¿Qué es SSH?

SSH (Secure Shell) es un protocolo que permite conectarse de forma segura a otros computadores a través de una red. A diferencia de métodos antiguos, SSH cifra toda la comunicación para proteger tus datos.

Lo que hace especial a SSH es su sistema de autenticación mediante claves:

1. La **clave privada**: Es como tu llave personal secreta que solo tú tienes y nunca compartes.
2. La **clave pública**: Es como un candado especial que puedes distribuir libremente a cualquier servidor al que quieras conectarte.

Cuando ambas claves se combinan correctamente, se confirma tu identidad sin necesidad de enviar contraseñas a través de la red.

Imagina que SSH funciona como el antiguo sistema de comunicación secreta de los espartanos llamado "escítala":

1. Tú (el cliente) tienes un bastón de madera con un diámetro específico (clave privada) que solo posees tú.
2. Tu amigo (el servidor) también tiene un bastón del exactamente mismo diámetro (conoce tu clave pública).
3. Cuando quieres enviar un mensaje secreto, enrollas una tira de cuero alrededor de tu bastón y escribes el mensaje a lo largo del bastón.
4. Al desenrollar la tira de cuero, las letras quedan dispersas y el mensaje se vuelve ilegible para cualquiera que lo intercepte.
5. Solo alguien que tenga un bastón del mismo diámetro exacto (tu clave pública) podrá enrollar la tira correctamente y leer el mensaje original.
6. Cuando tu amigo recibe la tira de cuero, la enrolla en su bastón idéntico y puede leer el mensaje perfectamente.

Esta antigua técnica criptográfica espartana representa perfectamente el concepto de SSH: solo quien posee la "medida correcta" (clave correspondiente) puede descifrar el mensaje, y cualquiera que intercepte la comunicación solo verá caracteres desordenados sin sentido.

Funcionamiento de SSH

Distribución de clave pública

La clave pública se copia al servidor donde queremos conectarnos

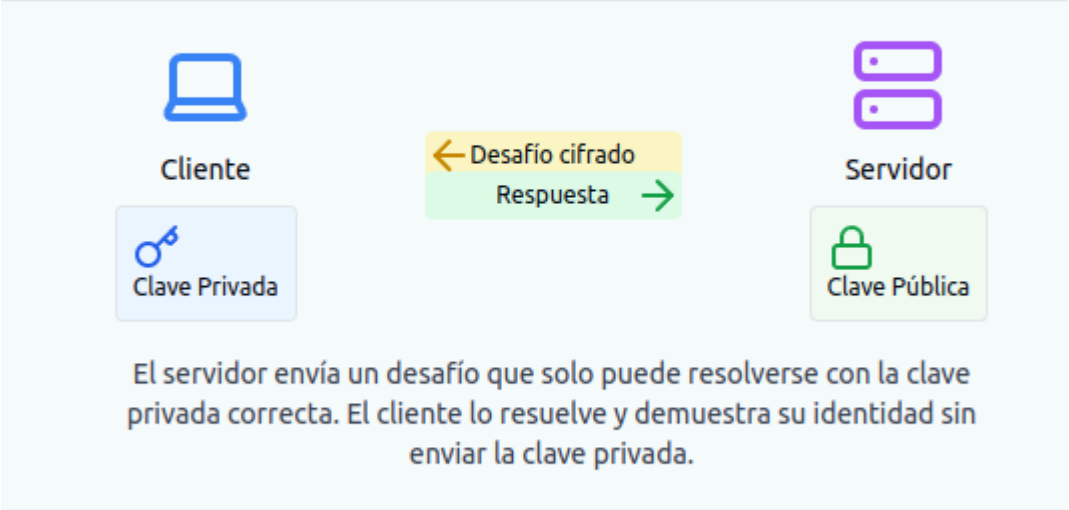
Funcionamiento de SSH

Conexión SSH

El cliente inicia una conexión SSH hacia el servidor

Funcionamiento de SSH

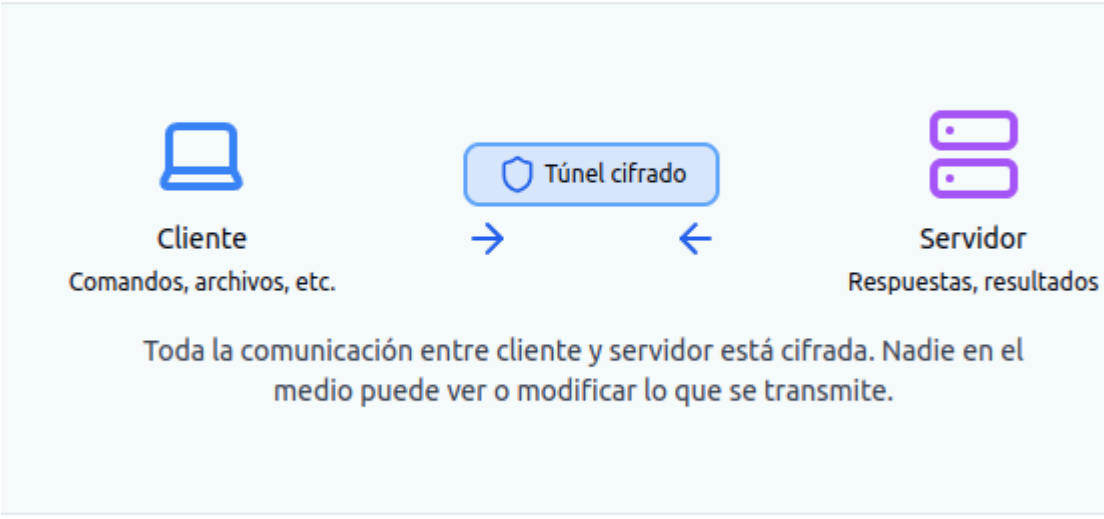
Autenticación



El servidor envía un desafío que solo puede resolverse con la clave privada correspondiente

