Guía de estudio - Gestión de claves SSH, parte I

¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

En el mundo de la administración remota y la seguridad informática, el uso de claves SSH se ha convertido en una herramienta esencial. Las claves SSH, compuestas de un par formado por una clave privada y una clave pública, permiten autenticaciones seguras y sin necesidad de contraseñas tradicionales.

Esta guía se centra en la creación, distribución y gestión de estas claves, proporcionando una base sólida para entender cómo implementar y mantener una infraestructura segura basada en autenticación SSH. Aprenderás a generar tus claves, distribuir la clave pública a servidores remotos y gestionar de forma adecuada el ciclo de vida de tus claves.

¡Vamos con todo!

Tabla de contenidos

¡Comencemos!

Creación de claves SSH

Conceptos básicos

- **Clave privada:** Se mantiene de forma segura en tu dispositivo y es el componente secreto del par de claves.
- **Clave pública:** Se distribuye a los servidores con los que deseas autenticarte. Es segura para ser compartida.

Pasos para la creación

1. **Abrir la terminal:**
   En cualquier sistema basado en Unix/Linux (incluyendo macOS), abre tu terminal preferida.
- **Ejecutar el comando de generación:**
  Se recomienda usar algoritmos modernos, como Ed25519, aunque RSA sigue siendo común. Para generar una clave Ed25519:

```
ssh-keygen -t ed25519 -C "tu_correo@dominio.com"
```

- O, para RSA:

```
ssh-keygen -t rsa -b 4096 -C "tu_correo@dominio.com"
```

2. El parámetro -C añade un comentario que ayuda a identificar la clave.

3. Definir la ubicación y la passphrase:
   - Se te preguntará dónde deseas guardar la clave (por defecto, en ~/.ssh/id_ed25519 o ~/.ssh/id_rsa).
   - Se recomienda establecer una passphrase para agregar una capa adicional de seguridad. Si optas por no usar passphrase, presiona Enter.

4. Verificar la creación:
   Una vez finalizado el proceso, encontrarás dos archivos:
   - La clave privada: ~/.ssh/id_ed25519 (o id_rsa)
   - La clave pública: ~/.ssh/id_ed25519.pub (o id_rsa.pub)

Distribución de la clave pública

Propósito y ventajas

Distribuir la clave pública al servidor remoto permite que, al iniciar sesión, el servidor verifique que la clave privada correspondiente está en posesión del usuario. Esto elimina la necesidad de enviar contraseñas a través de la red y mejora significativamente la seguridad.

Métodos para distribuir la clave

- Uso de ssh-copy-id:
  La forma más sencilla es utilizar el comando ssh-copy-id, que copia la clave pública al archivo authorized_keys del usuario en el servidor remoto.

```
ssh-copy-id usuario@direccion_del_servidor
```

- Este comando añade la clave pública al archivo ~/.ssh/authorized_keys del usuario remoto.
- **Copia manual:** Abre la clave pública:

```
cat ~/.ssh/id_ed25519.pub
```

- Conéctate al servidor remoto y edita (o crea) el archivo ~/.ssh/authorized_keys, pegando la clave pública en una nueva línea.

- **Verificación de la distribución:** Una vez copiada la clave, prueba conectarte al servidor remoto:

```
ssh usuario@direccion_del_servidor
```

    - Si la autenticación se realiza sin pedir contraseña (o solo la passphrase, si la configuraste), la distribución fue exitosa.

Gestión de claves SSH

Buenas prácticas

1. **Protección de la clave privada:**
   - Nunca compartas tu clave privada. Su seguridad es fundamental para mantener la integridad del acceso.
   - Guarda la clave en un lugar seguro y utiliza passphrase para agregar una barrera extra en caso de pérdida o robo.
2. **Uso de ssh-agent:**
   - El ssh-agent permite almacenar las claves en memoria, evitando que tengas que introducir la passphrase cada vez que inicias sesión.
   - Inicia el agente:

```
eval "$(ssh-agent -s)"
```

   - Añade tu clave:

```
ssh-add ~/.ssh/id_ed25519
```

3. **Gestión del archivo ~/.ssh/config:**
   - Configura un archivo ~/.ssh/config para simplificar las conexiones a diferentes servidores.
     Ejemplo: plaintext

     Host servidor_remoto

```
    HostName direccion_del_servidor
    User usuario
    Port 22
    IdentityFile ~/.ssh/id_ed25519
```

Esto permite conectarte simplemente con:

```
ssh servidor_remoto
```

4. **Rotación y revocación de claves:**
   - Regularmente, revisa las claves autorizadas en cada servidor.
   - En caso de comprometerse una clave, remuévela del archivo authorized_keys y genera una nueva.

Actividad guiada: Distribución de la clave pública a un servidor remoto

1. **Iniciar la máquina virtual con Rocky Linux.**
2. **Generar un par de claves SSH (si no las tienes):**

```
ssh-keygen -t rsa -b 4096
```

Esto generará dos archivos en ~/.ssh/:

- id_rsa → Clave privada (NO compartir).
- id_rsa.pub → Clave pública (se copia al servidor).

3. **Verificación: Conéctate al servidor remoto**

- Copia el archivo de llave a la máquina Windows tutora (puedes usar WinSCP para esto).
- Utilizar Putty desde la máquina Windows tutora agregando la llave generada.
- Observa si la conexión se establece sin necesidad de ingresar la contraseña del sistema remoto (o solo la passphrase, si la configuraste).

¡Manos a la obra! - Configuración del ssh-agent y el archivo de configuración

Iniciar ssh-agent:

```
eval "$(ssh-agent -s)"
```

Agregar tu clave:

```
ssh-add ~/.ssh/id_ed25519
```

Configurar ~/.ssh/config:
Crea o edita el archivo `~/.ssh/config` con:

```
Host mi_servidor
    HostName direccion_del_servidor
    User usuario
    IdentityFile ~/.ssh/id_ed25519
```

Prueba la conexión: Conéctate usando el alias definido:

```
ssh mi_servidor
```

Reflexiona:

- ¿Qué ventajas ofrece la autenticación mediante claves SSH frente al uso de contraseñas?
- ¿Por qué es crucial mantener la clave privada en un entorno seguro y con passphrase?
- ¿Cómo influye la correcta distribución y gestión de claves en la seguridad global de un sistema?
- ¿Qué estrategias podrías implementar para la rotación periódica y revocación de claves en un entorno multiusuario?

La gestión de claves SSH es un pilar fundamental en la administración segura de sistemas remotos. Mediante la correcta creación de un par de claves, su distribución adecuada y una gestión continua (utilizando herramientas como ssh-agent y configuraciones personalizadas en ~/.ssh/config), se minimiza el riesgo de accesos no autorizados y se optimiza la experiencia de conexión.