocalHost% ssh-add -D
Identity removed:
/home/jdoe/.ssh/id_rsa(/home/jdoe/.ssh/id_rsa.pub)
/home/jdoe/.ssh/id_dsa(DSA)
```

▼ Cómo administrar ZFS con shell seguro de forma remota

De manera predeterminada, el rol `root` no puede iniciar sesión de forma remota con Shell seguro. Históricamente, la raíz ha utilizado Shell seguro para las tareas importantes, como el envío de datos agrupados de ZFS para su almacenamiento en un sistema remoto. En este procedimiento, el rol `root` crea un usuario que puede actuar como administrador remoto de ZFS.

Antes de empezar Debe asumir el rol `root`. Para obtener más información, consulte “Uso de sus derechos administrativos asignados” de “Protección de los usuarios y los procesos en Oracle Solaris 11.2 ”.

1. **Cree el usuario en ambos sistemas.**

   Por ejemplo, cree el usuario `zfsroot` y proporcione una contraseña.

```
source # useradd -c "Remote ZFS Administrator" -u 1201 -d /home/zfsroot zfsroot
source # passwd zfsroot
Enter password:
Retype password:
#

dest # useradd -c "Remote ZFS Administrator" -u 1201 -d /home/zfsroot zfsroot
dest # passwd zfsroot
...
```

El usuario `zfsroot` debe estar definido de idéntica manera en ambos sistemas.

2. **Cree el par de claves del usuario para la autenticación de Shell seguro.**

   El par de claves se crea en el sistema de origen. A continuación, la clave pública se copia al usuario `zfsroot` en el sistema de destino.

   a. **Genere el par de claves y colóquelo en el archivo `id_migrate`.**

   ```
   # ssh-keygen -t rsa -P "" -f ~/id_migrate
   Generating public/private rsa key pair.
   Your identification has been saved in /root/id_migrate.
   Your public key has been saved in /root/id_migrate.pub.
   The key fingerprint is:
   3c:7f:40:ef:ec:63:95:b9:23:a2:72:d5:ea:d1:61:f0 root@source
   ```

   b. **Envíe la parte pública del par de claves al sistema de destino.**

   ```
   # scp ~/id_migrate.pub zfsroot@dest:
   The authenticity of host 'dest (10.134.76.126)' can't be established.
   RSA key fingerprint is 44:37:ab:4e:b7:2f:2f:b8:5f:98:9d:e9:ed:6d:46:80.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added 'dest,10.134.76.126' (RSA) to the list of known hosts.
   Password:
   id_migrate.pub 100% |*****************************| 399 00:00
   ```

3. **En ambos sistemas, asigne el perfil de derechos de administración de archivos ZFS a `zfsroot`.**

   ```
   source # usermod -P +'ZFS File System Management' -S files zfsroot
   dest # usermod -P +'ZFS File System Management' -S files zfsroot
   ```

4. **Verifique que el sistema de destino tiene asignado el perfil de derechos.**

   ```
   dest # profiles zfsroot
   zfsroot:
   ZFS File System Management
   Basic Solaris User
   All
   ```

5. **En el sistema de destino, mueva la parte pública del par de claves al directorio privado `/home/zfsroot/.ssh`.**

   ```
   root@dest # su - zfsroot
   Oracle Corporation      SunOS 5.11      11.1    May 2012
   zfsroot@dest $ mkdir -m 700 .ssh
   zfsroot@dest $ cat id_migrate.pub >> .ssh/authorized_keys
   ```

6. **Verifique que la configuración funciona.**

   ```
   root@source# ssh -l zfsroot -i ~/id_migrate dest \
   ```

```
pfexec /usr/sbin/zfs snapshot zones@test
root@source# ssh -l zfsroot -i ~/id_migrate dest \
pfexec /usr/sbin/zfs destroy zones@test
```

7. **(Opcional) Verifique que puede crear una instantánea y replicar los datos.**

```
root@source# zfs snapshot -r rpool/zones@migrate-all
root@source#  zfs send -rc rpool/zones@migrate-all | \
ssh -l zfsroot -i ~/id_migrate dest pfexec /usr/sbin/zfs recv -F zones
```

8. **(Opcional) Elimine la capacidad de utilizar la cuenta `zfsroot` para la administración de ZFS.**

```
root@dest# usermod -P -'ZFS File System Management' zfsroot
root@dest#  su - zfsroot
zfsroot@dest#  cp .ssh/authorized_keys .ssh/authorized_keys.bak
zfsroot@dest#  grep -v root@source .ssh/authorized_keys.bak> .ssh/authorized_keys
```

▼ Cómo utilizar el reenvío del puerto en el shell seguro

Puede especificar que un puerto local se reenvíe a un host remoto. Efectivamente, un socket se asigna para escuchar el puerto en el lado local. La conexión desde este puerto se realiza mediante un canal seguro al host remoto. Por ejemplo, puede especificar el puerto 143 para obtener correo electrónico remotamente con IMAP4. De forma similar, un puerto se puede especificar en el lado remoto.

