

**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

**Soto Ramos Manuel Alejandro**

**Administración de Servicios en Red**

**Manual de Instalación y Configuración - Servidor SSH**

Jiménez Maruri Pedro

Valencia Rodriguez Fernando  
Romero Serrano Luis

Grupo: 4CV4

# Índice

[**Índice**](#_aluvl3g2tzy) **2**

[**Requerimientos**](#_xoopottn43hu) **3**

[Hardware](#_hpriteko4dw2) 3

[Software](#_35w07g5g83gt) 3

[**Introducción**](#_7fhowhjndg9m) **3**

[¿Para qué sirve?](#_dav327vi6lt) 3

[¿Cómo funciona?](#_bl6wbbr5jq4i) 3

[Tipos de Encriptación SSH](#_brei74pfm9r) 4

[**Sistema**](#_gcfx02lakr1y) **4**

[**Licencia**](#_22cr43a4njz4) **4**

[**Instalación**](#_7rb9tqmwoj6u) **5**

[**Configuración**](#_dd1p4pv9ub2h) **5**

[Restringir los accesos al servidor con los siguientes criterios:](#_vwh94vo7rea) 6

[Denegación por segmento de red definido:](#_4qaxcbddmqzp) 6

[Acceso por usuario (dirección IP):](#_y7s6tlc1mbif) 7

[Lista de usuarios:](#_o6vg4c1dqc7y) 7

[Configuraciones de seguridad adicionales](#_t2auo7hzp958) 7

[Personalización de los niveles y formatos de bitácoras del servidor.](#_1gwconxmoiq1) 7

[Personalización de tiempo máximo de conexión](#_brb9lmtd1ypk) 8

[Personalización de usuarios recurrentes al sistema](#_ur5ue4d3b9e2) 8

[**Bitacoras**](#_m7k7oo0nv2k) **9**

[**Referencias**](#_aluvl3g2tzy) **10**

# 

# Requerimientos

## Hardware

* 2 CPU cores
* 4 GB RAM (o más)
* 100G fast disk (SAS 15K) (RAID-0 or RAID-10) además del disco OS.
* Solución de copia de seguridad

## Software

* PuTTY
* Apache Tomcat
* MySQL
* FileZilla

# Introducción

**SSH** es el acrónimo de **Secure Shell**, y es un protocolo que se utiliza en el manejo de servidores de forma remota, permitiendo a un usuario realizar toda clase de tareas o actividades sobre él. En las conexiones realizadas por medio de SSH,toda la información viaja de forma encriptada, lo cual lo convierte en uno de los medios más seguros a la hora de trabajar en un servidor. [1]

## ¿Para qué sirve?

SSH es un protocolo de conexión remota segura y encriptada para que cualquier usuario pueda transferir información. Pero no solamente se pueden transferir datos, también es una útil forma de ingresar al servidor para conectarnos a su intérprete de comandos, y así poder administrar el servidor remotamente. [1]

## ¿Cómo funciona?

La conexión SSH usa tres ítems:

* Un usuario.
* Un puerto.
* Un servidor.

Con solo esos tres elementos podemos establecer una conexión segura entre dos servidores. Esta seguridad se logra mediante el uso de llaves y técnicas de cifrado.   
Cada Servidor tiene su propia llave de cifrado, y al establecer una conexión por primera vez con un server tendremos que añadir el servidor en cuestión a una lista de servidores a los cuales es seguro conectarnos. [1]

## Tipos de Encriptación SSH

La seguridad que brinda SSH se logra por medio a las técnicas de cifrado, y hay tres de ellas:

* **Encriptación simétrica**: Se basa en el uso de una clave secreta. Esta clave se usa tanto en el cifrado como en el descifrado de la conexión. Por supuesto si alguien logra interceptar la conexión y de alguna forma obtuvo la clave secreta entonces podrá ver claramente la información que se está transfiriendo entre los servidores.
* **Encriptación asimétrica**: Se trata de una clave pública y de una clave privada. Cada clave pública está ligada a su propia clave privada, y la información encriptada sólo se puede desencriptar conociendo la clave privada, así que incluso si tenemos la clave pública no vamos a poder ver los datos, para eso sí o sí se necesita la privada, la cual por supuesto no es compartida con terceros y no puede ser calculada a partir de la pública.
* **Hashing**: Las conexiones cifradas mediante hash se logran convirtiendo la información en una nueva cadena de datos que poseen una cierta longitud que jamás cambia. El hash que se originó en un servidor tiene que ser idéntico al que es recibido por el otro servidor, si hubo alteraciones en el hash recibido quiere decir que la información de alguna forma fue interceptada y modificada. [1]

