Taller: Acceso remoto por SSH con certificados

SSH (*Secure Shell*) es un protocolo de red que utiliza técnicas criptográficas para proporcionar un canal de comunicación seguro sobre una red no segura en una arquitectura cliente-servidor. El puerto TCP asignado al servicio SSH es el 22.

SSH se utiliza, principalmente, para acceder de forma segura a un servidor remoto. Además de este uso, también se utiliza muy a menudo para transferir datos entre cliente y servidor de forma segura, gestionar claves RSA que permitan el acceso al servidor sin introducir la contraseña y redirigir los datos de cualquier aplicación utilizando un túnel seguro.

Contexto

Autenticación con clave asimétrica sin certificado

El acceso a SSH sin contraseña utilizando claves asimétricas tiene claras ventajas. No obstante, también presenta algunos problemas relacionados con la usabilidad, la operatividad y la seguridad.

En la autenticación con claves asimétricas sin certificados, un sistema o es un cliente, o es un servidor. No hay más roles en juego. El cliente inicia una conexión SSH con el servidor. El proceso `sshd` se ejecuta en el servidor.

El servidor demuestra su identidad presentando su clave pública. El cliente puede que ya tenga una copia de la clave del servidor en su fichero `known_hosts`, en cuyo caso confía en el servidor y acepta la autenticación. Si la clave del servidor todavía no está en `known_hosts`, el cliente pregunta al usuario si debe confiar en el servidor:

```
usuario@ubuntu-desktop:~$ ssh server
The authenticity of host 'server (172.17.1.27)' can't be established. ECDSA
key fingerprint is SHA256:v37rXqfIEvfv8LOBqZkOUxYhZQgJ0WSfavS43EsSTS4. Are you
sure you want to continue connecting (yes/no)?
```

La reacción habitual es escribir «*yes*» sin comprobar la huella. La aceptación incondicional de la solicitud de autenticación del servidor es una de las debilidades del protocolo de clave pública no certificada.

El cliente identifica al usuario enviando un mensaje firmado con la clave privada del usuario al servidor. El servidor acepta la autenticación si la clave pública del usuario se encuentra en su archivo `authorized_keys`. Esto resalta otras dos debilidades: ¿Cómo pueden los usuarios agregar sus claves a `authorized_keys` si no tienen acceso por contraseña? Y, ¿cómo administrar los cientos de copias de claves que se acumularán en un centro de datos?

Otra debilidad: Las claves nunca expiran. Ni los usuarios ni los equipos están obligados a actualizar sus claves.

Estos problemas en realidad no son inherentes a SSH, sino al procedimiento de autenticación por clave asimétrica no certificada. Los **certificados SSH** son una solución más adecuada para este escenario.

Cambio de enfoque: autenticación con certificados

Cuando se utilizan certificados, las claves de usuario no se almacenan en el archivo `authorized_keys` del servidor y las claves de servidor no se almacenan en el archivo `known_hosts` del cliente. Así es que, ¿cómo se validan las claves?

Una tercera entidad entra en juego: el servidor de confianza o sistema de control. En el servidor de confianza se encuentra una clave privada que actuará como Autoridad de Certificación (AC), que se utiliza para firmar digitalmente las claves de usuario y de equipo. El resultado del proceso de firma se denomina certificado.

Para autenticarse, el cliente y el servidor no intercambian claves sino certificados que se validan utilizando la clave pública de la AC.

De esta forma, se resuelven algunas de las debilidades vistas anteriormente e inherentes a la autenticación de clave pública no certificada:

- los certificados tienen una fecha de vencimiento;
- no es necesario copiar las claves de usuario a los servidores y administrar copias de claves, porque las claves de usuario no se almacenan en los servidores.

Existe un inconveniente: los usuarios no pueden firmar sus claves públicas por sí mismos, ya que no tienen acceso a la clave privada de la AC. Debe existir un proceso para obtener claves firmadas por un operador privilegiado o una aplicación.

> ***Nota**: La certificación del servidor OpenSSH sigue el estándar OpenPGP, no SSL/TLS, y el formato del certificado es OpenPGP, no X509.*

Configuración del escenario de prácticas

Todos los servidores se virtualizarán en VirtualBox o en otro sistema de virtualización. Si se utilizan instancias en AWS hay que tener la precaución de no perder nunca el acceso a la máquina.