Antes de empezar Para utilizar el reenvío del puerto, el administrador debe tener activado el reenvío del puerto en el servidor remoto de Shell seguro. Para obtener detalles, consulte Cómo configurar el reenvío del puerto en el shell seguro [16].

- **Defina el reenvío del puerto seguro desde un puerto remoto hasta un puerto local o desde un puerto local hasta un puerto remoto.**

  - **Para establecer que un puerto local reciba una comunicación segura de un puerto remoto, especifique ambos puertos.**
    Especifique el puerto local que escucha para la comunicación remota. Además, especifique el host remoto y el puerto remoto que reenvían la comunicación.

    ```
    mySystem% ssh -L localPort:remoteHost:remotePort
    ```

  - **Para establecer que un puerto remoto reciba una conexión segura de un puerto local, especifique ambos puertos.**
    Especifique el puerto remoto que escucha para la comunicación remota. Además, especifique el host local y el puerto local que reenvían la comunicación.

```
mySystem% ssh -R remotePort:localhost:localPort
```

ejemplo 1-4 Uso del reenvío del puerto local para recibir correo

El ejemplo siguiente muestra cómo puede utilizar el reenvío del puerto local para recibir correo de manera segura desde un servidor remoto.

```
myLocalHost% ssh -L 9143:myRemoteHost:143 myRemoteHost
```

Este comando reenvía conexiones del puerto `9143` en `myLocalHost` al puerto `143`. El puerto `143` es el puerto del servidor v2 IMAP en `myRemoteHost`. Cuando el usuario inicia una aplicación de correo, el usuario especifica el número de puerto local para el servidor IMAP, como en `localhost:9143`.

ejemplo 1-5 Uso del reenvío del puerto remoto para comunicarse fuera de un cortafuegos

En el siguiente ejemplo, se muestra cómo un usuario en un entorno empresarial puede reenviar conexiones desde un host en una red externa hasta un host dentro de un cortafuegos corporativo.

```
myLocalHost% ssh -R 9022:myLocalHost:22myOutsideHost
```

Este comando reenvía conexiones desde el puerto `9022` en `myOutsideHost` hasta el puerto `22`, el servidor `sshd`, en el host local.

```
myOutsideHost% ssh -p 9022 localhost
myLocalHost%
```

▼ Cómo copiar archivos con shell seguro

El siguiente procedimiento muestra cómo usar el comando `scp` para copiar archivos cifrados entre hosts. Puede copiar archivos cifrados ya sea entre un host local y un host remoto, o entre dos hosts remotos. El comando `scp` solicita la autenticación. Para obtener más información, consulte "Copia remota con el comando scp" de "Gestión de sistemas remotos en Oracle Solaris 11.2 " y la página del comando man `scp`(1).

También puede utilizar el programa de transferencia de archivos segura `sftp`. Para obtener más información, consulte la página del comando man `sftp`(1). Si desea ver un ejemplo, consulte Ejemplo 1-6, "Especificación de un puerto cuando se utiliza el comando `sftp`" y "Inicio de sesión en un sistema remoto para copiar un archivo (sftp)" de "Gestión de sistemas remotos en Oracle Solaris 11.2 ".

Nota - El servicio de auditoría puede auditar transacciones `sftp` a través de la clase de auditoría `ft`. Para `scp`, el servicio de auditoría puede auditar acceso y salida para la sesión `ssh`. Para obtener más información, consulte "Cómo auditar transferencias de archivos FTP y SFTP" de "Gestión de auditoría en Oracle Solaris 11.2 ".

1. **Inicie el programa de copia segura.**

   Especifique el archivo de origen, el nombre de usuario en el destino remoto y el directorio de destino.

   ```
   mySystem% scp myfile.1 username@myRemoteHost:~
   ```

2. **Indique la frase de contraseña cuando se le solicite.**

   ```
   Enter passphrase for key '/home/username/.ssh/id_rsa':      <Type passphrase>
   myfile.1       25% |*******                     |    640 KB  0:20 ETA
   myfile.1
   ```

   Después de escribir la frase de contraseña, se muestra un indicador de progreso, como se puede ver en la segunda línea en la salida. El indicador de progreso muestra:

   - El nombre del archivo
   - El porcentaje del archivo que se ha transferido
   - Una serie de asteriscos que indican el porcentaje del archivo que se ha transferido
   - La cantidad de datos transferidos
   - El tiempo calculado de llegada, o ETA, del archivo completo (es decir, la cantidad restante de tiempo)

ejemplo 1-6 Especificación de un puerto cuando se utiliza el comando `sftp`

En este ejemplo, el usuario desea que el comando `sftp` utilice un puerto concreto. El usuario utiliza la opción `-o` para especificar el puerto.

```
% sftp -o port=2222 guest@RemoteFileServer
```

▼ Cómo configurar conexiones de shell seguro predeterminadas a hosts fuera de un firewall

Puede utilizar Shell seguro para establecer una conexión desde un host dentro de un firewall hasta un host fuera del firewall. Esta tarea se realiza especificando un comando de proxy para `ssh` en un archivo de configuración o como una opción en la línea de comandos. Para la opción de línea de comandos, consulte el Ejemplo 1-7, "Conexión a hosts fuera de un firewall desde la línea de comandos de shell seguro".

