

IMPLEMENTACIÓN DE SERVICIOS SSH

Historia SSH

El protocolo SSH fue desarrollado en el año 1995 por el finlandés TatuYlönen, quien publicó su trabajo bajo una licencia de libre uso, pero ante el éxito del programa desarrollado pronto registró la marca SSH y fundó la empresa SSH. Communications Security con fines comerciales, la cual permitía el uso del protocolo gratuitamente para uso doméstico y educativo.

Ante este cambio en la política de uso del protocolo, los desarrolladores del sistema operativo OpenBSD empezaron a desarrollar en el año 1999 una versión libre de este protocolo que recibió el nombre de OpenSSH.

Desde la aparición de este protocolo han sido dos las versiones que han estado activas. En la primera de ellas, se ofrecía una alternativa a las sesiones interactivas mediante el uso de herramientas como TELNET, RSH o RLOGIN entre otras, sin embargo, pronto se descubrió que este protocolo tenía un punto débil que permitía a los hackers introducir datos en los flujos cifrados.

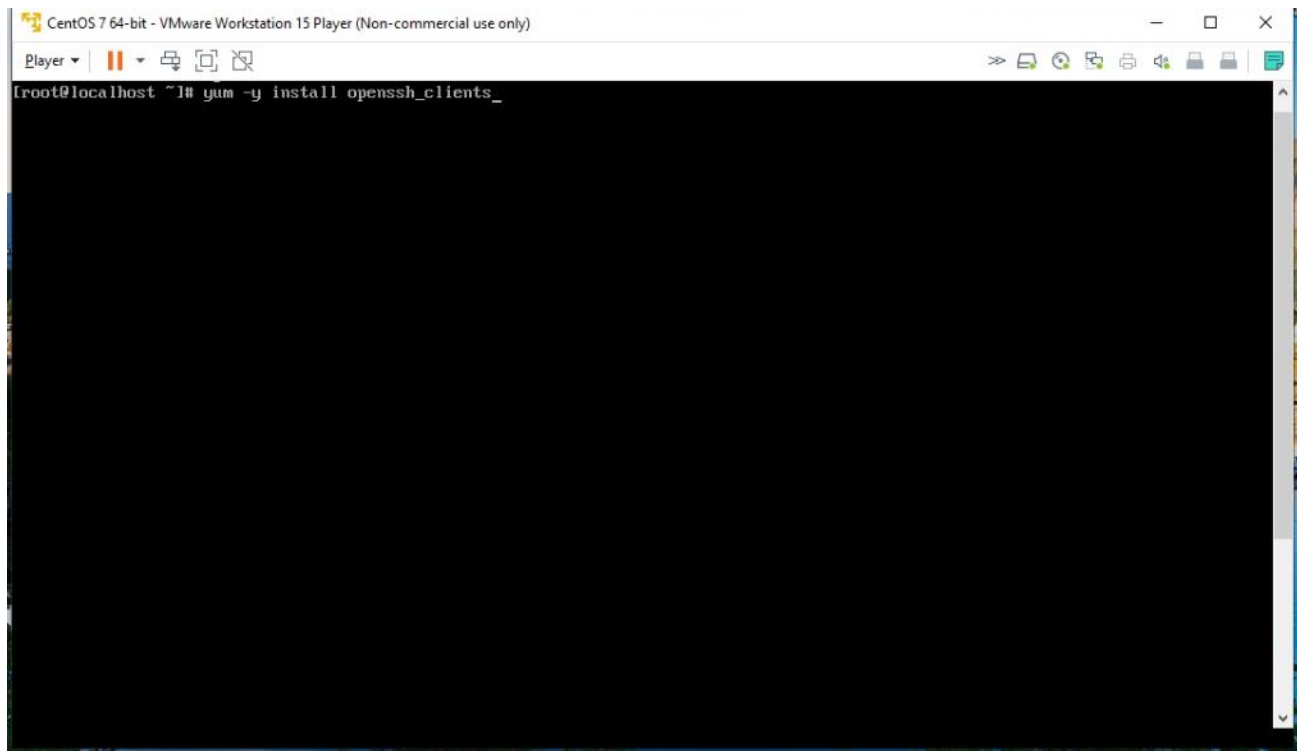
Ante este problema, en el año 1997 fue lanzada la versión 2, donde una serie de medidas solucionaban el problema descubierto en la primera versión. Además de corregir ese problema, esta segunda versión incorporaba el protocolo SFTP (Secure File Transfer Protocol – Protocolo seguro de transferencia de archivos) que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos de forma segura.

