



RSA: SISTEMAS DE AUTENTICACIÓN CLAVE PÚBLICA/PRIVADA.

SAD



2ºASIR
I.E.S. ANTONIO MACHADO
ANA OROZCO ASENSIO

Contenido

Introducción.....	2
1. SSH en servidor sin contraseña.....	2
2. Github sin contraseña por SSH.....	6
3. Mensaje cifrado.....	10

Introducción.

Realicé esta práctica en el ordenador de clase con Linux Mint 22 y máquina virtual Linux Mint 22 (no tenía Debian, por lo que aproveché la máquina de otra práctica), me referiré a esta máquina virtual como “servidor”, y los puntos 2 y 3 los realicé en mi máquina física en casa donde tengo sistema dual Linux Mint 22 y Windows 11.

1. SSH en servidor sin contraseña.

En el ordenador anfitrión vamos a generar claves SSH y para ello (además de tener instalado openssh) tendremos que usar el comando:

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

También existe un comando que genera una contraseña más fuerte ya que, en vez de 2048bits, es de 4096bits y es este comando:

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

```
ana@PC219:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ana/.ssh/id_rsa):
Created directory '/home/ana/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ana/.ssh/id_rsa
Your public key has been saved in /home/ana/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/+cdT4d1WxRxwQle04mynePuI+X9ygKZEI/BnawI044 ana@PC219
The key's randomart image is:
+---[RSA 3072]---+
|   . o . .0==|
|   0 . + + . oo=|
|   = . * . 0 +.|
|   E o + . + ...|
|       S. + + +|
|       .+ ...0+|
|       ..0.000|
|       oo+.++|
|       +*=.=|
+---[SHA256]---
```

Ahora vamos a copiar la clave pública al servidor (que en este caso es Linux Mint (la terminal es ana@ana-virtualbox y la IP 192.168.2.99).

ssh-copy-id remote_username@remote_IP_Address

```
ana@PC219:~$ ssh-copy-id ana@192.168.2.99
The authenticity of host '192.168.2.99 (192.168.2.99)' can't be established.
ED25519 key fingerprint is SHA256:0kxemp6ffpFEIwM4ibFSsl6YwDR8peJRAfr6Ep3Vds.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
ana@192.168.2.99's password:

Number of key(s) added: 1

Now try logging into the machine, with:    "ssh 'ana@192.168.2.99'"
and check to make sure that only the key(s) you wanted were added.
```

Al hacer esto haremos que no nos pida la contraseña al entrar con nuestro ordenador anfitrión al servidor porque guardará nuestra clave pública que nos autenticará.

Ahora me conectaré mediante el comando:

ssh usuario@ip

```
ana@PC219:~$ ssh ana@192.168.2.99

ana@ana-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP>
group default qlen 1000
    link/ether 08:00:27:b3:89:ce brd ff:ff:ff
    inet 192.168.2.99/21 brd 192.168.7.255 sc
        valid_lft forever preferred_lft forever
```

Como se observa en la imagen nos conectamos sin contraseña, para probar que estoy entrando a ese servidor usé un **ip a** para ver las ips..

Ahora vamos a cambiar unos archivos en la configuración del servidor para deshabilitar el login SSH con contraseña, para ello en la terminal escribiremos:

```
nano /etc/ssh/ssh_config
```

```
ana@ana-VirtualBox: ~/wordpress
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 7.2 /etc/ssh/ssh_config *
# 3. system-wide file
# Any configuration value is only changed the first time it is s
# Thus, host-specific definitions should be at the beginning of
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comp
# list of available options, their meanings and defaults, please
# ssh_config(5) man page.

Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no
# HostbasedAuthentication no
# GSSAPIAuthentication no

^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T E
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar      ^J J
```

En este archivo cambiaremos estas 3 líneas, para ello eliminamos la # para que no sea un comentario en el código.

Además del archivo ssh_config , tenemos que editar el archivo sshd_config.

Que además de las opciones que cambié en la imagen, tenemos otras opciones muy interesantes de configuración como por ejemplo dar acceso SSH solo a los usuarios que queramos mediante: (Esto no lo añadiré yo)

```
AllowUsers usuario1 usuario2
```

Aquí realmente cambiaremos las mismas opciones que ya cambiamos en el anterior paso, para entrar usaremos el siguiente comando:

nano /etc/ssh/sshd_config

```
ana@ana-VirtualBox: ~/wordpress
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 7.2 /etc/ssh/sshd_config *
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing
# and session processing. If this is enabled, PAM authentication
# will be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM no
```

¡IMPORTANTE!

