

INSTITUTO TECNOLÓGICO DE CULIACÁN



INGENIERIA EN SISTEMAS COMPUTACIONALES

ADMINISTRACION DE REDES

IMPLEMENTACION DE SERVICIOS SSH

ALUMNO:

ROSALES CORVERA HERNAN ENRIQUE

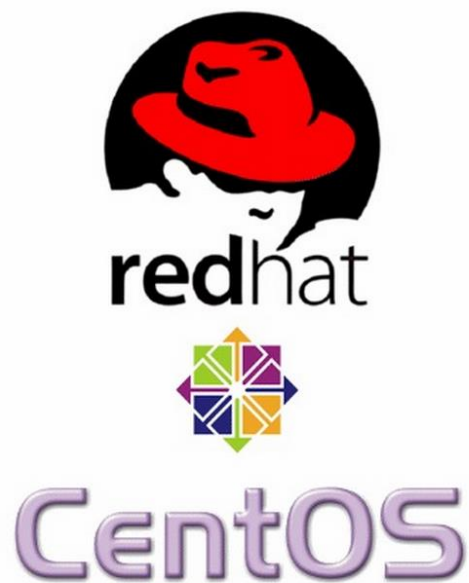
DOCENTE: LUIS ERNESTO LIZARRAGA BOLAÑOS

CULIACAN SINALOA, 30 DE JUNIO DEL 2019

SISTEMA OPERATIVO CENTOS

CentOS es un proyecto de código abierto gratuito de nivel empresarial con la misma funcionalidad, rendimiento y estabilidad que el sistema operativo de pago Redhat Enterprise Linux (RHEL). CentOS comparte casi el 95% de las características de la RHEL comercial con la gran diferencia de la falta de puerto IBM System z y algunas variantes limitadas para la virtualización.

CentOS Linux está desarrollado por un pequeño pero creciente equipo de desarrolladores centrales. A su vez, los desarrolladores principales cuentan con el respaldo de una comunidad activa de usuarios que incluye administradores de sistemas, administradores de redes, administradores, contribuyentes principales de Linux y entusiastas de Linux de todo el mundo.



SSH

SSH o Secure Shell, es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de Internet. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada. Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente.

El protocolo SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremadamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de reenviar aplicaciones X11 desde el servidor. Esta técnica, llamada reenvío por X11, proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

Ya que el protocolo SSH encripta todo lo que envía y recibe, se puede usar para asegurar protocolos inseguros. El servidor SSH puede convertirse en un conducto para convertir en seguros los protocolos inseguros mediante el uso de una técnica llamada reenvío por puerto, como por ejemplo POP, incrementando la seguridad del sistema en general y de los datos.

Red Hat Enterprise Linux contiene el paquete general de OpenSSH (openssh) así como también los paquetes del servidor OpenSSH (openssh-server) y del cliente (openssh-clients). Consulte el capítulo titulado OpenSSH en el Manual de administración del sistema de Red Hat Enterprise Linux para obtener instrucciones

sobre la instalación y el desarrollo de OpenSSH. Observe que los paquetes OpenSSH requieren el paquete OpenSSL (openssl). OpenSSL instala varias bibliotecas criptográficas importantes, permitiendo que OpenSSH pueda proporcionar comunicaciones encriptadas.

Historia

Al principio sólo existían los r-commands, que eran los basados en el programa rlogin, el cual funciona de una forma similar a telnet.

La primera versión del protocolo y el programa eran libres y los creó un finlandés llamado Tatu Ylönen, pero su licencia fue cambiando y terminó apareciendo la compañía SSH Communications Security, que lo ofrecía gratuitamente para uso doméstico y académico, pero exigía el pago a otras empresas. En el año 1997 (dos años después de que se creara la primera versión) se propuso como borrador en la IETF.

A principios de 1999 se empezó a escribir una versión que se convertiría en la implementación libre por excelencia, la de OpenBSD, llamada OpenSSH.