# Sistema

Como OpenSSH es una aplicación de comunicación entre ordenadores, debe funcionar en la mayor cantidad de sistemas operativos posibles, para lo que existe un equipo, llamado OpenSSH Portability Team que se responsabiliza de añadir el código necesario para portar el software a todas las plataformas posibles. [2]

# Licencia

Este servidor cuenta con licencia *Berkeley Software Distribution* (BSD) permite el uso del código fuente en software no libre. [3]

# Instalación

OpenSSH está disponible en los repositorios de software por defecto de Ubuntu, por lo que lo instalaremos utilizando las herramientas convencionales de gestión de paquetes.

Comenzaremos por actualizar el índice de paquetes locales para que refleje los últimos cambios. Después, podemos instalar el paquete OpenSSh:

sudo apt-get update

Por lado del servidor:

sudo apt-get install openssh-server

Por lado del cliente:

sudo apt-get install openssh-client

# Configuración

Los archivos de configuración son los siguientes:

* **/etc/ssh/ssh\_config:** Archivo de configuración del cliente ssh
* **/etc/ssh/sshd\_config:** Archivo de configuración del servidor ssh

Dentro del archivo /etc/ssh/sshd\_conf se realizarán los siguientes cambios de configuración. Para poder acceder a él y hacer las modificaciones pertinentes ocupamos el siguiente comando.

sudo nano /etc/ssh/sshd\_conf

Permitir acceso por usuarios del sistema menos al usuario root. se define la siguiente línea para no permitir al usuario root el acceso mediante ssh.

Accedemos a él y declaramos la siguiente línea:

PermitRooLogin no

Limitar el número de intentos de conexión por tiempo y contraseña fallidas. La línea siguiente tiene un argumento en segundos, que indican el tiempo permitido para intentar acceder remotamente, lo cual si es rebasado este tiempo, el intento se cancela.

LoginGraceTime 60

En esta línea se define la cantidad de intentos de acceso

remoto.

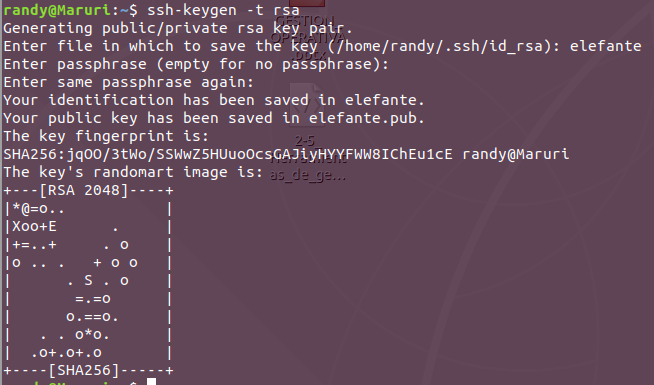
MaxAuthTries 5

Conexión segura utilizando llaves de identificación entre al menos 4 equipos (no requieren password para autenticarse.

Creación de llaves en el cliente mediante el comando.

ssh-keygen -t rsa

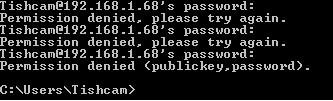
Solicita al cliente definir una palabra para cifrar dicha llave y se muestra a continuación en la siguiente imagen.



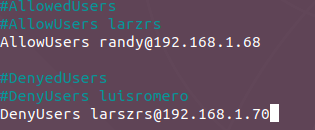
## Restringir los accesos al servidor con los siguientes criterios:

### Denegación por segmento de red definido:





### Acceso por usuario (dirección IP):



### Lista de usuarios:

A partir de los usuarios creados en el servidor, se anexan algunos a un grupo, dicho grupo no tendrá acceso remoto.



### 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.
* **AllowGroups/DenyUsers**: Exactamente igual a lo anterior, pero en lugar de crear una lista blanca/negra de usuarios, es de grupos de usuarios.

### Personalización de los niveles y formatos de bitácoras del servidor.

La siguiente línea muestra el nivel de la bitácora del servidor, donde el argumento

VERBOSE es el nivel más alto de registros a mostrar por parte de SSH.