Funcionalidad

El funcionamiento de este protocolo se puede resumir en los siguientes pasos que os dejamos a continuación:

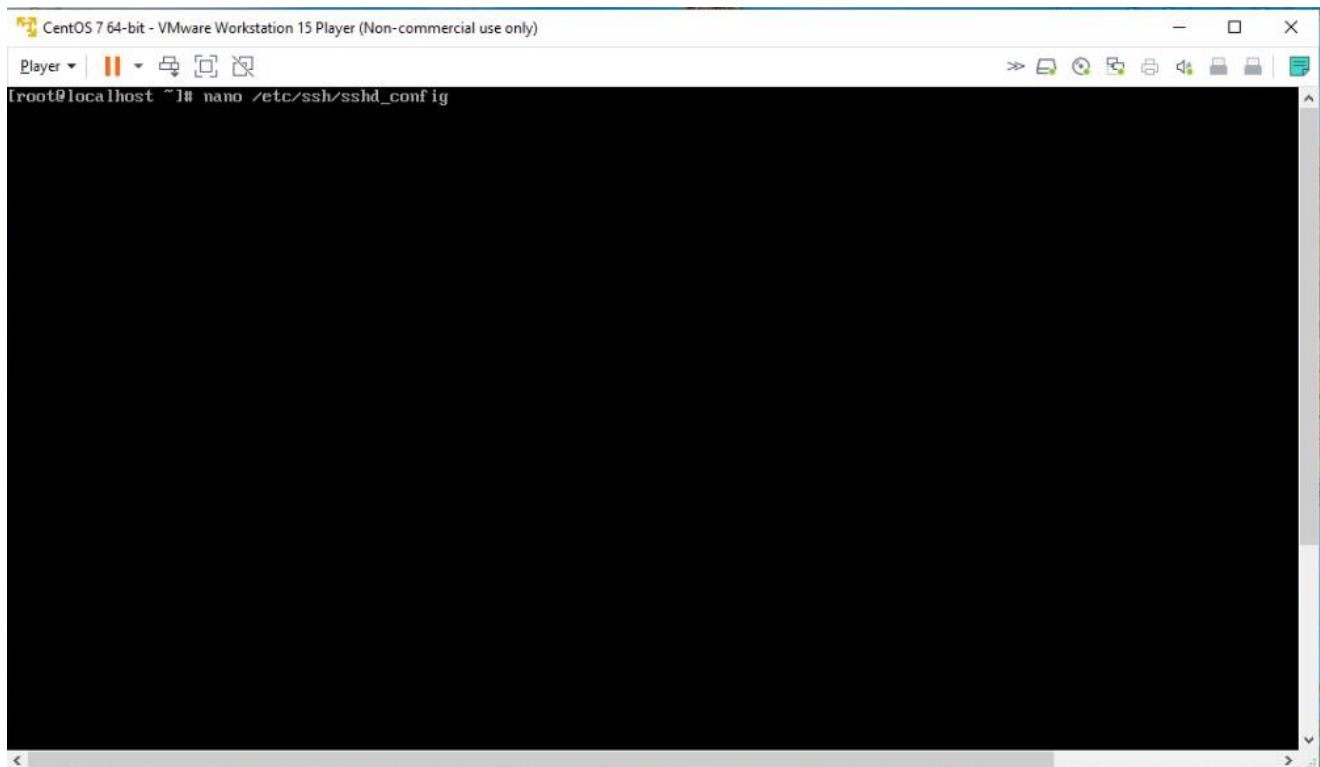
1. El cliente inicia una conexión TCP sobre el puerto 22 del servicio. Este puerto es el que utiliza por defecto
2. El cliente y el servidor se ponen de acuerdo en la versión del protocolo a utilizar, así como el algoritmo de cifrado utilizado para el intercambio de la información.
3. El servidor, que tiene en su poder dos claves (una privada y una pública), manda su clave pública al cliente.
4. Cuando el cliente recibe la clave enviada por el servidor, la compara con la que tiene almacenada para verificar su autenticidad. El protocolo SSH exige que el cliente la confirme la primera vez.
5. Con la clave pública del servidor en su poder, el cliente genera una clave de sesión aleatoria, creando un mensaje que contiene esa clave y el algoritmo seleccionado para la encriptación de la información. Toda esa información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada.
6. Si todo es correcto, el cliente queda autenticado, iniciando la sesión para comunicarse con el servido

Sección SSH



A terminal window titled "CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)". The prompt is [root@localhost ~]#. The command entered is yum -y install openssh-clients_. The terminal area is mostly black, indicating the command is running or has completed.

```
[root@localhost ~]# yum -y install openssh-clients_
```



A terminal window titled "CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)". The prompt is [root@localhost ~]#. The command entered is nano /etc/ssh/sshd_config. The terminal area is mostly black, indicating the nano editor is open.

```
[root@localhost ~]# nano /etc/ssh/sshd_config
```

CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)

Player ▾ | [Icons] | File: /etc/ssh/sshd_config

```
GNU nano 2.3.1

# Authentication:
#LoginGraceTime 2m
#PermitRootLogin no_
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#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

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and 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!
```

CentOS 7 64-bit - VMware Workstation 15 Player (Non-commercial use only)

Player ▾ | [Icons]

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