Los usuarios nefarios tienen a su disposición una variedad de herramientas que les permiten interceptar y redirigir el tráfico de la red para ganar acceso al sistema. En términos generales, estas amenazas se pueden catalogar del siguiente modo:

- Intercepción de la comunicación entre dos sistemas — En este escenario, existe un tercero en algún lugar de la red entre entidades en comunicación que hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada.

Este ataque se puede montar a través del uso de un paquete sniffer — una utilidad de red muy común.

- Personificación de un determinado host — Con esta estrategia, un sistema interceptor finge ser el recipiente a quien está destinado un mensaje. Si funciona la estrategia, el sistema del usuario no se da cuenta del engaño y continúa la comunicación con el host incorrecto.

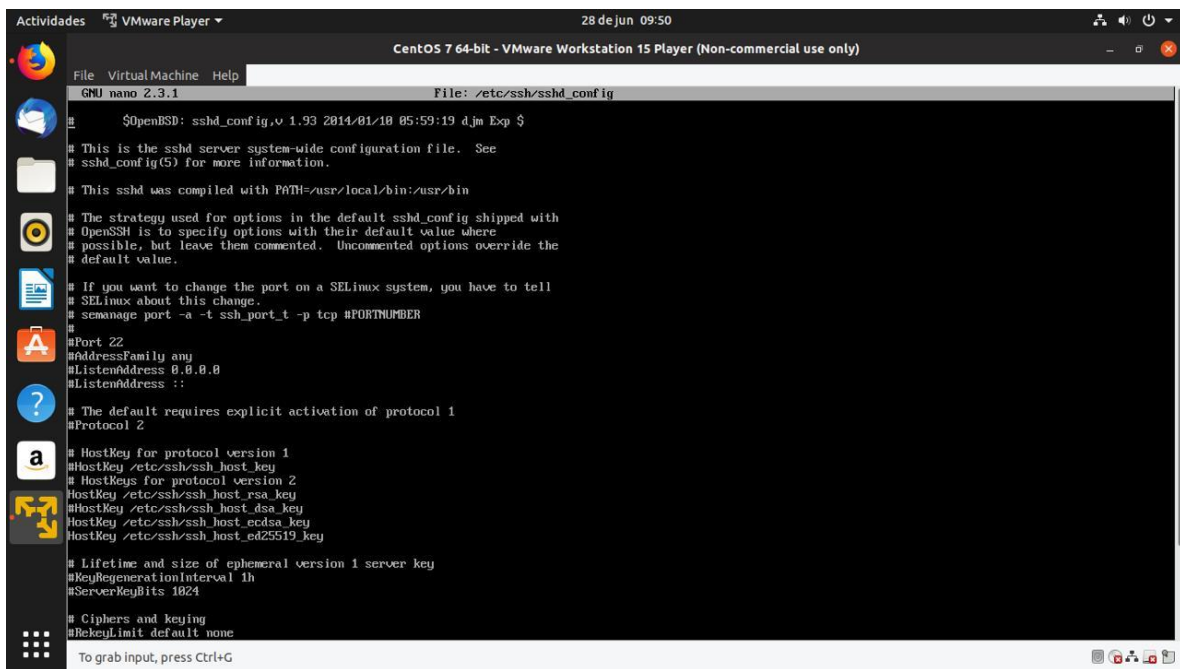
Esto se produce con técnicas como el envenenamiento del DNS o spoofing de IP (engaño de direcciones IP).

Ambas técnicas interceptan información potencialmente confidencial y si esta interceptación se realiza con propósitos hostiles, el resultado puede ser catastrófico.

Si se utiliza SSH para inicios de sesión de shell remota y para copiar archivos, se pueden disminuir estas amenazas a la seguridad notablemente. Esto es porque el cliente SSH y el servidor usan firmas digitales para verificar su identidad. Adicionalmente, toda la comunicación entre los sistemas cliente y servidor es encriptada. No servirán de nada los intentos de falsificar la identidad de cualquiera de los dos lados de la comunicación ya que cada paquete está cifrado por medio de una llave conocida sólo por el sistema local y el remoto.