Servidor de confianza (AC)

Las claves del servidor de confianza se deben crear y almacenar en un entorno seguro, tal como se haría con cualquier otra clave privada. En esta práctica se utilizará el usuario `root` en una máquina virtual, pero en un entorno real se recomienda utilizar una cuenta de usuario con privilegios administrativos en equipo sin conexión a la red.

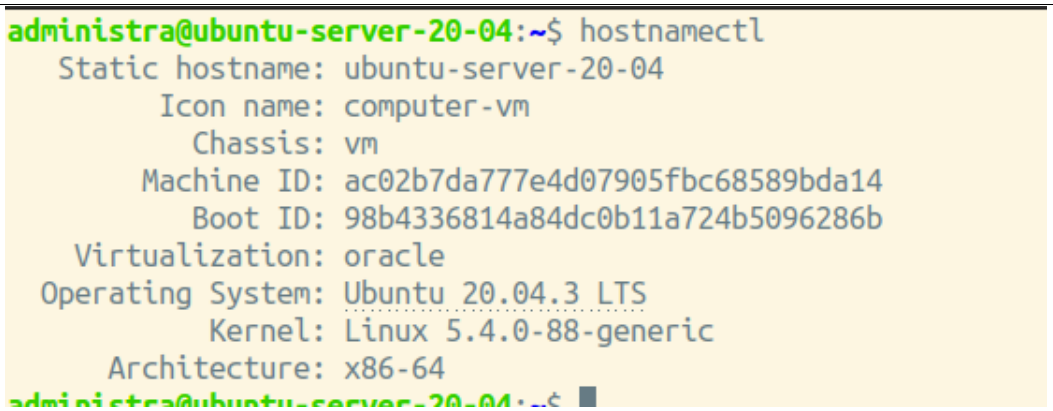
```
administrador@orion:~$ hostnamectl
   Static hostname: orion
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 6dcc4dac4b494927af75470c232ef2bd
           Boot ID: ea3ede2349d34780a1fc0d539b4ffd27
    Virtualization: oracle
  Operating System: Ubuntu 20.04.3 LTS
            Kernel: Linux 5.4.0-88-generic
      Architecture: x86-64
administrador@orion:~$ _
```

Servidor #1

Usuarios permitidos para acceder por SSH: `usuari` (con privilegios de administrador), `profesor` (con privilegios de administrador), `alumno`, `alumno01`.

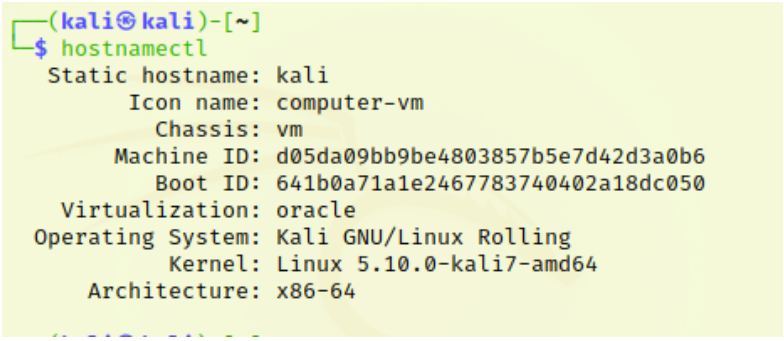
```
administra@ubuntu-server-20-04:~$ hostnamectl
   Static hostname: ubuntu-server-20-04
Transient hostname: server.20.04
         Icon name: computer-vm
           Chassis: vm
        Machine ID: ac02b7da777e4d07905fbc68589bda14
           Boot ID: 98b4336814a84dc0b11a724b5096286b
    Virtualization: oracle
  Operating System: Ubuntu 20.04.3 LTS
            Kernel: Linux 5.4.0-88-generic
      Architecture: x86-64
administra@ubuntu-server-20-04:~$
```

Para poder usar el nombre de la máquina se ha definido una entrada en el /etc/hosts con la Ip de la máquina asociada a ese nombre.



Cliente

Se puede utilizar cualquier usuario y cualquier sistema para actuar como cliente: un equipo con Windows 10 utilizando el cliente integrado, los propios servidores o cualquier otro equipo.