Puede personalizar las interacciones de `ssh` mediante su propio archivo de configuración personal, `~/.ssh/config`, o bien puede utilizar los valores del archivo de configuración administrativa, `/etc/ssh/ssh_config`.

Los archivos se pueden personalizar con dos tipos de comandos de proxy. Un comando de proxy es para conexiones HTTP. El otro comando de proxy es para conexiones SOCKS5. Para obtener más información, consulte la página del comando man `ssh_config`(4).

1. **Especifique los comandos de proxy y los hosts en un archivo de configuración.**

   Utilice la sintaxis siguiente para agregar tantas líneas como sea necesario:

   ```
   [Host outside-host]
   ProxyCommand proxy-command [-h proxy-server] \
   [-p proxy-port] outside-host|%h outside-port|%p
   ```

   `Host` *outside-host*

   Limita la especificación del comando de proxy a instancias cuando un nombre de host remoto se especifica en la línea de comandos. Si utiliza un carácter comodín para *outside-host*, aplica la especificación del comando de proxy a un conjunto de hosts.

   proxy-command

   Especifica el comando de proxy.

   El comando puede ser cualquiera de los siguientes:

   - `/usr/lib/ssh/ssh-http-proxy-connect` para conexiones HTTP
   - `/usr/lib/ssh/ssh-socks5-proxy-connect` para conexiones SOCKS5

   `-h` *servidor_proxy* y `-p` *puerto_proxy*

   Estas opciones especifican un servidor proxy y un puerto proxy, respectivamente. Si están presentes, los proxies sustituyen cualquier variable de entorno que especifica servidores proxy y puertos proxy, como `HTTPPROXY`, `HTTPPROXYPORT`, `SOCKS5_PORT`, `SOCKS5_SERVER` y `http_proxy`. La variable `http_proxy` especifica una URL. Si las opciones no se usan, las variables de entorno relevantes se deben definir. Para obtener más información, consulte las páginas del comando man `ssh-socks5-proxy-connect`(1) y `ssh-http-proxy-connect`(1).

   outside-host

   Designa un host específico para conectarse. Utilice el argumento de sustitución `%h` para especificar el host en la línea de comandos.

   outside-port

   Designa un puerto específico para conectarse. Utilice el argumento de sustitución `%p` para especificar el puerto en la línea de comandos. Al especificar `%h` y `%p` sin utilizar la opción

`Host` *outside-host*, el comando de proxy se aplica al argumento de host cada vez que se invoca el comando `ssh`.

2. **Ejecute Shell seguro especificando el host externo.**

   Por ejemplo:

   ```
   mySystem% ssh myOutsideHost
   ```

   Este comando busca una especificación de comando de proxy para `myOutsideHost` en su archivo de configuración personal. Si la especificación no se ha encontrado, el comando busca en el archivo de configuración de todo el sistema, `/etc/ssh/ssh_config`. El comando de proxy se sustituye por el comando `ssh`.

ejemplo 1-7 Conexión a hosts fuera de un firewall desde la línea de comandos de shell seguro

Cómo configurar conexiones de shell seguro predeterminadas a hosts fuera de un firewall [29] explica cómo especificar un comando de proxy en un archivo de configuración. En este ejemplo, un comando de proxy se especifica en la línea de comandos `ssh`.

```
% ssh -o'Proxycommand=/usr/lib/ssh/ssh-http-proxy-connect \
-h myProxyServer -p 8080 myOutsideHost 22' myOutsideHost
```

La opción `-o` para el comando `ssh` proporciona un método de línea de comandos para especificar un comando de proxy. En este ejemplo, el comando realiza lo siguiente:

- Sustituye el comando de proxy HTTP para `ssh`
- Utiliza el puerto `8080` y `myProxyServer` como el servidor proxy
- Se conecta al puerto `22` en `myOutsideHost`

♦ ♦ ♦ CAPÍTULO 2

Referencia de shell seguro

En este capítulo, se describen las opciones de configuración de la función Shell seguro de Oracle Solaris y se tratan los siguientes temas:

Si desea obtener procedimientos para configurar Shell seguro, consulte el Capítulo 1, Uso de shell seguro (tareas).

Sesiones típicas de shell seguro

El daemon de Shell seguro (`sshd`) se inicia, normalmente, durante el inicio cuando los servicios de red se inician. El daemon escucha conexiones de clientes. Una sesión de Shell seguro empieza cuando el usuario ejecuta un comando `ssh`, `scp` o `sftp`. Un daemon `sshd` nuevo se bifurca para cada conexión entrante. Los daemons bifurcados manejan intercambio de claves, cifrado, autenticación, ejecución de comandos e intercambio de datos con el cliente. Estas características de la sesión son determinadas por archivos de configuración del lado del cliente y archivos de configuración del lado del servidor. Los argumentos de la línea de comandos pueden sustituir los valores de los archivos de configuración.

El cliente y el servidor deben autenticarse entre ellos. Tras una autenticación con éxito, el usuario puede ejecutar comandos de manera remota y copiar datos entre hosts.

