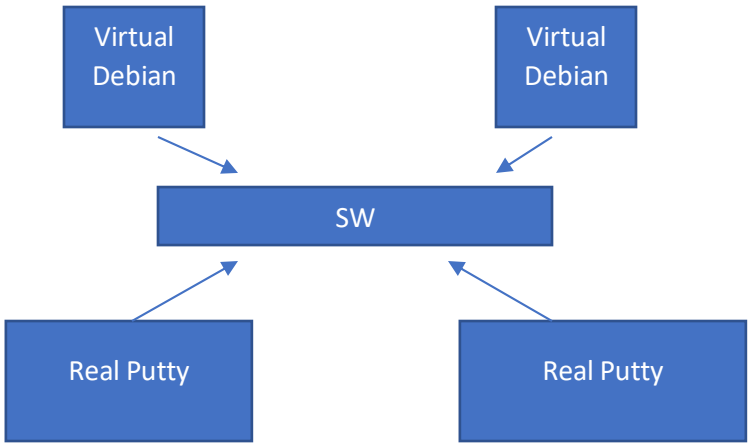
CENTRO DE FORMACIÓN PROFESIONAL RICALDONE	CURSO **Linux / Ciberseguridad**		
GUIA 1 SSH	**Docente:**	David Isaac Ramirez	Fecha:
Integrantes:			

Indicaciones:

Realizar las configuraciones de manera individual, consulte material o al docente.

El trabajo es en parejas, deberá acceder a la maquina de su compañero y validar las configuraciones



Calcular una red para 4 host

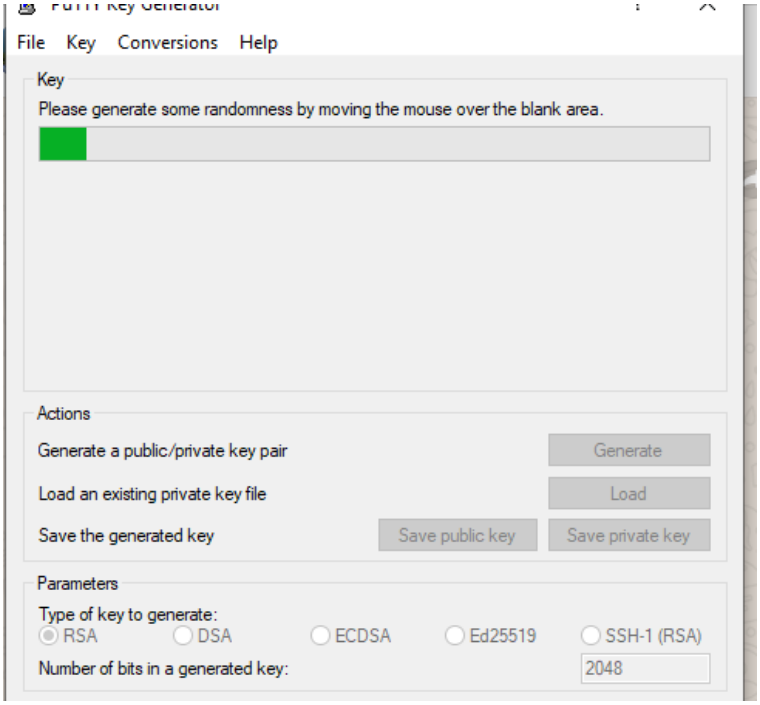
Crear el fichero en la ruta de o carpeta del usuario, en caso de no existir la carpeta .ssh se deberá crear .

Crear el fichero authorized_keys

```
root@debian:/home/dzometa/.ssh# ls
authorized_keys
root@debian:/home/dzometa/.ssh#
```