Antes de comprobar que la configuración la hemos cambiado bien, debemos **eliminar nuestra clave** pública del servidor SSH, para ello debemos eliminar manualmente en el archivo del servidor:

nano ~/.ssh/authorized_keys

Aquí buscaremos la línea correspondiente a la que copiamos y comprobaremos intentando acceder antes de reiniciar el servidor SSH (que es cuando se activarán los cambios).

Por último, vamos a comprobar que al acceder no nos pide la contraseña, reiniciamos el servicio SSH en el servidor y una vez inicie, entraremos de nuevo usando:

```
ana@PC219:~$ ssh ana@192.168.2.99
Last login: Thu Oct 31 12:14:45 2024 from 192.168.2.95
ana@ana-VirtualBox:~$
```

2. Github sin contraseña por SSH.

*Nota: aquí genero de nuevo la clave ya que esta parte es desde el ordenador de casa.

Además de usar el comando ya mencionado para la creación de claves, he usado un comando para comprobar que se habían creado.

```
ana@ana-sobremesa:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ana/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ana/.ssh/id_rsa
Your public key has been saved in /home/ana/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:B6pwiBzmRTBhkNJC74UFN7DhafBIMoC71h6J8og8yak ana@ana-sobremesa
The key's randomart image is:
+---[RSA 3072]---+
|X@o=o+
|O.X B .
|.* X . .
|=.=.. . .
|.=+.o . S .
|oo * . .
|*o+ o
|o*..
|E .
+---[SHA256]---+
ana@ana-sobremesa:~$ ls ~/.ssh
id_rsa  id_rsa.pub  known_hosts
```

Una vez creadas las claves vamos a ir a la configuración de nuestra cuenta de Github, al apartado claves SSH y GPG.

The screenshot shows the GitHub Settings page for the user 'Ana Orozco (anaorozco99)'. The left sidebar is visible with various account settings like Public profile, Account, Appearance, Accessibility, Notifications, Access (Billing and plans, Emails, Password and authentication, Sessions), and SSH and GPG keys (selected). The main content area is titled 'SSH keys' and displays a message: 'There are no SSH keys associated with your account.' It includes a link to 'connecting to GitHub using SSH keys' and 'common SSH problems'. Below this is the 'GPG keys' section, which also shows no keys associated and a link to generate a GPG key. At the bottom, there's a 'Vigilant mode' section with an unchecked checkbox for 'Flag unsigned commits as unverified'.

Ahora vamos a darle a añadir nueva clave SSH, tendremos que hacer un **cat** al archivo de la clave pública(id_rsa.pub) y copiaremos el contenido.

The screenshot shows the 'Add new SSH Key' form on the GitHub Settings page. The left sidebar shows the 'SSH and GPG keys' section selected. The main form has a 'Title' field set to 'SSH' and a 'Key type' dropdown set to 'Authentication Key'. The 'Key' field contains the public SSH key content:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQCUV6Ltvzxg6dwEEMzDIHH8xxiXR8ROuKEOkzX/sFghcx5oYUfk+srT5RZoLlbwcEdd2Dyfx3wAFmM02tf+CLs+FebgMlezBw/P72KVMKzYSqzIyzp38Gd+UadBvsPmmPV53Rb+akj3HtDifypbrdWkoINnt7iiw6Hq8fnxtpr9gjOScmrhR/n0ZmXf5WIHGJ41DAtpAxwFwd5G4BBnpUGZl22D6n5jGbC8p46+x0TuUkH2zgDM9PsF0mjJFd1entg6b2h+3eHygxVElhzhA9p0inr6/j7wBKo2MofgGP154yRjeFkQI972BwopfPkBUmeQTuwW42a1c1rlaP3OcTwZ52HnCt12Xigb9cYwxVdqHO/3nxZHmoBUpM/j6VstED49AG0NsdoIfr6G+CZ4nvvcUHCWluImdAkVPUTusScA6rRarVNdhMFRN+i6PrImIrRYejxdtLh9QxKa4Zybli3zWIH8VHI+mrXIwseW8Y26WIdWVj/ncu3JM= ana@ana-sobremera
```

. A blue 'Add SSH key' button is at the bottom.

Lo pegaremos, pondremos el título y como tipo de clave dejaremos clave autenticación.

The screenshot shows the GitHub Settings page for the user 'Ana Orozco (anaorozco99)'. The left sidebar is open, showing options like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (which is selected), Organizations, and Enterprises. The main content area is titled 'SSH keys' and displays a message: 'You have successfully added the key 'SSH''. It lists one SSH key: 'SSH' (SHA256:bPcq...), which was added on Oct 30, 2024, and has never been used (Read/write). A 'Delete' button is visible next to the key entry. Below this, there's a note about connecting to GitHub using SSH keys or troubleshooting common SSH problems. The 'GPG keys' section below shows no keys associated with the account.

Este es el link de mi clave: <https://github.com/anaorozco99.keys>

Ahora vamos a crear un repositorio nuevo en nuestra cuenta de github, yo le he puesto SAD2425 para tenerlo todo más ordenado.

Lo siguiente que vamos a hacer será clonar un repositorio usando SH con el comando: (Necesitamos tener instalado git)