Características de la sesión de shell seguro

El comportamiento del lado del servidor del daemon `sshd` se controla mediante valores de palabra clave en el archivo `/etc/ssh/sshd_config`. Por ejemplo, el archivo `sshd_config`

controla los tipos de autenticación que se permiten para acceder al servidor. El comportamiento del lado del servidor también se puede controlar mediante las opciones de línea de comandos cuando el daemon `sshd` se inicia.

El comportamiento en el lado del cliente está controlado por palabras clave de Shell seguro en este orden de prioridad:

- Opciones de línea de comandos
- Archivo de configuración del usuario, `~/.ssh/config`
- Archivo de configuración de todo el sistema, `/etc/ssh/ssh_config`

Por ejemplo, un usuario puede sustituir el valor `Ciphers` de la configuración de todo el sistema, que prefiere `aes128-ctr`, especificando `-c aes256-ctr,aes128-ctr,arcfour` en la línea de comandos. Ahora se prefiere el primer cifrado, `aes256-ctr`.

Autenticación e intercambio de claves en shell seguro

El protocolo de Shell seguro admite autenticación de host/usuario de cliente y autenticación de host de servidor. Las claves criptográficas se cambian para la protección de sesiones de Shell seguro. Shell seguro proporciona varios métodos de autenticación e intercambio de claves. Algunos métodos son opcionales. Los mecanismos de autenticación de clientes se muestran en la Tabla 1-1, “Métodos de autenticación para shell seguro”. Los servidores se autentican con claves públicas de host conocidas.

Para la autenticación, Shell seguro admite la autenticación de usuario y la autenticación interactiva genérica, que generalmente involucra contraseñas. Shell seguro también admite la autenticación con claves públicas de usuario y con claves públicas de host de confianza. Las claves pueden ser RSA o DSA. Los intercambios de claves de sesión constan de intercambios de claves efímeras Diffie-Hellman que se firman en el paso de autenticación de servidor. Además, Shell seguro puede usar credenciales GSS para la autenticación.

Adquirir credenciales GSS en shell seguro

A fin de utilizar la GSS-API para la autenticación en Shell seguro, el servidor debe tener credenciales de aceptador GSS-API y el cliente debe tener credenciales de iniciador GSS-API. Se admiten los mecanismos `mech_dh` y `mech_krb5`.

Para `mech_dh`, el servidor tiene credenciales de aceptador GSS-API si `root` ha ejecutado el comando `keylogin`.

Para `mech_krb5`, el servidor tiene credenciales de aceptador GSS-API cuando el principal host que corresponde al servidor tiene una entrada válida en `/etc/krb5/krb5.keytab`.

El cliente tiene credenciales de iniciador para `mech_dh` si se ha realizado una de las siguientes acciones:

- El comando `keylogin` se ha ejecutado.
- El módulo `pam_dhkeys` se utiliza en el archivo `pam.conf`.

El cliente tiene credenciales de iniciador para `mech_krb5` si se ha realizado una de las siguientes acciones:

- El comando `kinit` se ha ejecutado.
- El módulo `pam_krb5` se utiliza en el archivo `pam.conf`.

Para obtener más información acerca del uso de `mech_dh` en la llamada a procedimiento remoto (RPC) segura, consulte Capítulo 10, “Configuración de autenticación de servicios de red” de “Gestión de Kerberos y otros servicios de autenticación en Oracle Solaris 11.2 ”. Para obtener más información acerca del uso de `mech_krb5`, consulte Capítulo 2, “Acerca del servicio Kerberos” de “Gestión de Kerberos y otros servicios de autenticación en Oracle Solaris 11.2 ”. Para obtener más información acerca de los mecanismos, consulte las páginas del comando man `mech`(4) y `mech_spnego`(5).

Ejecución de comandos y reenvío de datos en shell seguro

Una vez completada la autenticación, el usuario puede utilizar Shell seguro, generalmente, mediante la solicitud de un shell o la ejecución de un comando. Mediante las opciones del comando `ssh`, el usuario puede realizar solicitudes. Las solicitudes pueden incluir la asignación de un pseudo-TTY, el reenvío de conexiones X11 o conexiones TCP/IP, o la activación de un programa de autenticación `ssh-agent` por medio de una conexión segura.

Los componentes básicos de una sesión de usuario son los siguientes:

1. El usuario solicita un shell o la ejecución de un comando, que inicia el modo de sesión.

   En este modo, los datos se envían o se reciben por medio del terminal en el lado del cliente. En el lado del servidor, los datos se envían por medio del shell o de un comando.
2. Cuando la transferencia de datos se completa, el programa de usuario finaliza.
3. Todos los reenvíos de X11 y de TCP/IP se detienen, excepto para las conexiones que ya existen. Las conexiones X11 y TCP/IP existentes permanecen abiertas.
4. El servidor envía un mensaje de estado de salida al cliente. Cuando todas las conexiones están cerradas, como los puertos reenviados que habían permanecido abiertos, el cliente cierra la conexión al servidor. A continuación, el cliente se cierra.

Configuración de servidor y cliente en shell seguro

Las características de una sesión de shell seguro son controladas por los archivos de configuración. Los archivos de configuración se pueden sustituir a un cierto grado por opciones en la línea de comandos.