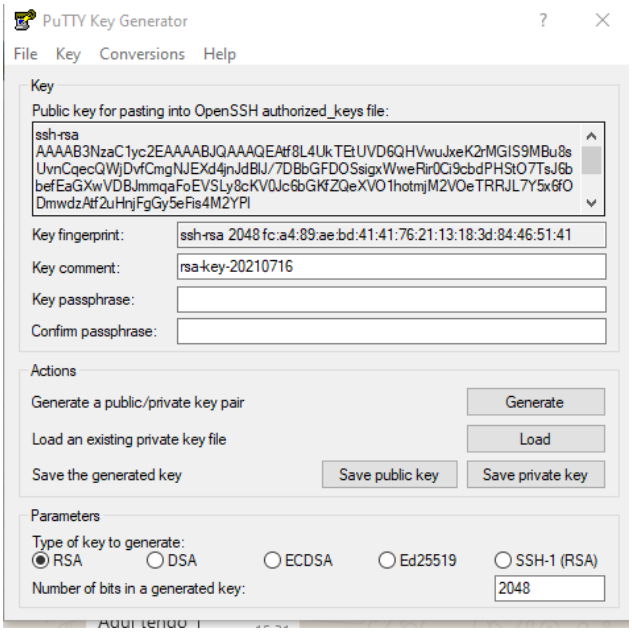
Desde Windows se debe generar la clave privada y publica

Para eso se utiliza el PuttyGen

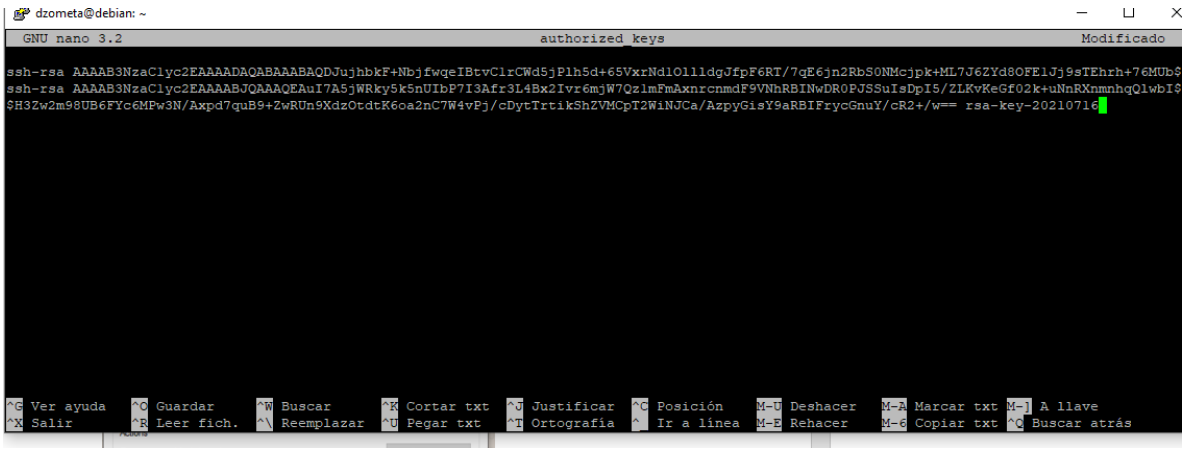


Dar clic en Generar y mover el mouse sobre la barra de progreso hasta que cargue

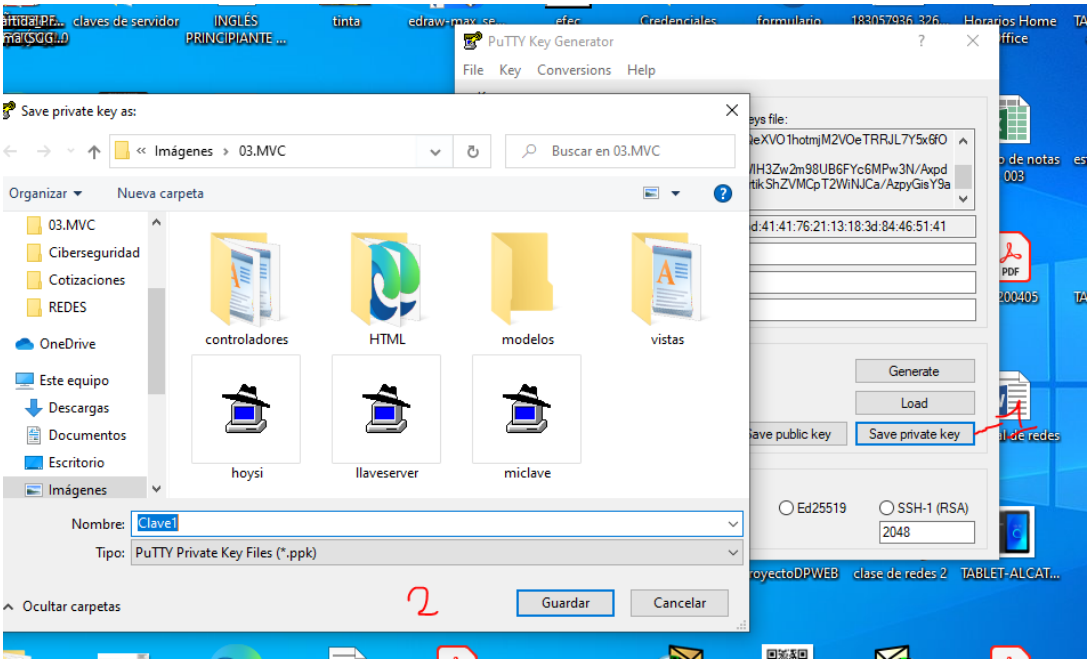
Cuando Cargue la clave vamos a copiar lo que se genero.