Funcionamiento de SSH

Comunicación segura



Una vez autenticado, se establece un túnel cifrado para toda la comunicación

Ventajas y desventajas: SSH vs. autenticación por contraseña

Aspecto	Autenticación por clave SSH	Autenticación por contraseña
Seguridad	Mayor seguridad al no transmitir contraseñas por la red	Vulnerable a ataques de fuerza bruta y captura de paquetes
Facilidad de uso	Requiere configuración inicial, pero después permite acceso sin contraseña	Más intuitivo inicialmente para usuarios no técnicos
Automatización	Ideal para procesos automatizados y scripts	Difícil de automatizar sin comprometer la seguridad
Revocación	Requiere eliminar la clave del servidor para revocar acceso	Cambiar la contraseña revoca acceso inmediatamente
Portabilidad	Necesita tener el archivo de clave privada disponible	Solo requiere recordar la contraseña
Resistencia a ataques	Prácticamente imposible de descifrar con la tecnología actual	Susceptible a contraseñas débiles y ataques de diccionario
Gestión	Más compleja para múltiples usuarios y servidores	Más sencilla con sistemas centralizados de autenticación
Auditoría	Permite identificar claramente qué clave se utilizó	Más difícil rastrear quién conoce una contraseña compartida

Ejemplo práctico

Vamos a crear un par de claves SSH en una máquina Ubuntu y luego usarlas para conectarnos desde otra terminal. Este proceso es muy sencillo y te permitirá acceder sin contraseña a tu servidor.

Nota importante sobre modo bridge: Recuerda que la máquina virtual donde está Ubuntu debe estar configurada en modo bridge para que obtenga una dirección IP directa en tu red. Esto se cambia en la configuración de la máquina virtual antes de iniciarla, en la sección de configuración de red.

En el servidor Ubuntu (donde queremos conectarnos):

1. Abre una terminal y genera un par de claves SSH:

```
ssh-keygen -t rsa
```

2. Presiona Enter para aceptar la ubicación predeterminada (esto guardará las claves en la carpeta ~/.ssh/).
3. Presiona Enter dos veces para crear la clave sin contraseña (para hacerlo más sencillo).
4. Ahora tienes dos archivos:
    - `~/.ssh/id_rsa` (tu clave privada - nunca la compartas)
    - `~/.ssh/id_rsa.pub` (tu clave pública - esta sí se comparte)
5. Asegúrate de que el servidor SSH esté instalado:

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

6. Verifica la dirección IP de tu servidor Ubuntu:

```
ip addr show
```

Anota la dirección IP (normalmente comienza con 192.168.x.x).

Preparando el archivo authorized_keys:

1. En el servidor, crea el directorio ~/.ssh si no existe y establece permisos correctos:

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

2. Crea o abre el archivo authorized_keys:

```
nano ~/.ssh/authorized_keys
```

3. Ahora necesitamos obtener el contenido de la clave pública. En la misma máquina, ejecuta:

```
cat ~/.ssh/id_rsa.pub
```

4. **IMPORTANTE**: Copia TODO el texto que aparece (comienza con "ssh-rsa" y termina con algo como "usuario@maquina").
5. Pega este contenido en el archivo authorized_keys abierto con nano.
6. Guarda el archivo presionando Ctrl+O, luego Enter, y sal con Ctrl+X.
7. Establece los permisos correctos:

```
chmod 600 ~/.ssh/authorized_keys
```

En el cliente (desde donde nos conectaremos):

Para usuarios de Linux/Mac:

1. Si es otra máquina, necesitas crear una copia de la clave privada. En el servidor, muestra el contenido de la clave privada:

```
cat ~/.ssh/id_rsa
```

2. Copia TODO el contenido (desde "-----BEGIN OPENSSH PRIVATE KEY-----" hasta "-----END OPENSSH PRIVATE KEY-----").
3. En la máquina cliente, crea el directorio ~/.ssh:

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

4. Crea un archivo para la clave privada:

```
nano ~/.ssh/id_rsa
```

5. Pega el contenido copiado y guarda (Ctrl+O, Enter, Ctrl+X).
6. Establece los permisos correctos:

```
chmod 600 ~/.ssh/id_rsa
```

7. Ahora puedes conectarte simplemente con:

```
ssh usuario@IP_SERVIDOR
```

Para usuarios de Windows:

- **PuTTY**: Descárgalo de la página oficial. Usa PuTTYgen para importar o crear claves.
    1. Abre PuTTYgen
    2. Selecciona "Load" y carga la clave privada
    3. Guarda la clave en formato PuTTY (.ppk)
    4. Usa PuTTY para conectarte indicando la clave en Connection > SSH > Auth
- **Git Bash**: Una alternativa más sencilla si ya tienes Git instalado.
    1. Sigue los mismos pasos que en Linux/Mac
    2. Git Bash proporciona los mismos comandos ssh