Configuración de cliente en shell seguro

En la mayoría de los casos, las características del lado del cliente de una sesión de Shell seguro son determinadas por el archivo de configuración de todo el sistema, `/etc/ssh/ssh_config`. Los valores del archivo `ssh_config` se pueden sustituir por el archivo de configuración del usuario, `~/.ssh/config`. Además, el usuario puede sustituir ambos archivos de configuración en la línea de comandos.

Los valores en el archivo `/etc/ssh/sshd_config` del servidor determinan qué solicitudes de clientes son permitidas por el servidor. Para obtener una lista de valores de configuración del servidor, consulte “Palabras clave en shell seguro” [36]. Para obtener más información, consulte la página del comando man `sshd_config`(4).

Las palabras clave en el archivo de configuración del cliente se muestran en “Palabras clave en shell seguro” [36]. Si la palabra clave tiene un valor predeterminado, el valor se proporciona. Estas palabras clave se describen detalladamente en las páginas del comando man `ssh`(1), `scp`(1), `sftp`(1) y `ssh_config`(4). Para obtener una lista de palabras clave en orden alfabético y sus valores de sustitución de línea de comando equivalentes, consulte la Tabla 2-5, “Equivalentes de línea de comandos para palabras clave de shell seguro”.

Configuración del servidor en shell seguro

Las características del lado del servidor de una sesión de Shell seguro son determinadas por el archivo `/etc/ssh/sshd_config`. Las palabras clave en el archivo de configuración del servidor se muestran en “Palabras clave en shell seguro” [36]. Si la palabra clave tiene un valor predeterminado, el valor se proporciona. Para obtener una descripción completa de las palabras clave, consulte la página del comando man `sshd_config`(4).

Palabras clave en shell seguro

En las tablas siguientes, se enumeran las palabras clave y sus valores predeterminados (si hay). Las palabras clave están en orden alfabético. Las palabras clave que se aplican al cliente están

en el archivo `ssh_config`. Las palabras clave que se aplican al servidor están en el archivo `sshd_config`. Algunas palabras clave se establecen en ambos archivos. Las palabras clave para un servidor de Shell seguro que ejecuta el protocolo v1 están marcadas.

TABLA 2-1 Palabras clave en archivos de configuración de shell seguro

Palabra clave	Valor predeterminado	Ubicación
`AllowGroups`		Servidor
`AllowTcpForwarding`	`yes`	Servidor
`AllowUsers`		Servidor
`AuthorizedKeysFile`	`~/.ssh/authorized_keys`	Servidor
`Banner`	`/etc/issue`	Servidor
`Batchmode`	`no`	Cliente
`BindAddress`		Cliente
`CheckHostIP`	`yes`	Cliente
`ChrootDirectory`	`no`	Servidor
`Cipher`	algoritmo de cifrado `blowfish`, `3des`	Cliente
`Ciphers`	`aes128-ctr, aes128-cbc, 3des-cbc, blowfish-cbc, arcfour`	Ambos
`ClearAllForwardings`	`no`	Cliente
`ClientAliveCountMax`	`3`	Servidor
`ClientAliveInterval`	`0`	Servidor
`Compression`	`no`	Ambos
`CompressionLevel`		Cliente
`ConnectionAttempts`	`1`	Cliente
`ConnectTimeout`	Tiempo de espera de TCP de sistema	Cliente
`DenyGroups`		Servidor
`DenyUsers`		Servidor
`DisableBanner`	`no`	Cliente
`DynamicForward`		Cliente
`EscapeChar`	`~`	Cliente
`FallBackToRsh`	`no`	Cliente
`ForwardAgent`	`no`	Cliente
`ForwardX11`	`no`	Cliente
`ForwardX11Trusted`	`yes`	Cliente
`GatewayPorts`	`no`	Ambos
`GlobalKnownHostsFile`	`/etc/ssh/ssh_known_hosts`	Cliente
`GSSAPIAuthentication`	`yes`	Ambos
`GSSAPIDelegateCredentials`	`no`	Cliente
`GSSAPIKeyExchange`	`yes`	Ambos

Palabra clave	Valor predeterminado	Ubicación
GSSAPIStoreDelegateCredentials	yes	Servidor
HashKnownHosts	no	Cliente
Host	* Para obtener más información, consulte “Parámetros específicos del host en shell seguro” [39].	Cliente
HostbasedAuthentication	no	Ambos
HostbasedUsesNameFromPacketOnly	no	Servidor
HostKey (v1)	/etc/ssh/ssh_host_key	Servidor
HostKey (v2)	/etc/ssh/host_rsa_key, /etc/ssh/host_dsa_key	Servidor
HostKeyAlgorithms	ssh-rsa, ssh-dss	Cliente
HostKeyAlias		Cliente
HostName		Cliente
IdentityFile	~/.ssh/id_dsa, ~/.ssh/id_rsa	Cliente
IgnoreIfUnknown		Cliente
IgnoreRhosts	yes	Servidor
IgnoreUserKnownHosts	yes	Servidor
KbdInteractiveAuthentication	yes	Ambos
KeepAlive	yes	Ambos
KeyRegenerationInterval	3600 (segundos)	Servidor
ListenAddress		Servidor
LocalForward		Cliente