LogLevel VERBOSE

### Personalización de tiempo máximo de conexión

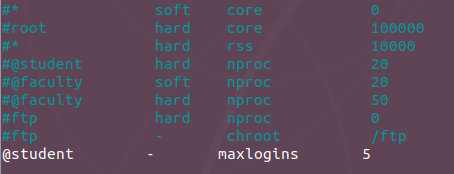


Lo que quiere decir esta configuración es que en las sesiones inactivas el servidor comprobará que la conexión sigue activa cada 60 segundos y si es así no cerrará la sesión. En el caso de que realmente se interrumpa la comunicación entre el servidor y el cliente, el servidor la dará por "muerta" a los 10 intentos fallidos, es decir, en 10 minutos (10x60).

### Personalización de usuarios recurrentes al sistema

Para esta parte hay que cambiar la configuración de security, que se encuentra en:

cd /etc/security/limits.conf



Para más información sobre los valores y sus significados, puede consultar la pagina https://linux.die.net/man/5/limits.conf y ver los ejemplos, así como los cambios que se pueden realizar.

# 

# 

# 

# Bitacoras

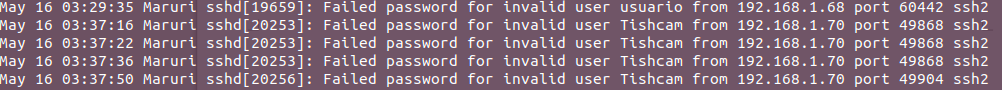
Cómo configuramos este apartado con VERBOSE nos entregará todos los registros que se verán reflejados en el archivo “auth.log” que se encuentran en:

cd /var/log

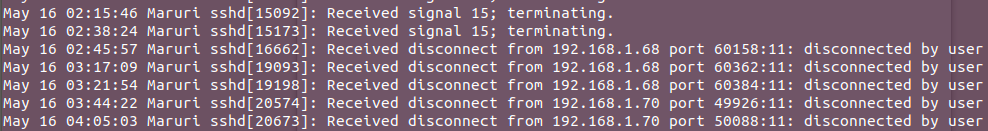
Para agrupar solo la parte del ssh de los log:

grep sshd.\\*Failed /var/log/auth.log | less

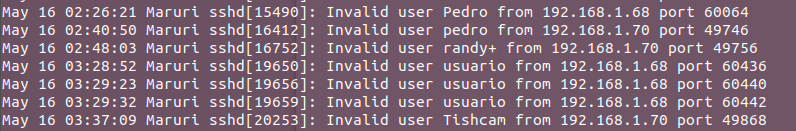
Contraseñas no válidas



Peticiones



Usuarios denegados



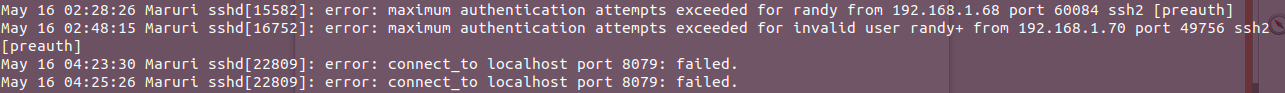
Contraseñas aceptadas



Desconectar usuarios



Errores



# Referencias

[1]. Redeszone, 13/05/2019, https://www.redeszone.net/seguridad-informatica/servidor-ssh-en-linux-manual-de-configuracion-para-maxima-seguridad/

[2]. Linuxconfig, 13/05/2019, https://linuxconfig.org/enable-ssh-on-ubuntu-18-04-bionic-beaver-linux

[3]. Ubuntu, 13/05/2019, https://help.ubuntu.com/lts/serverguide/openssh-server.html.en

[4]. Putty, 13/05/2019, https://www.putty.org/

[5]. Serverfault, 13/05/2019, https://serverfault.com/questions/617081/how-to-use-both-allowgroups-and-allowusers-in-sshd-config

[6]. Latascadexela, 13/05/2019, http://www.latascadexela.es/2011/11/evitar-timeout-de-ssh-en-centos-red-hat.html

[7]. Poftut, 13/05/2019, https://www.poftut.com/limits-conf-file-limit-users-process-linux-examples/

[8]. Solvetic, 13/05/2019, https://www.solvetic.com/topic/1248-%C2%BFcomo-puedo-limitar-los-usuarios-concurrentes-conectados-en-linux/

[9]. Linux, 13/05/2019, https://linux.die.net/man/5/limits.conf