IMPLEMENTACION DE SERVICIOS SSH

- 1.- crear usuario: redes password: redes (ESTE PASO SE REALIZO PREVIAMENTE EN LA PRACTICA 2.2)
- 2.- deshabilitar firewall y selinux: (ESTE PASO SE REALIZO PREVIAMENTE EN LA PRACTICA 2.2.).
- 3.- Configure SSH Server to login to a server from remote computer. (Realizar sección ssh puntos 1, 3 y 4.)



```
File Virtual Machine Help
CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)
GNU nano 2.3.1 File: /etc/ssh/sshd_config
#
#OpenBSD: sshd_config,v 1.93 2014/01/10 05:59:19 djm Exp $
#
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
#
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin
#
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
#
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#
# The default requires explicit activation of protocol 1
#Protocol 2
#
# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
#
# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 1024
#
# Ciphers and keying
#RekeyLimit default none
#
To grab input, press Ctrl+G
```

Actividades VMware Player 28 de jun 09:53
CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)

```
File Virtual Machine Help
#HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 1024

# Ciphers and keying
#RekeyLimit default none

# Logging
# obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 3
#MaxSessions 3

#RSAAuthentication yes
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

[ Wrote 153 lines ]

[root@localhost ~]#
```

To grab input, press Ctrl+G

Actividades VMware Player 28 de jun 09:55
CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)

```
File Virtual Machine Help
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 1024

# Ciphers and keying
#RekeyLimit default none

# Logging
# obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 3
#MaxSessions 3

#RSAAuthentication yes
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

[ Wrote 153 lines ]

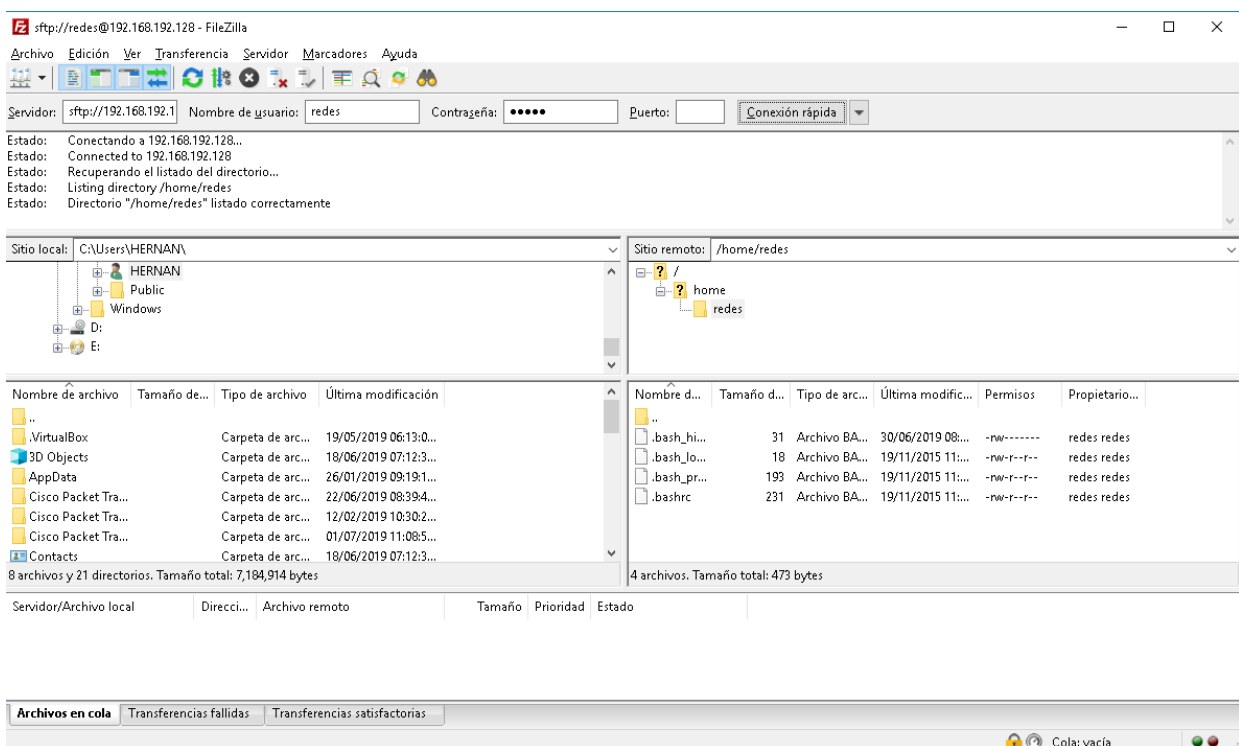
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]#
```