Acceso remoto por SSH con certificados

Los errores más comunes cuando se generan certificados SSH son:

- Firmar de claves de forma incorrecta, lo que conduce a certificados no válidos.
- Realizar los pasos en el sistema incorrecto. En el proceso están involucrados tres sistemas diferentes: el cliente, el servidor y un tercer sistema al que nos referiremos como sistema de controlo servidor de confianza. Confundir unos con otros puede ser fácil.

Preparación del sistema de control

En el sistema de control se generarán un par de claves para firmar. Si se desea, se puede definir una contraseña. PASSPHRASE: miclavesecreta

```
root@orion:/# mkdir -p /etc/ssh/ac
root@orion:/# ssh-keygen -f /etc/ssh/ac/ac-brs -t ed25519 -C "AC-BRS-SSH"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ac/ac-brs
Your public key has been saved in /etc/ssh/ac/ac-brs.pub
The key fingerprint is:
SHA256:/BnosRCQgW5Y/aaVM/FOU3O3tVOXQKoPi4ZaJnwLdqw AC-BRS-SSH
The key's randomart image is:
+--[ED25519 256]--+
|   ooo      .o...|
|  o o..   o o . +|
| +   ..+ . + . o=|
|. o   Bo+..    ...|
| .   +.=S+.      |
|  . o .oo++o     |
|   = B oo.o.     |
|  . X o          |
|   E .           |
+----[SHA256]-----+
```

Esto genera dos ficheros: la clave privada `ac-brs` y la clave pública `ac-brs.pub` que se pueden almacenar en cualquier ubicación dentro del servidor de confianza.

- La clave privada será la que se utilice para firmar los certificados. Obviamente, los usuarios sin privilegios no deben poder firmar sus propias claves, así es que debe quedar bien custodiada.
- La clave pública se distribuirá por los diferentes equipos, tanto los clientes como los servidores, para que puedan validar los certificados que se presentan.

En algunos entornos puede ser interesante disponer de dos pares de claves[1]: uno se utilizará para firmar y validar los certificados de los equipos (`ac_host_key-brs`) y el otro para los certificados de los usuarios (`ac_user_key-brs`).

1 La documentación de Red Hat Enterprise Linux incluida en el apartado *Referencias* recomienda esta configuración.

Certificados de servidor

Paso 1: Firmar la clave pública del equipo

Cuando se realiza la instalación del servidor OpenSSH se generan unos pares de claves para el servidor.

- En algunos casos, es recomendable eliminar las claves del servidor OpenSSH que se generaron durante la instalación y generar nuevos pares de claves solo para los algoritmos que se vayan a utilizar[2].
- Los algoritmos recomendados son:
  - RSA con una longitud mínima de 3072 bits. Recomendado para mantener compatibilidad.
  - Ed25519 con una longitud mínima de 256 bits. Actualmente, el más recomendado.

```
administra@ubuntu-server-20-04:~$ ls -l /etc/ssh/ssh_host_*
-rw------- 1 root root  227 jun 24  2020 /etc/ssh/ssh_host_ecdsa_key
-rw-r--r-- 1 root root  185 jun 24  2020 /etc/ssh/ssh_host_ecdsa_key.pub
-rw------- 1 root root  419 jun 24  2020 /etc/ssh/ssh_host_ed25519_key
-rw-r--r-- 1 root root  105 jun 24  2020 /etc/ssh/ssh_host_ed25519_key.pub
-rw------- 1 root root 1675 jun 24  2020 /etc/ssh/ssh_host_rsa_key
-rw-r--r-- 1 root root  405 jun 24  2020 /etc/ssh/ssh_host_rsa_key.pub
administra@ubuntu-server-20-04:~$
```

Se pueden utilizar estas claves o generar unas nuevas si se desea. No se debe poner contraseña a la clave privada.