El paso siguiente, vamos a conectarnos con putty para poder pegar esa clave en el archivo authorized_keys que generamos anteriormente.



La llave que copiamos es nuestra llave publica, falta guardar en el putty nuestra llave privada.



Con esto ya tenemos configurado nuestras claves

CONFIGURANDO PARAMETROS DE SEGURIDAD

Podemos aplicar algunas configuraciones para poder restringir algunos accesos.

1. Cambiar de puerto predeterminado
2. Limitar que no inicien con el usuario Root
3. Limitar que los usuarios de sistema no se autentiquen con sus credenciales
4. Limitar a usuarios específicos que inicien

Ruta de fichero de configuración

```
dzometa@debian: ~
root@debian:/home/dzometa# cd .ssh/
root@debian:/home/dzometa/.ssh# nano authorized_keys
root@debian:/home/dzometa/.ssh# nano /etc/ssh/sshd_config
```

Los cambios realizados serian los siguientes

```
Port 2225
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 6
MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
AllowUsers dzometa
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
```

```
ClientAliveInterval 0
ClientAliveCountMax 3
UseDNS no
PidFile /var/run/sshd.pid
MaxStartups 10:30:100
PermitTunnel no
ChrootDirectory none
VersionAddendum none

no default banner path
Banner none

Allow client to pass locale environment variables
AcceptEnv LANG LC_*

override default of no subsystems
Subsystem       sftp    /usr/lib/openssh/sftp-server

Example of overriding settings on a per-user basis
Match User anoncvs
        X11Forwarding no
        AllowTcpForwarding no
        PermitTTY no
        ForceCommand cvs server
```

PermitRootLogin no

De esta manera las conexiones root quedarán bloqueadas evitando que usuarios no autorizados puedan realizar ataques de fuerza bruta contra nuestro servidor SSH para adivinar las credenciales del usuario Root. También tenemos otras opciones en este apartado, como por ejemplo «PermitRootLogin without-password» donde se permite autenticación, pero no con usuario y contraseña, sino con claves criptográficas RSA.

Configuraciones de seguridad adicionales

Existen otras configuraciones recomendadas para evitar las conexiones no deseadas a nuestro servidor SSH. Estas conexiones son:

LoginGraceTime: Estableceremos el tiempo necesario para introducir la contraseña, evitando que el atacante tenga que «pensar mucho».

MaxAuthTries: Número de intentos permitidos al introducir la contraseña antes de desconectarnos.

MaxStartups: Número de logins simultáneos desde una IP, para evitar que se pueda utilizar la fuerza bruta con varias sesiones a la vez.

AllowUsers: Es crear una lista blanca de usuario. Este parámetro nos permite configurar los usuarios que podrán conectarse. Una medida muy restrictiva, pero a la vez muy segura ya que bloqueará todas las conexiones de los usuarios que no estén en el listado. Los usuarios que tengamos aquí podrán conectarse, y el resto no.

DenyUsers: Parecido al anterior, pero ahora creamos una lista negra. Los usuarios que tengamos aquí no podrán conectarse, y el resto sí.

Tras terminar las configuraciones vamos a guardar los cambios