To grab input, press Ctrl+G


```
CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)
root@localhost~#
root@LabRed09:/home/network# ssh root@192.168.41.216
The authenticity of host '192.168.41.216 (192.168.41.216)' can't be established.
ECDSA key fingerprint is SHA256:B4V4CS566/PH9HTIAl+Ce3T9SWqYEk5MwRdNtk2j4lo.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.41.216' (ECDSA) to the list of known hosts.
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Received disconnect from 192.168.41.216 port 22:2: Too many authentication failu
res for root
Disconnected from 192.168.41.216 port 22
root@LabRed09:/home/network# ssh root@192.168.41.216
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Received disconnect from 192.168.41.216 port 22:2: Too many authentication failu
res for root
Disconnected from 192.168.41.216 port 22
root@LabRed09:/home/network# ssh redes@192.168.41.216
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:ba:f7:ee brd ff:ff:ff:ff:ff:ff
    inet 192.168.41.216/23 brd 192.168.41.255 scope global dynamic eno16777736
        valid_lft 6179sec preferred_lft 6179sec
    inet6 fe80::20c:29ff:feba:f7ee:64 scope link
        valid_lft forever preferred_lft forever
root@localhost ~#
```

```
CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)
root@localhost~#
root@LabRed09:/home/network# ssh root@192.168.41.216
Received disconnect from 192.168.41.216 port 22:2: Too many authentication failu
res for root
Disconnected from 192.168.41.216 port 22
root@LabRed09:/home/network# ssh root@192.168.41.216
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Permission denied, please try again.
root@192.168.41.216's password:
Received disconnect from 192.168.41.216 port 22:2: Too many authentication failu
res for root
Disconnected from 192.168.41.216 port 22
root@LabRed09:/home/network# ssh redes@192.168.41.216
Auth: redes@192.168.41.216's password:
Last login: Fri Jun 28 09:20:17 2019
[redes@localhost ~]$ su -
Contraseña:
su: Fallo de autenticación
[redes@localhost ~]$ su -
Contraseña:
Último inicio de sesión: vie jun 28 09:39:55 MDT 2019 en tty1
[redes@localhost ~]$ su -
Último inicio de sesión fallido: vie jun 28 10:00:00 MDT 2019 en pts/0
[redes@localhost ~]$ su -
Hubo 7 intentos de logueo fallidos desde el último logueo exitoso.
[redes@localhost ~]$
1: lo: loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
   link/ether 00:0c:29:ba:f7:ee brd ff:ff:ff:ff:ff:ff
   inet 192.168.41.216/23 brd 192.168.41.255 scope global dynamic eno16777736
       valid_lft 6179sec preferred_lft 6179sec
   inet6 fe80::20c:29ff:feba:f7ee:64 scope link
       valid_lft forever preferred_lft forever
[redes@localhost ~]$
```

Una vez que accedimos mediante la terminal, procederemos al paso 3. Procederemos a instalar Filezilla para poder compartir archivos mediante la máquina virtual y la maquina real. Lo descargaremos desde la página oficial y lo instalaremos como cualquier otro programa. Una vez instalado procederemos a poner la dirección ip del servidor, el nombre de usuario, contraseña y el puerto por el cual accederemos, en este caso el 22. Una vez dentro seremos capaces de hacer cualquier compartición de archivos requerida.

