SSH: Comunicación Cliente-Servidor

1. Introducción a SSH

SSH (Secure Shell) es un protocolo de red que permite la administración segura de sistemas y la transferencia de datos a través de una red no segura. Proporciona autenticación, confidencialidad e integridad de los datos. SSH se utiliza comúnmente para acceder de forma remota a sistemas Linux y Unix.

2. Componentes de SSH

- **Cliente SSH**: Programa que permite a los usuarios conectarse a un servidor SSH.
- **Servidor SSH**: Programa que escucha las conexiones SSH en un puerto específico (por defecto, el puerto 22).

3. Establecimiento de la Comunicación

3.1. Proceso de Conexión

1. **Inicio de Conexión**: El cliente SSH inicia la conexión al servidor SSH especificando la dirección IP o el nombre del host y el puerto.
2. **Negociación de Protocolo**: El cliente y el servidor intercambian versiones del protocolo SSH.
3. **Intercambio de Claves**: Se establece un canal cifrado utilizando un algoritmo de cifrado simétrico.
4. **Autenticación**: El servidor autentica al cliente usando uno de los métodos de autenticación disponibles.

4. Métodos de Autenticación

4.1. Autenticación por Contraseña

1. **Solicitud de Contraseña**: Si el cliente se conecta al servidor, este le solicita la contraseña del usuario.
2. **Verificación**: El servidor verifica la contraseña. Si es correcta, se establece la sesión.

Ventajas:

- Sencillez en la configuración inicial.
- No requiere configuración adicional.

Desventajas:

- Menos seguro, ya que las contraseñas pueden ser interceptadas.
- La gestión de contraseñas puede ser un riesgo.

4.2. Autenticación por Clave Pública/Privada

1. **Generación de Claves**: El usuario genera un par de claves (privada y pública) utilizando un comando como `ssh-keygen`.
2. **Distribución de Clave Pública**: La clave pública se copia al servidor, generalmente en el archivo `~/.ssh/authorized_keys` del usuario.
3. **Conexión**: Cuando el cliente intenta conectarse, el servidor envía un reto cifrado con la clave pública.
4. **Desencriptación**: El cliente desencripta el reto utilizando su clave privada y lo envía de vuelta al servidor.
5. **Verificación**: El servidor verifica la respuesta. Si es correcta, se establece la sesión.

Ventajas:

- Más seguro, ya que la clave privada nunca se transmite por la red.
- Permite la autenticación sin necesidad de ingresar contraseñas repetidamente.

Desventajas:

- Requiere una configuración inicial más compleja.
- La clave privada debe ser protegida adecuadamente.

5. Configuración Básica de SSH

5.1. Configuración del Servidor SSH

1. **Instalación**:

   ```
   apt-get install openssh-server
   ```

2. **Configuración**: Editar el archivo de configuración `/etc/ssh/sshd_config` para ajustar opciones como:
   - Cambiar el puerto por defecto.
   - Deshabilitar la autenticación por contraseña si se usa clave pública.
   - Permitir o denegar acceso a usuarios específicos.
3. **Reiniciar el Servicio**:

   ```
   systemctl restart ssh
   ```

5.2. Configuración del Cliente SSH

1. **Instalación** (en sistemas Linux suele venir preinstalado):

   ```
   apt-get install openssh-client
   ```

2. **Conexión a un Servidor**:

   ```
   ssh usuario@servidor
   ```

3. **Uso de Claves**: Para usar autenticación con claves, primero genera el par de claves:

```
ssh-keygen -t rsa -b 2048
```

Luego copia la clave pública al servidor:

```
ssh-copy-id usuario@servidor
```

6. Tipos de Cifrado en SSH

SSH utiliza varios algoritmos de cifrado para garantizar la seguridad de la comunicación entre el cliente y el servidor. Estos algoritmos se dividen en varias categorías, cada una con sus propias características y niveles de seguridad.

6.1. Algoritmos de Cifrado de Sesión

Estos algoritmos se utilizan para cifrar los datos transmitidos durante una sesión SSH. Algunos de los más comunes incluyen:

- **AES (Advanced Encryption Standard)**: Es uno de los algoritmos más utilizados. Soporta claves de 128, 192 y 256 bits. Es rápido y seguro, ideal para la mayoría de las aplicaciones.
- **ChaCha20**: Un algoritmo de cifrado moderno que ofrece alta seguridad y rendimiento, especialmente en dispositivos con recursos limitados. Suele ser preferido en entornos móviles.
- **3DES (Triple Data Encryption Standard)**: Aunque alguna vez fue un estándar de la industria, se considera obsoleto y menos seguro en comparación con AES.

6.2. Algoritmos de Hash

Los algoritmos de hash se utilizan para verificar la integridad de los datos. Algunos de los más comunes son:

- **SHA-2 (Secure Hash Algorithm 2)**: Incluye SHA-256 y SHA-512. Proporciona una alta seguridad y es ampliamente utilizado en aplicaciones modernas.
- **SHA-1**: Anteriormente popular, se considera inseguro debido a vulnerabilidades descubiertas en su implementación.

6.3. Algoritmos de Intercambio de Claves

Estos algoritmos se utilizan para intercambiar claves de cifrado de forma segura. Algunos ejemplos son:

- **Diffie-Hellman**: Permite que dos partes establezcan una clave compartida de manera segura sobre un canal no seguro. Hay variantes como DH con grupos de 2048 bits que ofrecen una buena seguridad.
- **ECDH (Elliptic Curve Diffie-Hellman)**: Utiliza matemáticas de curvas elípticas para ofrecer la misma seguridad que Diffie-Hellman pero con claves más pequeñas, lo que mejora el rendimiento.

6.4. Algoritmos de Autenticación

Para la autenticación de los usuarios, SSH utiliza:

- **RSA**: Un algoritmo de clave pública que es ampliamente utilizado, aunque el tamaño mínimo recomendado de la clave es de 2048 bits para garantizar la seguridad.
- **DSA (Digital Signature Algorithm)**: Aunque menos utilizado hoy en día, es una opción de firma digital en el contexto de SSH.
- **ECDSA (Elliptic Curve Digital Signature Algorithm)**: Utiliza curvas elípticas y es más eficiente que RSA en términos de tamaño de clave y rendimiento.

7. Conclusiones

SSH es una herramienta poderosa para la administración remota de sistemas. Comprender los métodos de autenticación, los tipos de cifrado y cómo se establece la comunicación entre el cliente y el servidor es esencial para mantener la seguridad de las conexiones remotas. La autenticación mediante claves públicas es la opción más segura y recomendada para la mayoría de los entornos.

Los tipos de cifrado utilizados en SSH son fundamentales para garantizar la confidencialidad, integridad y autenticación de las comunicaciones. Elegir los algoritmos adecuados y mantener el software actualizado son prácticas esenciales para asegurar la conexión SSH.