LoginGraceTime	120 (segundos)	Servidor
LogLevel	info	Ambos
LookupClientHostnames	yes	Servidor
MACs	Algoritmos hmac-sha1-*, hmac-md5-* y hmac-sha2-*.	Ambos
Match		Servidor
MaxStartups	10:30:60	Servidor
NoHostAuthenticationForLocalHost	no	Cliente
NumberOfPasswordPrompts	3	Cliente
PAMServiceName		Servidor
PAMServicePrefix		Servidor
PasswordAuthentication	yes	Ambos
PermitEmptyPasswords	no	Servidor
PermitRootLogin	no	Servidor
PermitUserEnvironment	no	Servidor
PidFile	/system/volatile/sshd.pid	Servidor

Palabra clave	Valor predeterminado	Ubicación
Port	22	Ambos
PreferredAuthentications	hostbased,publickey,keyboard-interactive,passwor	Cliente
PreUserauthHook		Servidor
PrintLastLog	yes	Servidor
PrintMotd	no	Servidor
Protocol	2,1	Ambos
ProxyCommand		Cliente
PubkeyAuthentication	yes	Ambos
RekeyLimit	1 G a 4 G	Cliente
RemoteForward		Cliente
RhostsAuthentication	no	Servidor, v1
RhostsRSAAuthentication	no	Servidor, v1
RSAAuthentication	no	Servidor, v1
ServerAliveCountMax	3	Cliente
ServerAliveInterval	0	Cliente
ServerKeyBits	512 a 768	Servidor, v1
StrictHostKeyChecking	ask	Cliente
StrictModes	yes	Servidor
Subsystem	sftp /usr/lib/ssh/sftp-server	Servidor
SyslogFacility	auth	Servidor
UseFIPS140	no	Ambos
UseOpenSSLEngine	yes	Ambos
UsePrivilegedPort	no	Ambos
User		Cliente
UserKnownHostsFile	~/.ssh/known_hosts	Cliente
UseRsh	no	Cliente
VerifyReverseMapping	no	Servidor
X11DisplayOffset	10	Servidor
X11Forwarding	yes	Servidor
X11UseLocalHost	yes	Servidor
XAuthLocation	/usr/bin/xauth	Ambos

Parámetros específicos del host en shell seguro

A veces, tener diferentes características de shell seguro para diferentes hosts locales es útil. El administrador puede definir conjuntos separados de parámetros en el archivo `/etc/ssh/`

`ssh_config` que se aplicará según la expresión regular o de host mediante la agrupación de entradas en el archivo por la palabra clave `Host`. Si la palabra clave `Host` no se utiliza, las entradas en el archivo de configuración del cliente se aplican a cualquier host local en el que un usuario está trabajando.

Variables de entorno de inicio de sesión y shell seguro

Cuando las siguientes palabras clave de shell seguro no están establecidas en el archivo `sshd_config` obtienen el valor de entradas equivalentes en el archivo `/etc/default/login`.

Entrada en `/etc/default/login`	**Palabra clave y valor en `sshd_config`**
`CONSOLE=*`	`PermitRootLogin=without-password`
`#CONSOLE=*`	`PermitRootLogin=yes`
`PASSREQ=YES`	`PermitEmptyPasswords=no`
`PASSREQ=NO`	`PermitEmptyPasswords=yes`
`#PASSREQ`	`PermitEmptyPasswords=no`
`TIMEOUT=`*seconds*	`LoginGraceTime=`*seconds*
`#TIMEOUT`	`LoginGraceTime=120`
`RETRIES` y `SYSLOG_FAILED_LOGINS`	Sólo se aplican a métodos de autenticación de `password` y `keyboard-interactive`

Cuando las siguientes variables están establecidas por las secuencias de comandos de inicialización del shell de inicio de sesión del usuario, el daemon `sshd` utiliza dichos valores. Cuando las variables no están establecidas, el daemon utiliza el valor predeterminado.

`TIMEZONE` Controla la configuración de la variable de entorno `TZ`. Cuando no está establecida, el daemon `sshd` utiliza el valor de `TZ` cuando se inició el daemon.

`ALTSHELL` Controla la configuración de la variable de entorno `SHELL`. El valor predeterminado es `ALTSHELL=YES`, donde el daemon `sshd` utiliza el valor del shell del usuario. Cuando el valor predeterminado es `ALTSHELL=NO`, el valor `SHELL` no está establecido.

`PATH` Controla la configuración de la variable de entorno `PATH`. Cuando el valor no está establecido, la ruta predeterminada es `/usr/bin`.

SUPATH	Controla la configuración de la variable de entorno PATH para root. Cuando el valor no está establecido, la ruta predeterminada es /usr/sbin:/usr/bin.

Para obtener más información, consulte las páginas del comando man login(1) y sshd(1M).

Mantenimiento de hosts conocidos en shell seguro

Cada host que necesita comunicarse de manera segura con otro host debe tener la clave pública del servidor almacenada en el archivo /etc/ssh/ssh_known_hosts del host local. Aunque una secuencia de comandos podría utilizarse para actualizar los archivos /etc/ssh/ssh_known_hosts, esta práctica es fuertemente desalentada, porque una secuencia de comandos abre una importante vulnerabilidad de seguridad.