- Si se crean las claves en el equipo que actúa como servidor OpenSSH, se copiará la clave pública en el equipo que actúa como sistema de control.
- Si se crean las claves en el equipo que actúa como sistema de control, se copiarán las dos claves en el equipo que actúa como servidor y no se almacenará la clave privada.

```
root@orion:/etc/ssh/ac# mkdir -p /etc/ssh/ac_
```

```
root@orion:/etc/ssh/ac# scp usuari@ubuntu-server-20-04:/etc/ssh/ssh_host_ed25519_key.pub \
> /etc/ssh/ac/hosts/ubuntu-server20-04/
usuari@ubuntu-server-20-04's password:
ssh_host_ed25519_key.pub                      100%  105    52.5KB/s   00:00
root@orion:/etc/ssh/ac#
```

En el servidor de confianza, se utiliza la clave privada (almacenada en el fichero `ac-brs`) para firmar la clave pública de cada servidor que se encuentre en el centro de datos.

```
root@orion:/# ssh-keygen -s /etc/ssh/ac/ac-brs \
> -I ubuntu-server-20-04 \
> -n ubuntu-server-20-04,server.20.04 \
> -V -5d:+52w \
> -h \
> /etc/ssh/ac/hosts/ubuntu-server20-04/ssh_host_ed25519_key.pub
Enter passphrase:
Signed host key /etc/ssh/ac/hosts/ubuntu-server20-04/ssh_host_ed25519_key-cert.pub: id "ubuntu-serve
r-20-04" serial 0 for ubuntu-server-20-04,server.20.04 valid from 2021-11-04T09:19:33 to 2022-11-08T
09:19:33
```

- La opción `-h` indica que se van a firmar claves de equipo.
- La opción `-s` especifica la clave privada que se utilizará para firmar.

2 Hay que tener en cuenta que los clientes que ya conectaron con el servidor ya habrán aceptado la huella de la clave pública y pueden interpretar este cambio como un ataque de intermediario (*Man-in-the-Middle Attack*).

- La opción `-I` define la identidad del certificado, habitualmente el FQDN del equipo para el que se emite el certificado. Puede ser cualquier cadena alfanumérica.
- La opción `-n` especifica uno o más principales (nombres de equipo a los que se accederá) que se incluirán en el certificado. Si se incluye más de un principal se separarán por comas.
- La opción `-V` es el intervalo de tiempo para el que el certificado será válido; en el ejemplo, desde hace 5 días hasta dentro de 52 semanas.
- `ssh_host_ed25519_key.pub` es la clave pública del servidor que se quiere firmar (con su ruta absoluta).

El resultado de este comando es un nuevo archivo llamado `server_key-cert.pub`. Este es el certificado que utilizará el servidor para demostrar su identidad a partir de ahora. Se copiará en el directorio `/etc/ssh` del servidor OpenSSH.

```
root@orion:~# scp \
> /etc/ssh/ac/hosts/ubuntu-server20-04/ssh_host_ed25519_key-cert.pub \
> usuari@ubuntu-server-20-04:~
```

```
usuari@server-20:~$ sudo chown root:root ssh_host_ed25519_key-cert.pub
usuari@server-20:~$ sudo mv ssh_host_ed25519_key-cert.pub /etc/ssh
```

Paso 2: Configuración del servidor OpenSSH

Los servidores OpenSSH deben saber que han de presentar sus certificados de equipo a los clientes. Para ello, se agrega la cláusula `HostCertificate` en el fichero `sshd_config` del servidor:

```
HostCertificate /etc/ssh/ssh_host_ed25519_key-cert.pub
```

Si se ha generado una clave nueva diferente de las que el servidor OpenSSH generó durante la instalación, también hay que agregar la cláusula `HostKey`, (**sino, no hace falta**):

```
HostKey /etc/ssh/ssh_host_ed25519_key-cert.pub
HostCertificate /etc/ssh/ssh_host_ed25519_key-cert.pub
```

Después de la configuración se debe reiniciar el servicio OpenSSH.

```
usuario@server-20:~$ sudo systemctl restart ssh
```

Paso 3: Indicar a los clientes que confíen en el servidor de confianza (la AC)

Cuando se utilizan claves públicas no certificadas, los clientes conservan las claves de los equipos en las que confían en `known_hosts`. Por ejemplo, una línea del estilo:

```
server ecdsa-sha2-nistp256 AAAAE2VjZHN.....
```

le dice al cliente que el servidor `server` debe presentar la clave de host que comienza con `ecdsa-sha2-nistp256`. Si el servidor presenta una clave de host diferente, el cliente se negará a conectar.

