Rasgos Esenciales de SSH

Introducción

SSH es un protocolo seguro utilizado como el medio principal para conectarse a servidores Linux de forma remota. Proporciona una interfaz basada en texto al generar un shell remoto. Después de conectarse, todos los comandos que escribe en su terminal local se envían al servidor remoto y se ejecutan allí.

En esta guía de estilo de hoja de trucos, cubriremos algunas formas comunes de conectarse con SSH para lograr sus objetivos. Esto puede usarse como una referencia rápida cuando necesite saber cómo conectarse o configurar su servidor de diferentes maneras.

La forma más común de conectarse a un servidor Linux remoto es a través de SSH. SSH significa Secure Shell y proporciona una forma segura de ejecutar órdenes, realizar cambios y configurar servicios de forma remota. Cuando se conecta a través de SSH inicia sesión con una cuenta que existe en el servidor remoto.

El cifrado de SSH

La ventaja significativa ofrecida por SSH sobre sus predecesores es el uso del cifrado para asegurar la transferencia segura de información entre el host y el cliente. Host se refiere al servidor remoto al que estás intentando acceder, mientras que el cliente es el equipo que estás utilizando para acceder al host. Hay tres tecnologías de cifrado diferentes utilizadas por SSH:

1. Cifrado simétrico
2. Cifrado asimétrico
3. Hashing

Cifrado Simétrico

El cifrado simétrico es utilizado para cifrar toda la comunicación durante una sesión SSH. Tanto el cliente como el servidor derivan la clave simétrica secreta mediante un algoritmo de intercambio de claves para cifrado simétrico. Gracias a este algoritmo la clave nunca se transmite entre el cliente y el host; en lugar de eso, los dos equipos comparten datos públicos y luego los manipulan para calcular de forma independiente la clave secreta. Incluso si otra máquina captura los datos públicamente compartidos, no será capaz de calcular la clave simétrica secreta por la dureza del problema al que el algoritmo la hace enfrentarse.

Debe tenerse en cuenta, sin embargo, que eltoken secreto es específico para cada sesión SSH, y se genera antes de la autenticación del cliente. Una vez generada la clave, todos los paquetes que se mueven entre las dos máquinas deben ser cifrados por la clave simétrica privada o secreta. Esto incluye la contraseña escrita en la consola por el usuario, por lo que las credenciales siempre están protegidas de los fisgones de paquetes de red.

Existen varios criptosistemas de cifrado simétrico incluyendo, pero no limitado a, AES (Advanced Encryption Standard), CAST128, Blowfish, etc. Antes de establecer una conexión segura, el cliente y un host deciden qué cifrado usar, publicando una lista de cifrados soportados por orden de preferencia. Es utilizado como cifrado bidireccional el realizado por el criptosistema preferido de entre los soportados por los clientes y que está

presente en la lista de criptosistemas del host.

Por ejemplo, si dos máquinas Ubuntu 18.04 LTS se comunican entre sí a través de SSH, utilizarán aes128-ctr como su cifrado predeterminado.

Cifrado Asimétrico

SSH utiliza cifrado asimétrico en varias ocasiones. Durante el proceso inicial de intercambio de claves utilizado para configurar el cifrado simétrico (empleado para cifrar la sesión), se utiliza el cifrado asimétrico. En esta etapa, ambas partes producen pares de claves pública-privada para uso temporal e intercambian la clave pública para producir el secreto compartido que se utilizará para el cifrado simétrico.

El uso más discutido del cifrado asimétrico con SSH proviene de la autenticación basada en claves SSH. Los pares de claves SSH se pueden usar para autenticar un cliente en un servidor. El cliente crea un par de claves y luego carga la clave pública en cualquier servidor remoto al que desee acceder. Esto se coloca en un archivo llamado `certified_keys` dentro del directorio `~/.ssh` en el directorio principal de la cuenta de usuario en el servidor remoto.