El archivo /etc/ssh/ssh_known_hosts sólo debería ser distribuido por un mecanismo seguro, de la siguiente manera:

- Por medio de una conexión segura, como shell seguro, IPsec o ftp Kerberizado de una máquina conocida y de confianza.
- En el tiempo de instalación del sistema

Para evitar la posibilidad de que un intruso obtenga acceso insertando claves públicas falsas en un archivo known_hosts, debe utilizar un origen conocido y de confianza del archivo ssh_known_hosts. El archivo ssh_known_hosts se puede distribuir durante la instalación. Más tarde, las secuencias de comandos que utiliza el comando scp se pueden utilizar para copiar la última versión.

Archivos de shell seguro

En la siguiente tabla, se muestran los principales archivos de shell seguro y los permisos de archivo sugeridos.

TABLA 2-2 Archivos de shell seguro

Nombre del archivo	Descripción	Permisos sugeridos y propietario
~/.rhosts	Contiene los pares de host y nombre de usuario que especifican los hosts en los que el usuario puede iniciar sesión sin una contraseña. Este archivo también es utilizado por los daemons rlogind y rshd.	-rw-r--r-- *username*
~/.shosts	Contiene los pares de host y nombre de usuario que especifican los hosts en los que el usuario puede iniciar sesión sin una contraseña. Este archivo no es utilizado por	-rw-r--r-- *username*

Nombre del archivo	Descripción	Permisos sugeridos y propietario
	otras utilidades. Para obtener más información, consulte la página del comando man sshd(1M) en la sección FILES.	
~/.ssh/authorized_keys	Contiene las claves públicas del usuario que tiene permitido iniciar sesión en la cuenta de usuario.	-rw-r--r-- *username*
~/.ssh/config	Configura los valores del usuario que sustituyen los valores del sistema.	-rw-r--r-- *username*
~/.ssh/environment	Contiene asignaciones iniciales en el momento del inicio de sesión. De manera predeterminada, este archivo no se lee. La palabra clave PermitUserEnvironment en el archivo sshd_config se debe establecer en yes para que este archivo se lea.	-rw-r--r-- *username*
/etc/hosts.equiv	Contiene los hosts que se utilizan en la autenticación .rhosts. Este archivo también es utilizado por los daemons rlogind y rshd.	-rw-r--r-- root
~/.ssh/known_hosts	Contiene las claves públicas de host de todos los hosts con los que el cliente puede comunicarse de forma segura. El archivo se mantiene automáticamente. Cada vez que el usuario se conecta con un host desconocido, la clave del host remoto se agrega al archivo.	-rw-r--r-- *username*
/etc/default/login	Proporciona valores predeterminados para el daemon sshd cuando los parámetros sshd_config correspondientes no están establecidos.	-r--r--r-- root
/etc/nologin	Si el archivo existe, el daemon sshd permite que sólo root inicie sesión. El contenido de este archivo se muestra a los usuarios que intentan iniciar sesión.	-rw-r--r-- root
~/.ssh/rc	Contiene las rutinas de inicialización que se ejecutan antes de que el shell del usuario se inicie. Para ver un ejemplo de rutina de inicialización, consulte la página del comando man sshd(1M).	-rw-r--r-- *username*
/etc/ssh/shosts.equiv	Contiene los hosts que se utilizan en la autenticación basada en host. Este archivo no es utilizado por otras utilidades.	-rw-r--r-- root
/etc/ssh/ssh_config	Configura los valores del sistema en el sistema cliente.	-rw-r--r-- root
/etc/ssh/ssh_host_dsa_key o /etc/ssh/ssh_host_rsa_key	Contiene la clave privada de host.	-rw------- root
/etc/ssh_host_key.pub o /etc/ssh/ssh_host_dsa_key.pub o /etc/ssh/ssh_host_rsa_key.pub	Contiene la clave pública de host, por ejemplo, /etc/ssh/ssh_host_rsa_key.pub. Se utiliza para copiar la clave del host en el archivo known_hosts local.	-rw-r--r-- root
/etc/ssh/ssh_known_hosts	Contiene las claves públicas de host de todos los hosts con los que el cliente puede comunicarse de forma segura. El archivo es rellenado por el administrador.	-rw-r--r-- root
/etc/ssh/sshd_config	Contiene datos de configuración para sshd, el daemon de shell seguro.	-rw-r--r-- root

Nombre del archivo	Descripción	Permisos sugeridos y propietario
`/system/volatile/sshd.pid`	Contiene el ID de proceso del daemon de shell seguro, `sshd`. Si hay varios daemons en ejecución, el archivo contiene el último daemon que se ha iniciado.	`-rw-r--r-- root`
`/etc/ssh/sshrc`	Contiene rutinas de inicialización específicas de host que son especificadas por un administrador.	`-rw-r--r-- root`

Nota - El archivo `sshd_config` se pueden sustituir por un archivo de un paquete personalizado para el sitio. Para obtener más información, consulte la definición de atributo de archivo `overlay` en la página del comando man `pkg`(5).