Cuando se utilizan certificados, el servidor presenta al cliente una huella de su propia clave pública junto con la huella de la clave pública de la AC.

```
usuario@ubuntu-server-20-04:~$ ssh-keygen -L -f /etc/ssh/ssh_host_ed25519_key-cert.pub
```

```
usuari@ubuntu-server-20-04:~$ ssh-keygen -L -f /etc/ssh/ssh_host_ed25519_key-cert.pub
/etc/ssh/ssh_host_ed25519_key-cert.pub:
        Type: ssh-ed25519-cert-v01@openssh.com host certificate
        Public key: ED25519-CERT SHA256:kPSZvsqr1iKpiiHYLl9JTF3dx8TxmLvlqIlM4MqmZhA
        Signing CA: ED25519 SHA256:/BnosRCQgW5Y/aaVM/FOU3O3tV0XQKoPi4ZaJnwLdqw (using ssh-ed25519)
        Key ID: "ubuntu-server-20-04"
        Serial: 0
        Valid: from 2021-11-04T13:45:15 to 2022-11-08T13:45:15
        Principals:
                ubuntu-server-20-04
                server.20.04
        Critical Options: (none)
        Extensions: (none)
```

Ubuntu Software

Si el cliente confía en la AC, en la primera conexión no se muestra la huella del servidor para aceptarla y no se añadirá ninguna línea para identificar al equipo en el fichero `known_hosts`. En este fichero se pueden almacenar de forma conjunta las claves de los equipos que no utilizan certificados junto con la clave pública de la AC para los equipos que sí utilizan certificados.

En todos los clientes se deben eliminar las líneas de clave de equipo de `known_hosts` para los servidores que utilizarán certificados a los que se haya accedido anteriormente. Se añadirá una línea nuevaque contenga la clave pública de la AC.

```
@cert-authority LISTA-DE-SERVIDORES ssh-ed25519 AAAAC3NzaC1lZDI1N...
```

- *LISTA-DE-SERVIDORES* es una lista separada por comas de los servidores que firmaron su clave de host. Se permiten comodines. Por ejemplo `ubuntu-server-20-04,*.dominio.lan`.
- La cadena que comienza con `ssh-ed25519` es la clave pública de la AC. Si se ha copiado en cada uno de los clientes,

```
┌──(kali㉿kali)-[~]
└─$ scp administrador@192.168.1.75:/etc/ssh/ac/ac-brs.pub ./
The authenticity of host '192.168.1.75 (192.168.1.75)' can't be established.
ECDSA key fingerprint is SHA256:C4bJSCNnJJiAopMz6rHuZjMG7QSdYsdEtefxwO2UHvQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.75' (ECDSA) to the list of known hosts.
administrador@192.168.1.75's password:
ac-brs.pub                                    100%   92    22.5KB/s   00:00
```

- se puede añadir así:

```
kali@kali:~$ echo @cert-authority ubuntu-server-20-04,*.domini.lan \
> $(cat ac-brs.pub) | tee -a .ssh/known_hosts

@cert-authority ubuntu-server-20-04,*.domini.lan ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIBb+pvHY1fQ9PJFhad+caFB+F4uVFy+5xHnllDoMPe4T AC-BRS-SSH
```

```
┌──(kali㉿kali)-[~/.ssh]
└─$ cat known_hosts
@cert-authority ubuntu-server-20-04,*.domini.lan ssh-ed25519 AAAAC3NzaC1lZDI1
NTE5AAAAIBb+pvHY1fQ9PJFhad+caFB+F4uVFy+5xHnllDoMPe4T AC-BRS-SSH
```

Cada cuenta de usuario dispone de su propio fichero `~/.ssh/known_hosts`. Además, existe el fichero `/etc/ssh/ssh_known_hosts` que se puede utilizar para que todos los usuarios del equipo confíen en la AC.

A partir de ahora, al establecer una conexión a cualquier servidor incluido en la lista de servidores que utilizan certificados el cliente utiliza la clave pública de la AC para validar la clave del equipo.