Después de establecer el cifrado simétrico para asegurar las comunicaciones entre el servidor y el cliente, el cliente debe autenticarse para poder acceder. El servidor puede usar la clave pública en este archivo para cifrar un mensaje de desafío al cliente. Si el cliente puede probar que fue capaz de descifrar este mensaje, ha demostrado que posee la clave privada asociada. El servidor puede entonces configurar el entorno para el cliente.

Hashing

Otra forma de manipulación de datos que SSH aprovecha es el hash criptográfico. Las funciones hash criptográficas son métodos para crear una "firma" o resumen sucinto de un conjunto de información. Sus principales atributos distintivos son que nunca deben ser revertidos, son virtualmente imposibles de influenciar de manera predecible y son prácticamente únicos.

El uso de la misma función y mensaje de hash debería producir el mismo hash; modificar cualquier parte de los datos debería producir un hash completamente diferente. Un usuario no debería poder producir el mensaje original a partir de un hash determinado, pero debería poder saber si un mensaje dado produjo un hash dado.

Dadas estas propiedades, los hashes se utilizan principalmente para fines de integridad de datos y para verificar la autenticidad de la comunicación. El uso principal en SSH es con HMAC, o códigos de autenticación de mensajes basados en hash (hash-based message authentication codes). Estos se utilizan para garantizar que el texto del mensaje recibido esté intacto y sin modificaciones.

Como parte de la negociación de cifrado simétrico descrita anteriormente, se selecciona un algoritmo de código de autenticación de mensaje (MAC, acrónimo de "message authentication code"). El algoritmo se elige trabajando a través de la lista de opciones de MAC aceptables del cliente. Se utilizará el primero de esta lista que el servidor admite.

Cada mensaje que se envía después de negociar el cifrado debe contener un MAC para que la otra parte pueda verificar la integridad del paquete. El MAC se calcula a partir del secreto simétrico compartido, el número de secuencia del paquete del mensaje y el contenido real del mensaje.

El MAC mismo se envía fuera del área cifrada simétricamente como la parte final del paquete. Los investigadores generalmente recomiendan este método de cifrar los datos primero y luego calcular el MAC.

El Funcionamiento de SSH

Probablemente ya tenga una comprensión básica de cómo funciona SSH. El protocolo SSH emplea un modelo cliente-servidor para autenticar a dos partes y cifrar los datos entre ellas.

El componente del servidor escucha las conexiones en un puerto designado. Es responsable de negociar la conexión segura, autenticar a la parte que se conecta y generar el entorno correcto si se aceptan las credenciales.

El cliente es responsable de comenzar el protocolo de enlaceTCP inicial con el servidor, negociar la conexión segura, verificar que la identidad del servidor coincida con la información registrada anteriormente y proporcionar credenciales para autenticarse.

La sesión SSH se establece en dos etapas separadas. El primero es acordar y establecer cifrado para proteger la comunicación futura. La segunda etapa es autenticar al usuario y descubrir si se debe otorgar acceso al servidor.

Negociación de Cifrado para la Sesión

Cuando un cliente realiza una conexión TCP, el servidor responde con las versiones de protocolo que admite. Si el cliente puede coincidir con una de las versiones de protocolo aceptables, la conexión continúa. El servidor también proporciona su clave de host pública, que el cliente puede usar para verificar si este fue el host previsto.

En este punto, ambas partes negocian una clave de sesión utilizando una versión de algo llamado algoritmo Diffie-Hellman. Este algoritmo (y sus variantes) hacen posible que cada parte combine sus propios datos privados con datos públicos del otro sistema para llegar a una clave de sesión secreta idéntica.

La clave de sesión se utilizará para cifrar toda la sesión. Los pares de claves pública y privada utilizados para esta parte del procedimiento están completamente separados de las claves SSH utilizadas para autenticar un cliente en el servidor.