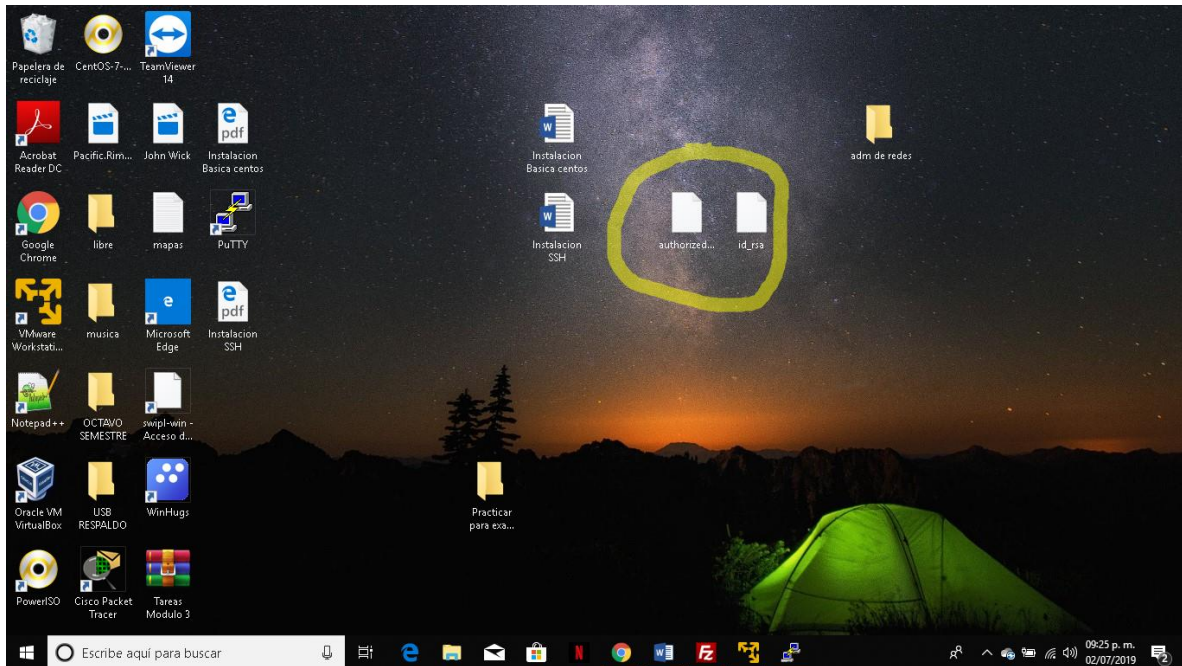


Paso 4- crear una llave de usuario para acceder mediante ella.

Generamos la llave mediante el comando `ssh-keygen -t rsa`

```
[redes@localhost ~]$  
[redes@localhost ~]$  
[redes@localhost ~]$  
[redes@localhost ~]$  
[redes@localhost ~]$  
[redes@localhost ~]$ ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/redes/.ssh/id_rsa):  
Created directory '/home/redes/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/redes/.ssh/id_rsa.  
Your public key has been saved in /home/redes/.ssh/id_rsa.pub.  
The key fingerprint is:  
2f:9f:1e:07:6f:3c:52:c1:84:c9:df:60:b6:2d:e0:90 redes@localhost.localdomain  
The key's randomart image is:  
+--[ RSA 2048 ]-----+  
|      o o.      |  
|    E =o+      |  
|      o =o=     |  
|      . +.o     |  
|    S . . .     |  
|      . =       |  
|    . + *       |  
|      o * .     |  
|      .+        |  
+-----+  
[redes@localhost ~]$
```

Procederemos a transferir la llave creada en el server al cliente. Puedes transferirlos mediante el filezilla o mediante la línea de comandos, en mi caso no me funcionaron los comandos así que los traspase manualmente con el filezilla y los copie al escritorio.



El siguiente paso es ingresar mediante al servidor, tomando como dirección la ubicación donde está el archivo key, al iniciar de ese modo nos pedirá la palabra clave ingresada y al ponerla correctamente, habremos ingresado mediante la llave autorizada.

```
Authenticating with public key "imported-openssh-key"  
Passphrase for key "imported-openssh-key":
```