Prueba del certificado de equipo

En un cliente que confía en la AC (tal como se ha configurado en el *Paso 3: Indicar a los clientes que confíen en el servidor de confianza (la AC)*) se establece una conexión `ssh` a uno de los servidores con clave de equipo firmada. La prueba no se pasa si ve un mensaje como:

```
The authenticity of host 'ubuntu-server-20-04 (172.17.1.13)' can't be established.
ED25519 key fingerprint is SHA256:v37rXqfIEvfv8LOBqZkOUxYhZQgJ0WSfavS43EsSTS4.
Are you sure you want to continue connecting (yes/no[fingerprint])?
```

Los certificados de equipo no requieren que el usuario confirme que este es el servidor correcto. Se puede ejecutar `ssh -vvv` para averiguar qué está mal configurado.

```
┌──(kali㉿kali)-[~/.ssh]
└─$ ssh -vvv usuari@ubuntu-server-20-04
OpenSSH_8.4p1 Debian-5, OpenSSL 1.1.1k  25 Mar 2021
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug3: expanded UserKnownHostsFile '~/.ssh/known_hosts' ->
'/home/kali/.ssh/known_hosts'
.....
.....
debug1: Authenticating to ubuntu-server-20-04:22 as 'usuari'
debug3: hostkeys_foreach: reading file "/home/kali/.ssh/known_hosts"
debug3: record_hostkey: found ca key type ED25519 in file /home/kali/.ssh/known_hosts:1
.....
debug3: load_hostkeys: loaded 1 keys from ubuntu-server-20-04
debug1: Host 'ubuntu-server-20-04' is known and matches the ED25519-CERT host
certificate.
debug1: Found CA key in /home/kali/.ssh/known_hosts:1
debug1: Authentication succeeded (password).
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)
.....
```

Certificados de usuario

Paso 1: Firmar la clave pública del usuario

El usuario debe disponer de una clave asimétrica para obtener su certificado firmado por la AC. Esta clave puede tener contraseña, aunque si se utiliza para tareas de automatización es conveniente dejarla sin contraseña.

```
kali@kali:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/kali/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase): miclavesecreta
Enter same passphrase again: miclavesecreta
Your identification has been saved in /home/kali/.ssh/id_ed25519
Your public key has been saved in /home/kali/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:UHVSS3tlDSAUh+TIvIOBH3HmNf3reS2vgRj9fVxtJP4 kali@kali
The key's randomart image is:
+--[ED25519 256]--+
|       . ++X=*...+|
|      . O +.B.o o.|
|     . + = . o.o .|
|      . = . . o.o.|
|       o S . . ..+|
|          . o o.=.|
|           . ..o.E|
|    .  o.         |
+----[SHA256]
```

Los usuarios sin privilegios no tienen acceso a la clave privada de la AC (en el sistema de control o servidor de confianza) así es que no pueden firmar por sí mismos sus claves. Se requiere un administrador o una aplicación especial que realice esta operación en el servidor de confianza, en el que **se habrá copiado la clave pública del usuario en el directorio /etc/ssh/ac/users/kali (si no existe habrá que crearlo)**

```
root@orion:~# ssh-keygen -s /etc/ssh/ac/ac-brs \
> -I kali \
> -n usuari,profesor,alumno01 \
> -V -5d:+52w \
> /etc/ssh/ac/users/kali/id_ed25519.pub
Enter passphrase: miclavesecreta
```

```
Enter passphrase:
Signed user key /etc/ssh/ac/users/kali/id_ed25519-cert.pub: id "kali" serial 0 for usuari,profesor,a
lumno01 valid from 2021-11-06T18:57:51 to 2022-11-10T18:57:51
```

- La opción `-h` no se utiliza, dado que no se desea firmar una clave de equipo sino de usuario.
- La opción `-I` define la identidad del certificado, habitualmente el nombre de usuario en el dominio para quien se emite el certificado. Puede ser cualquier cadena alfanumérica.
- La opción `-n` especifica uno o más principales (nombres de usuarios) que se incluirán en el cer tificado. Si se incluye más de un principal se separarán por comas.
- `id_ed25519.pub` es la clave pública del usuario que se quiere firmar.