```
git clone git@github.com:anaorozco99/SAD2425.git
```

```
ana@ana-sobremesa:~$ git clone git@github.com:anaorozco99/SAD2425.git
Clonando en 'SAD2425'...
warning: Pareces haber clonado un repositorio sin contenido.
```

Básicamente lo que haremos será crear una carpeta que será nuestro repositorio de Github, por lo tanto todo lo que creemos ahí que sea compatible con Github deberá aparecer en nuestro repositorio.

Vamos a comprobarlo y haremos un archivo de texto y añadiremos algo de contenido.

```
ana@ana-sobremesa:~/SAD2425$ nano prueba.txt
ana@ana-sobremesa:~/SAD2425$ git add prueba.txt
ana@ana-sobremesa:~/SAD2425$ cat prueba.txt
Holaaaaa
```

Y también usaremos el comando **git add** que lo que hará será añadir estos archivos y o cambios a una especie de “cola” que simplemente esperará a que hagamos el próximo **commit** para poder agregarse al repositorio.

En esta captura vemos cómo hago el **commit** y uso el comando **git push origin main**, esto es lo que hará será enviar esos **commit** desde nuestro repositorio local hasta nuestro repositorio de Github, que además en nuestro caso no usaremos usuario ni contraseña porque estamos conectados con clave SSH.

```
ana@ana-sobremesa:~/SAD2425$ git commit -m "Añadido prueba.txt"
[main (commit-raíz) 5a30de8] Añadido prueba.txt
 1 file changed, 1 insertion(+)
   create mode 100644 prueba.txt
ana@ana-sobremesa:~/SAD2425$ git push origin main
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 228 bytes | 228.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:anaorozco99/SAD2425.git
 * [new branch]      main -> main
```

3. Mensaje cifrado.

Aquí te dejo mi clave de Github (aunque también está arriba) y el mensaje cifrado con el programa PGP TOOL.

Para hacer el mensaje cifrado lo que hice fue añadir tu clave pública y crear un texto donde lo cifré y se supone que solo lo puedes leer tú usando tu clave.

Este es el link de mi clave: <https://github.com/anaorozco99.keys>

-----BEGIN PGP MESSAGE-----

Version: BCPG v1.63

```
hQEOAzp0CtqgRRu7EAP/WsLpOHmjIhimU0VaSgX8c4xmvr6YZ3ByXI4BmHrYq/gtrsP1  
E6GaUQCBnRUgjOGvCuWmcHtS5vREefGc3XRL3OXXmDvznDtzlOlFkQQ9k3cs  
PqYXZrwBTralp3684kjP+EbcSycMxRO7B0hOBN7H4OpkNz14pWMhsW3Pq4MzdT0D  
/Ri5WR8yjq7m6d01TO4te2JK4h98e2QgMXsP5j7CMWjLeex/r12aUkqWurDgas+j  
q31+nRXzlaUqKFagvE/scVMvgDN57xNM4730iY/CG4FsvlnGtzUy2O/tDa6clmxF  
0saOWIP1ETQ30Vw8et/vgOHXPXiC9F/J2HMERH1VWvlB0mcBZYkOkJyi8p56liUD  
zpgoukoFGN/+7HjCLGz3xXf1swE3Hemwp4wK0Mf91k7noxMPloZPkLXgciy9Xib8  
6oAFoWU4cn15kwUnN2GQXcS09gJjiE7i1q2eA8Xeexm7w+YM3TFSYqN  
=y778
```

-----END PGP MESSAGE-----