

# authorized\_keys

contiene las claves públicas de los sistemas que permiten la conexión al sistema actual se le pueden añadir restricciones del tipo:

- `from=«xxx.xxx.xxx.xxx» ssh-dss...` → restricción por IP
- `no-pty ssh-dss...` → evita obtener una shell
  - se puede saltar ejecutando: `ssh sistema_remoto 'echo $HOSTNAME' * command=«echo 'pues va a ser que no'» ssh-dss...` → solo permite ejecutar el comando especificado
  - \* `no-port-forwarding ssh-dds...` → no permite port-forwarding \*
  - `permitopen«xxx.xxx.xxx.xxx:xxxx»` → limita el port-forwarding a una determinada IP y puerto
  - ¿entrar en un sistema que se conecta con otro y que ese usuario solo sirva para eso?
- `command=«/home/user/comando_autorizado»`

las restricciones separadas por comas y sin espacios

## creación y distribución de llaves

- crear llave 2048 bits:

```
ssh-keygen -b 2048 -f /home/user/.ssh/myKey
```

```
ssh-keygen -t rsa -f /home/user/.ssh/myKey
```

curva eliptica

```
ssh-keygen -t ed25519 -f /home/user/.ssh/myKey -C "comentario en clave"
```

- asegurarnos que el directorio `~/ .ssh` tiene los permisos 700
- copiar clave pública en sistema remoto (en el directorio `.ssh` de la home del usuario, añadiendo o creando el fichero `authorized_keys`, cambiando los permisos a 600)

```
ssh-copy-id -i /home/user/.ssh/myKey.pub user@sistema.remoto
```

- esto nos permite entrar en el sistema remoto sin recordar la contraseña de ese usuario en ese sistema. Solo tenemos que recordar la contraseña de nuestra clave privada

```
ssh -i<fichero_clave_privada> usuario@sistema.remoto
```

## recrear llave pública la privada

```
ssh-keygen -y -f privatekey.pem > publickey.pem
```

## obligar autenticación password

```
ssh -o PreferredAuthentications=password -o PubkeyAuthentication=no user@server
```

## known\_hosts

- este fichero contiene las claves públicas de los servidores a los que nos conectamos por SSH.
- por defecto, el nombre de los servidores se esconde con un hash
  - para cambiar eso, podemos editar `/etc/ssh/ssh_config` y modificar `HashKnownHosts Yes` → No
- comandos útiles para tratar con este fichero:
  - `ssh-keygen -R <ip|dominio> -f <known_hosts_file>` → solicita la clave pública de ese servidor y la elimina de la lista
  - `ssh-keyscan rsa,dsa <dominio>` → recupera las claves solicitadas (por pantalla)
    - `-H` → key en formato HASH ¿?
    - `-v` → verbose
  - `ssh-keygen -H -F <dominio>` → busca y muestra si hay la clave de un host
    - `-H` → busca
    - `-F` → en el fichero `known_hosts`
    - `<dominio>` no es parámetro de `-F`

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux:ssh:keys?rev=1604414910>

Last update: **03/11/2020 06:48**