La base de este procedimiento para el clásico Diffie-Hellman es:

1. Ambas partes acuerdan un número primo grande, que servirá como valor inicial.
2. Ambas partes acuerdan un generador de cifrado (generalmente AES), que se utilizará para manipular los valores de una manera predefinida.
3. Independientemente, cada parte obtiene otro número primo que se mantiene en secreto de la otra parte. Este número se utiliza como clave privada para esta interacción (diferente de la clave SSH privada utilizada para la autenticación).
4. La clave privada generada, el generador de cifrado y el número primo compartido se utilizan para generar una clave pública que se deriva de la clave privada, pero que se puede compartir con la otra parte.

5. Ambos participantes intercambian sus claves públicas generadas.
6. La entidad receptora usa su propia clave privada, la clave pública de la otra parte y el número primo compartido original para calcular una clave secreta compartida. Aunque esto es calculado independientemente por cada parte, usando claves privadas y públicas opuestas, dará como resultado la misma clave secreta compartida.
7. El secreto compartido se usa para cifrar toda la comunicación que sigue.

El cifrado secreto compartido que se utiliza para el resto de la conexión se denomina protocolo de paquete binario. El proceso anterior permite a cada parte participar igualmente en la generación del secreto compartido, lo que no permite que un extremo controle el secreto. También cumple la tarea de generar un secreto compartido idéntico sin tener que enviar esa información a través de canales inseguros.

El secreto generado es una clave simétrica, lo que significa que la misma clave utilizada para cifrar un mensaje se puede utilizar para descifrarlo en el otro lado. El propósito de esto es envolver toda la comunicación adicional en un túnel encriptado que no pueda ser descifrado por personas externas.

Una vez que se establece el cifrado de la sesión, comienza la etapa de autenticación del usuario.

Autenticar el acceso del usuario al servidor

La siguiente etapa consiste en autenticar al usuario y decidir el acceso. Existen algunos métodos diferentes que se pueden usar para la autenticación, en función de lo que acepta el servidor.

La más simple es probablemente la autenticación de contraseña, en la que el servidor simplemente le solicita al cliente la contraseña de la cuenta con la que intenta iniciar sesión. La contraseña se envía a través del cifrado negociado, por lo que es segura frente a terceros.

Aunque la contraseña estará cifrada, este método generalmente no se recomienda debido a las limitaciones en la complejidad de la contraseña. Las secuencias de órdenes automatizadas pueden romper muy fácilmente contraseñas de longitudes normales en comparación con otros métodos de autenticación.

La alternativa más popular y recomendada es el uso de pares de claves SSH. Los pares de claves SSH son claves asimétricas, lo que significa que las dos claves asociadas cumplen funciones diferentes.

La clave pública se usa para cifrar datos que solo se pueden descifrar con la clave privada. La clave pública se puede compartir libremente porque, aunque puede cifrar la clave privada, no existe ningún método para derivar la clave privada de la clave pública.

La autenticación mediante pares de claves SSH comienza después de que se haya establecido el cifrado simétrico como se describe en la última sección. El procedimiento ocurre así:

1. El cliente comienza enviando una ID para el par de claves con el que desea autenticarse al servidor.
2. El servidor verifica el archivo de claves autorizadas de la cuenta en la que el cliente intenta iniciar sesión para obtener la ID de la clave.
3. Si se encuentra una clave pública con ID coincidente en el archivo, el servidor genera un número aleatorio (un string aleatorio de 256 bits) y utiliza la clave pública para cifrar el número.
4. El servidor envía al cliente este mensaje cifrado.
5. Si el cliente realmente tiene la clave privada asociada, podrá descifrar el mensaje utilizando esa clave,

revelando el número original.
6. El cliente combina el número descifrado con la clave de sesión compartida que se utiliza para cifrar la comunicación y calcula el hash (el algoritmo fue MD5 y ahora es SHA2) de este valor.
7. Luego, el cliente envía este hash al servidor como respuesta al mensaje de número cifrado.
8. El servidor usa la misma clave de sesión compartida y el número original que envió al cliente para calcular el valor hash por sí mismo. Compara su propio cálculo con el que el cliente envió de vuelta. Si estos dos valores coinciden, demuestra que el cliente estaba en posesión de la clave privada y que el cliente está autenticado.