El resultado del comando es un certificado con nombre `id_ed25519-cert.pub`. Debe almacenar se en el cliente.

```
┌──(kali㉿kali)-[~/.ssh]
└─$ scp administrador@192.168.1.75:/etc/ssh/ac/users/kali/id_ed25519-cert.pub
 ./
administrador@192.168.1.75's password:
id_ed25519-cert.pub                           100%  668   585.2KB/s   00:00

┌──(kali㉿kali)-[~/.ssh]
└─$ ls -la
total 24
drwx------  2 kali kali 4096 nov 11 20:09 .
drwxr-xr-x 15 kali kali 4096 nov 11 20:07 ..
-rw-------  1 kali kali  444 nov 11 19:33 id_ed25519
-rw-r--r--  1 kali kali  668 nov 11 20:09 id_ed25519-cert.pub
-rw-r--r--  1 kali kali   91 nov 11 19:33 id_ed25519.pub
-rw-r--r--  1 kali kali  363 nov 11 19:47 known_hosts
```

***Nota a título informativo, no hace falta ejecutarlo**: Es posible firmar utilizando la clave de una Autoridad de Certificación almacena- da en un dispositivo PKCS#11[3] proporcionando con la opción `-D` la biblioteca de tokens y con la opción `-s` la mitad pública de la clave de la AC,*

```
ssh-keygen -s ca_key.pub -D libpkcs11.so -I key_id user_key.pub
```

Paso 2: Instalar la clave pública de la AC en los servidores

Para permitir que un servidor valide un certificado de usuario, debe tener la clave pública de la AC. La clave `ac-brs.pub` del sistema de control se deberá copiar en el directorio `/etc/ssh` de cada servidor.

Paso 3: Configuración de OpenSSH

En cada servidor, se agrega la cláusula `TrustedUserCAKeys` en el fichero `sshd_config`. Se debe reiniciar el servicio.

```
TrustedUserCAKeys /etc/ssh/ac-brs.pub
```

Prueba de certificado de usuario

En cada uno de los usuarios de los servidores a los que ya se accedió sin certificado se eliminará la clave del usuario del fichero `authorized_keys`. También se puede renombrar para mantener una copia del fichero por si algo va mal.

Desde un cliente que confíe en la CA, se establecerá una conexión `ssh` con el servidor. Aunque la clave ya no exista en `authorized_keys`, se debería poder iniciar sesión sin contraseña para cualquier usuario que se haya incluido en el certificado. Los usuarios que no estén incluidos en el certificado no podrán conectar

```
kali@kali:~$ ssh ubuntu-server-20-04 -l alumno
alumno@ubuntu-server-20-04's password:

kali@kali:~$ ssh ubuntu-server-20-04 -l alumno01
Enter passphrase for key '/home/kali/.ssh/id_ed25519': miclavesecreta
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)

alumno01@ubuntu-server-20-04:~$
```

Se puede comprobar que la conexión ha sido correcta revisando el registro de autenticación del servidor (`/var/log/secure` o `/var/log/auth.log`).

3 PKCS#11 define una API para la comunicación con tokens con seguridad criptográfica como tarjetas inteligentes, llaves USB y módulos de seguridad de hardware (HSMs). El DNI-e, en entornos GNU/Linux y macOS, hace uso de este módulo criptográfico.

Conclusión

La autenticación en SSH utilizando clave asimétrica sin certificados presenta una serie de problemas de usabilidad, operatividad y seguridad. Muchas de estas debilidades desaparecen o se mitigan lo suficiente cuando se utilizan certificados SSH.

Los errores más comunes cuando se generan certificados SSH son:

- Firmar de claves de forma incorrecta.
- Realizar los pasos en el sistema incorrecto.

Aunque la configuración de este taller se haya realizado con una única AC, no hay ningún impedimento para tener varias AC en un único entorno.

Referencias

- If you're not using SSH certificates you're doing SSH wrong – Mike Malone @ SmallStep (2019). https://smallstep.com/blog/use-ssh-certificates/
- Using SSH certificates. Pros and cons – Ramblings of a Unix Geek (2016). https://www.sweharris.org/post/2016-10-30-ssh-certs/
- Certificate-based SSH authentication – Umbriel Security (2017). https://umbrielsecurity.wordpress.com/2017/03/07/certificate-based-ssh-auth/
- Deployment Guide. Red Hat Enterprise Linux 6 (2017). https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/sec-using_openssh_certificate_authentication
- Página de manual de `ssh`.
- Página de manual de `ssh-keygen`.