

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
```

- 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
```

From:

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

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux:ssh:keys?rev=1455122591>

Last update: **10/02/2016 08:43**