Como puede verse, la asimetría de las claves permite al servidor cifrar mensajes al cliente utilizando la clave pública. El cliente puede demostrar que posee la clave privada descifrando el mensaje correctamente. Los dos tipos de cifrado que se utilizan (secreto compartido simétrico y claves público-privadas asimétricas) pueden aprovechar sus fortalezas específicas en este modelo.

Instalación de ssh

Para usar `ssh` en un equipo remoto, al que llamaremos en adelante "host remoto" o "servidor remoto" o simplemente "servidor", ésta utilidad debe estar instalada por su administrador. En un equipo basado en Ubuntu/Debian se haría mediante:

```
sudo apt install openssh-server
```

o forma similar como superusuario (en cuyo caso no escribiremos `sudo`) en equipos bajo Debian. A partir de este momento tendremos unas vastas posibilidades de acceso al servidor y de control sobre él.

Lo Básico de la Conexión con SSH

Como usuario de Linux o Mac usar SSH es verdaderamente fácil; como usuario de Windows, será necesario utilizar un cliente SSH para abrir conexiones SSH y el más popular es PuTTY.

En el supuesto de que el nombre de usuario en la máquina local sea el mismo que el nombre de usuario de la cuenta a la que va a acceder en el servidor remoto, puede conectar mediante la orden:

```
ssh remote_host
```

La orden `ssh` sigue la siguiente sitaxis:

```
ssh username@remote_host
```

y si usa el nombre del servido remoto, entonces sería:

```
ssh username@server_name.domain.ext
```

La orden básica de SSH indica al sistema local que desea abrir una Conexión de Shell Segura y cifrada. `username` representa la cuenta a la que deseas acceder:

- `username` es el nombre de la cuenta a la que se quiere acceder.
- `remote_host` es la dirección IP del equipo, por ejemplo `244.235.23.19`. Puede ser sustituido por `server_name.domain.ext`, que es el nombre del equipo en la que existe creada la cuenta `username`.

Al pulsar intro, será solicitada la contraseña de la cuenta de nombre `username`. Al escribirla, nada aparecerá en la pantalla pero tras hacerlo y pulsar de nuevo enter la contraseña se estará transmitiendo. Si es correcta, verá una ventana de terminal remota y realmente ya estará en la cuenta `username` del servidor remoto y podrá hacer en él todo lo que esta cuenta le permita.

Ejecutar una Orden en un Servidor Remoto

Para ejecutar una sola orden en un servidor remoto sin generar una sesión de shell puede agregar la orden después de la información de conexión, de esta manera se tendría:

```
ssh username@remote_host command_to_run
```

Con ello se conectará al host remoto, se autenticará con sus credenciales y ejecutará la orden especificada. La conexión será cerrará inmediatamente después.

Iniciar Sesión en un Servidor con un Puerto Diferente

De forma predeterminada, el demonio SSH en un servidor se ejecuta en el puerto 22; su cliente SSH asumirá que éste es el caso cuando intente conectarse. Si su servidor SSH está escuchando en un puerto no estándar (esto se demuestra en una sección posterior), deberá especificar el nuevo número de puerto cuando se conecte con su cliente.

Puede hacer esto especificando el número de puerto con la opción `-p`:

```
ssh -p port_num username@remote_host
```

Para evitar tener que hacer esto cada vez que inicie sesión en su servidor remoto, puede crear o editar un archivo de configuración en el directorio `~/.ssh` dentro del directorio de inicio de su computadora local.

La edición o creación el archivo puede ser así:

```
nano ~/.ssh/config
```

Aquí puede establecer opciones de configuración específicas del host. Para especificar su nuevo puerto, use un formato como este:

```
Host remote_alias
    HostName remote_host
    Port port_num
```

Esto le permitirá iniciar sesión sin señalar en la línea de órdenes el número de puerto específico.