En la siguiente tabla, se muestran los archivos de shell seguro que se pueden sustituir por palabras clave u opciones de comandos.

TABLA 2-3 Sustituciones para la ubicación de los archivos de shell seguro

Nombre del archivo	Valor de sustitución de palabra clave	Valor de sustitución de línea de comandos
`/etc/ssh/ssh_config`		`ssh -F` *config-file* `scp -F` *config-file*
`~/.ssh/config`		`ssh -F` *config-file*
`/etc/ssh/host_rsa_key` `/etc/ssh/host_dsa_key`	`HostKey`	
`~/.ssh/identity` `~/.ssh/id_dsa`, `~/.ssh/id_rsa`	`IdentityFile`	`ssh -i` *ID-file* `scp -i` *ID-file*
`~/.ssh/authorized_keys`	`AuthorizedKeysFile`	
`/etc/ssh/ssh_known_hosts`	`GlobalKnownHostsFile`	
`~/.ssh/known_hosts`	`UserKnownHostsFile` `IgnoreUserKnownHosts`	

Comandos de shell seguro

En la siguiente tabla, se resumen los principales comandos de shell seguro.

TABLA 2-4 Comandos de shell seguro

Página del comando man	Descripción
ssh(1)	Inicia sesión de un usuario en un equipo remoto y ejecuta de manera segura comandos en un equipo remoto. El comando `ssh` permite comunicaciones cifradas seguras entre dos hosts que no son de confianza por medio de una red no segura. Las conexiones X11 y los puertos TCP/IP arbitrarios también se pueden reenviar por medio del canal seguro.
sshd(1M)	El daemon de shell seguro. El daemon escucha conexiones de clientes y permite comunicaciones cifradas seguras entre dos hosts que no son de confianza por medio de una red no segura.
ssh-add(1)	Agrega identidades RSA o DSA al agente de autenticación, `ssh-agent`. Las identidades también se denominan *claves*.
ssh-agent(1)	Contiene claves privadas que se utilizan para la autenticación de clave pública. El programa `ssh-agent` se inicia al principio de una sesión X o de una sesión de inicio de sesión. Todas las demás ventanas y otros programas se inician como clientes del programa `ssh-agent`. Mediante el uso de variables de entorno, el agente se puede localizar y utilizar para la autenticación cuando los usuarios utilizan el comando `ssh` para iniciar sesión en otros sistemas.
ssh-keygen(1)	Genera y gestiona claves de autenticación para shell seguro.
ssh-keyscan(1)	Recopila las claves públicas de un número de hosts de shell seguro. Ayuda en la generación y la verificación de archivos `ssh_known_hosts`.
ssh-keysign(1M)	Es utilizado por el comando `ssh` para acceder a las claves de host en el host local. Genera la firma digital que se requiere durante la autenticación basada en host con shell seguro v2. El comando es invocado por el comando `ssh`, no por el usuario.
scp(1)	Copia de manera segura archivos entre hosts en una red por medio de un transporte `ssh` cifrado. A diferencia del comando `rcp`, el comando `scp` solicita contraseñas o frases de contraseña si la información de contraseña es necesaria para la autenticación.
sftp(1)	Es un programa de transferencia de archivos interactivo similar al comando `ftp`. A diferencia del comando `ftp`, el comando `sftp` realiza todas las operaciones por medio de un transporte `ssh` cifrado. El comando se conecta, inicia sesión en el nombre de host especificado y, a continuación, introduce el modo de comando interactivo.

En la siguiente tabla, se muestran las opciones de comandos que sustituyen palabras clave de shell seguro. Las palabras clave se especifican en los archivos `ssh_config` y `sshd_config`.

TABLA 2-5 Equivalentes de línea de comandos para palabras clave de shell seguro

Palabra clave	Valor de sustitución de línea de comandos `ssh`	Valor de sustitución de línea de comandos `scp`
`BatchMode`		`scp -B`
`BindAddress`	`ssh -b` *bind-addr*	`scp -a` *bind-addr*
`Cipher`	`ssh -c` *cipher*	`scp -c` *cipher*
`Ciphers`	`ssh -c` *cipher-spec*	`scp -c` *cipher-spec*
`Compression`	`ssh -C`	`scp -C`
`DynamicForward`	`ssh -D` *puerto SOCKS4*	
`EscapeChar`	`ssh -e` *escape-char*	

Palabra clave	Valor de sustitución de línea de comandos `ssh`	Valor de sustitución de línea de comandos `scp`
`ForwardAgent`	`ssh -A` para activar `ssh -a` para desactivar	
`ForwardX11`	`ssh -X` para activar `ssh -x` para desactivar	
`GatewayPorts`	`ssh -g`	
`IPv4`	`ssh -4`	`scp -4`
`IPv6`	`ssh -6`	`scp -6`
`LocalForward`	`ssh -L` *puerto_local:host_remoto: puerto_remoto*	
`MACS`	`ssh -m` *MAC-spec*	
`Port`	`ssh -p` *port*	`scp -P` *port*
`Protocol`	`ssh -2` sólo para v2	
`RemoteForward`	`ssh -R` *puerto_remoto:host_local: puerto_local*	

Índice

B

C

P

R

S

T

U

V