

2019

MONITORIZACIÓN DE SERVICIOS DE RED

JUAN JOSE CANO FERNÁNDEZ
2ºASIR

Contenido

1. Para que escuche por un puerto distinto.....	2
2. Crea una lista blanca de usuarios permitidos.	2
3. Permitir acceso a root solo con clave instalada en el servidor.	4
4. Activar el registro de logs para el servicio sshd.	4
5. Probar conexión varias veces, con fallo y de manera exitosa.....	5
6. Mostrar los últimos 10 mensajes generados por el servicio sshd y, exportarlos en formato json.....	6
7. Configurar journalctl para hacer persistentes los registros, pero que no ocupen más de 100MB de espacio en disco.....	7
8. Mostrar los registros autorizados de acceso remoto de la últimos 5 usuarios.	7
9. Revisar los intentos con error de acceso generados por el servicio sshd en las últimas 24 horas.....	8

1. Para que escuche por un puerto distinto.

Por defecto los servidores SSH utilizan el puerto 22 para las conexiones. Es recomendable cambiar este número de puerto, para evitar que bots o cibercriminales puedan intentar iniciar sesión, aunque por sí solo esto no proporciona seguridad, sí podremos pasar desapercibidos a los escaneos masivos desde Internet. Si por ejemplo queremos usar el puerto 22445 debemos poner en el fichero de configuración lo siguiente:

```
GNU nano 2.7.4 Fichero: /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
# 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.
Port 22445_
#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 prohibit-password
[ 123 líneas escritas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^E Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

2. Crea una lista blanca de usuarios permitidos.

Por un lado, podemos crear una lista blanca por usuarios o por ips, vamos a crearla primero por ips y después por usuarios:

Primero definimos el valor predeterminado de las políticas para borrar todo lo que está:

iptables -P INPUT DROP

iptables -P OUTPUT DROP

Segundo creamos una nueva cadena:

```
iptables -N allowed_ips
```

Tercero si los usuarios que se conecten tienen la ip entre este rango permitimos el acceso.

```
iptables -A allowed_ips -m iprange --src-range 30.1.168.192-50 j ACCEPT
```

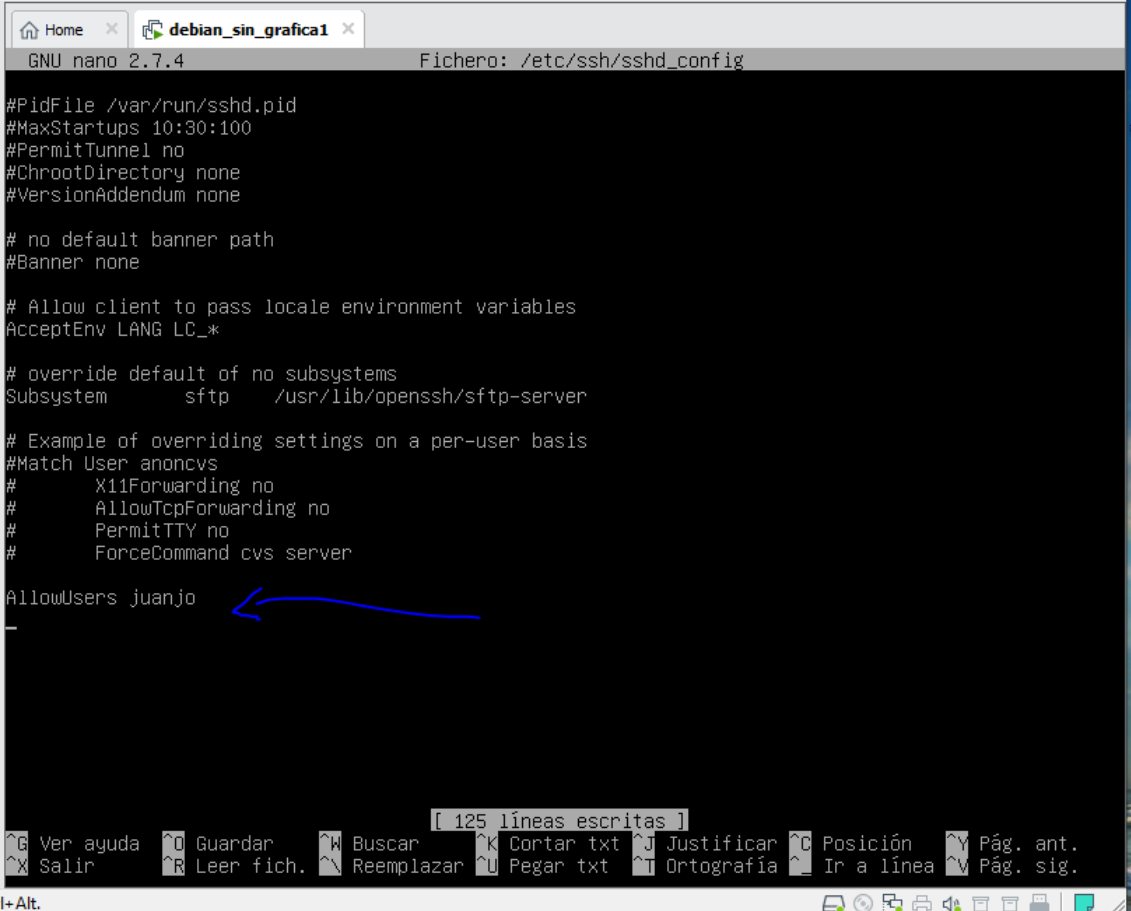
Por último, vemos todo el tráfico que entra y sale de la máquina a través de nuestra nueva cadena:

```
iptables -A INPUT -j allowed_ips
```

```
iptables -A OUTPUT -j allowed_ips
```

Pasemos a la otra opción que había, crear una lista blanca por usuario:

Simplemente tendríamos que editar el archivo `sshd_config` de esta manera:



```
GNU nano 2.7.4 Fichero: /etc/ssh/sshd_config

#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

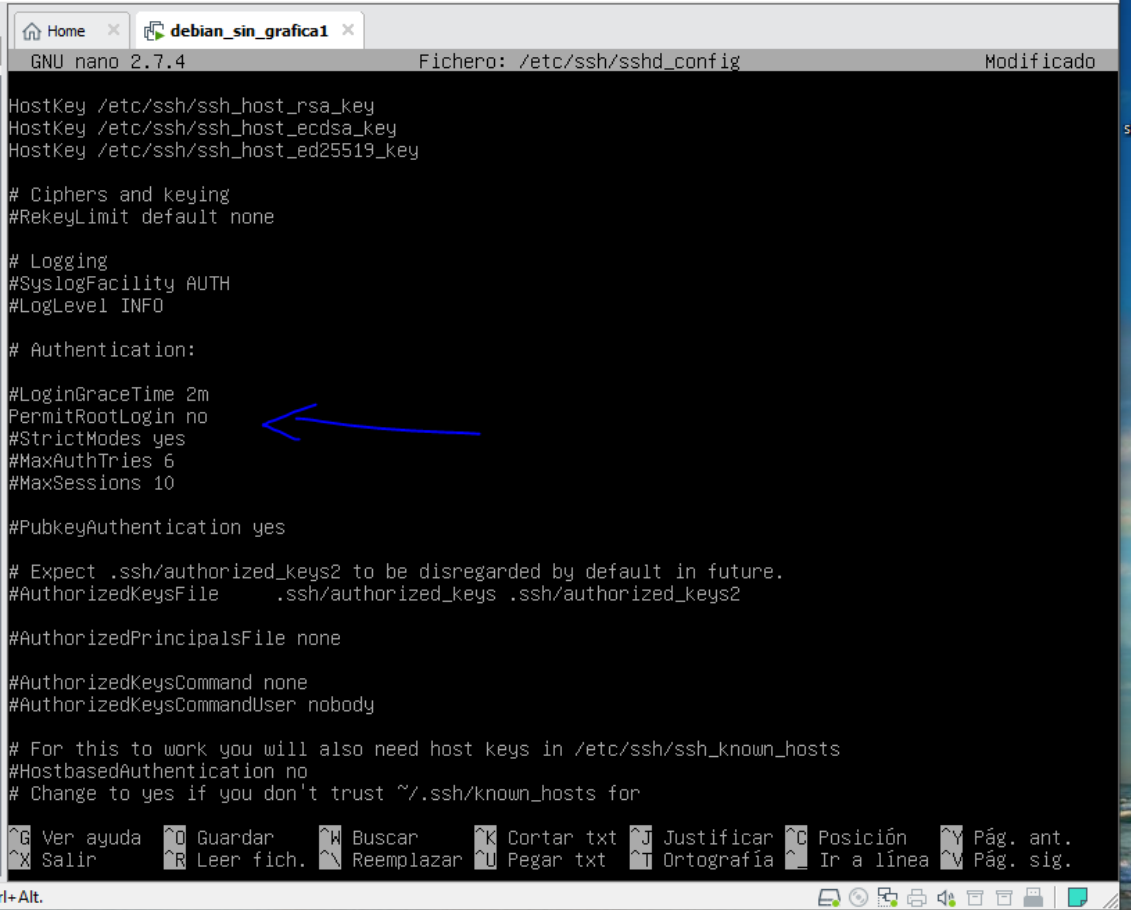
AllowUsers juanjo
_
```

Colocando usuarios detrás del `AllowUsers` daríamos acceso a hacer ssh con dicho usuario.

3. Permitir acceso a root solo con clave instalada en el servidor.

Por defecto, cualquier usuario en el sistema operativo que tenga permisos de Shell, podrá iniciar sesión en el servidor. Además, debemos tener en cuenta que si tenemos activado el usuario root, también podrá conectarse al servidor de forma local o remota, evitando al atacante tener que «adivinar» el nombre de usuario. Por defecto, los bots siempre intentan atacar el puerto 22 y al usuario «root». Permitiendo al propio usuario root el acceso con clave instalada conseguiremos tener una mayor seguridad:

Dentro del archivo `sshd_config` podemos permitir solo el acceso a root con la clave instalada en el servidor. De esta manera:



```
GNU nano 2.7.4 Fichero: /etc/ssh/sshd_config Modificado
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

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición ^Y Pág. ant.
^X Salir ^R Leer fich. ^N Reemplazar ^U Pegar txt ^T Ortografía ^I Ir a línea ^V Pág. sig.
rl+Alt.
```

4. Activar el registro de logs para el servicio sshd.

Con el comando `journalctl -u ssh` ó `journalctl -t sshd` puedes consultar los logs de esta manera:

```
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 20:51:31 CEST. --
oct 11 19:13:05 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 19:13:08 debian sshd[405]: Server listening on 0.0.0.0 port 22.
oct 11 19:13:08 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 19:14:06 debian systemd[1]: Reloading OpenBSD Secure Shell server.
oct 11 19:14:06 debian sshd[405]: Received SIGHUP; restarting.
oct 11 19:14:06 debian systemd[1]: Reloaded OpenBSD Secure Shell server.
oct 11 19:14:06 debian sshd[405]: Server listening on 0.0.0.0 port 22.
oct 11 19:14:06 debian sshd[405]: Server listening on :: port 22.
oct 11 19:18:11 debian systemd[1]: Reloading OpenBSD Secure Shell server.
oct 11 19:18:11 debian sshd[405]: Received SIGHUP; restarting.
oct 11 19:18:11 debian sshd[405]: Server listening on 0.0.0.0 port 22.
oct 11 19:18:11 debian sshd[405]: Server listening on :: port 22.
oct 11 19:18:11 debian systemd[1]: Reloaded OpenBSD Secure Shell server.
oct 11 20:10:22 debian systemd[1]: Reloading OpenBSD Secure Shell server.
oct 11 20:10:22 debian sshd[405]: Received SIGHUP; restarting.
oct 11 20:10:22 debian sshd[405]: Server listening on 0.0.0.0 port 22445.
oct 11 20:10:22 debian systemd[1]: Reloaded OpenBSD Secure Shell server.
oct 11 20:10:22 debian sshd[405]: Server listening on :: port 22445.
oct 11 20:10:47 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:10:47 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:10:47 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:10:47 debian sshd[1042]: Server listening on 0.0.0.0 port 22445.
oct 11 20:10:47 debian sshd[1042]: Server listening on :: port 22445.
oct 11 20:10:47 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:24:26 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:24:26 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:24:26 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:24:26 debian sshd[1099]: Server listening on 0.0.0.0 port 22445.
oct 11 20:24:26 debian sshd[1099]: Server listening on :: port 22445.
oct 11 20:24:26 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:42:03 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:42:03 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:42:03 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:42:03 debian sshd[1193]: Server listening on 0.0.0.0 port 22.
lines 1-36
```

5. Probar conexión varias veces, con fallo y de manera exitosa.

En este caso hemos provocado un fallo y nos lo muestra de esta manera:

```

oct 11 20:10:22 debian sshd[405]: Server listening on 0.0.0.0 port 22445.
oct 11 20:10:22 debian systemd[1]: Reloaded OpenBSD Secure Shell server.
oct 11 20:10:22 debian sshd[405]: Server listening on :: port 22445.
oct 11 20:10:47 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:10:47 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:10:47 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:10:47 debian sshd[1042]: Server listening on 0.0.0.0 port 22445.
oct 11 20:10:47 debian sshd[1042]: Server listening on :: port 22445.
oct 11 20:10:47 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:24:26 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:24:26 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:24:26 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:24:26 debian sshd[1099]: Server listening on 0.0.0.0 port 22445.
oct 11 20:24:26 debian sshd[1099]: Server listening on :: port 22445.
oct 11 20:24:26 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:42:03 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:42:03 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:42:03 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:42:03 debian sshd[1193]: Server listening on 0.0.0.0 port 22.
oct 11 20:42:03 debian sshd[1193]: Server listening on :: port 22.
oct 11 20:42:03 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:42:50 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:42:50 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:42:50 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:42:50 debian sshd[1230]: Server listening on 0.0.0.0 port 22.
oct 11 20:42:50 debian sshd[1230]: Server listening on :: port 22.
oct 11 20:42:50 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:44:32 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:44:32 debian sshd[1271]: Server listening on 0.0.0.0 port 22.
oct 11 20:44:32 debian sshd[1271]: Server listening on :: port 22.
oct 11 20:44:32 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:51:22 debian sshd[1321]: Accepted password for juanjo from 192.168.1.31 port 60022 ssh2
oct 11 20:55:29 debian sshd[1353]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
oct 11 20:55:31 debian sshd[1353]: Failed password for juanjo from 192.168.1.31 port 60024 ssh2
lines 18-53/53 (END)

```

- Mostrar los últimos 10 mensajes generados por el servicio sshd y, exportarlos en formato json.

Para mostrar los últimos 10 mensajes sería algo tal que así:

```

^[[[root@debian:/home/juanjo# journalctl -u ssh -n
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 20:55:33 CEST. --
oct 11 20:42:50 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:44:32 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:44:32 debian sshd[1271]: Server listening on 0.0.0.0 port 22.
oct 11 20:44:32 debian sshd[1271]: Server listening on :: port 22.
oct 11 20:44:32 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:51:22 debian sshd[1321]: Accepted password for juanjo from 192.168.1.31 port 60022 ssh2
oct 11 20:55:29 debian sshd[1353]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
oct 11 20:55:31 debian sshd[1353]: Failed password for juanjo from 192.168.1.31 port 60024 ssh2
lines 1-11/11 (END)
root@debian:/home/juanjo# _

```

Para que estos últimos 10 mensajes se exporten en formato json sería de esta manera:

Journalctl -u ssh -n > /home/juanjo/ssh.json

```
GNU nano 2.7.4 Fichero: ssh.json
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 21:00:04 CEST. --
oct 11 20:42:50 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Stopping OpenBSD Secure Shell server...
oct 11 20:44:32 debian systemd[1]: Stopped OpenBSD Secure Shell server.
oct 11 20:44:32 debian systemd[1]: Starting OpenBSD Secure Shell server...
oct 11 20:44:32 debian sshd[1271]: Server listening on 0.0.0.0 port 22.
oct 11 20:44:32 debian sshd[1271]: Server listening on :: port 22.
oct 11 20:44:32 debian systemd[1]: Started OpenBSD Secure Shell server.
oct 11 20:51:22 debian sshd[1321]: Accepted password for juanjo from 192.168.1.31 port 60022 ssh2
oct 11 20:55:29 debian sshd[1353]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid$
oct 11 20:55:31 debian sshd[1353]: Failed password for juanjo from 192.168.1.31 port 60024 ssh2

[ 11 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar txt ^J Justificar ^C Posición  ^Y Pág. ant.
^X Salir      ^R Leer fich. ^E Reemplazar ^U Pegar txt  ^T Ortografía ^_ Ir a línea ^V Pág. sig.
```

7. Configurar journalctl para hacer persistentes los registros, pero que no ocupen más de 100MB de espacio en disco.

Con SystemMaxFileSize se determina el tamaño máximo del archivo de registro y con SytemMaxFiles se determina el número máximo de ficheros de registros que puede haber.

8. Mostrar los registros autorizados de acceso remoto de la últimos 5 usuarios.

Para mostrar los registros de los últimos 5 usuarios primero listamos los usuarios que han creado registros log con:

Journalctl -list-boots

Por último lo con “journalctl -t sshd -b” vemos el ultimo usuario que ha añadido registros logs y que se haya autorizado con acceso remoto. Con opciones como -0 -1 -2 podemos ver los distintos usuarios que se han autorizado remotamente, jemplo: “journalctl -y sshd -b -1”, “journalctl -y sshd -b -2”, “journalctl -y sshd -b -3”, etc.

9. Revisar los intentos con error de acceso generados por el servicio sshd en las últimas 24 horas.

Primero vemos durante que transcurso del tiempo ha estado recopilando información los logs:

```
root@debian:/home/juanjo# journalctl --until '24:00'
Failed to parse timestamp: 24:00
root@debian:/home/juanjo# journalctl --until '00:00'
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 21:04:13 CEST. --
root@debian:/home/juanjo# _
```

Posteriormente vemos los “warning” o peligros de los logs para el servidor de ssh:

```
Component is omitted, '00' is assumed. If the date component is omitted, the current
root@debian:/home/juanjo# journalctl --priority=waaarning
Unknown log level waaarning
root@debian:/home/juanjo# journalctl --priority=warning
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 21:17:01 CEST. --
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'cpu cycles' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'instructions' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'bus cycles' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'cache references' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'cache misses' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'branch instructions' unavailable
oct 11 19:12:54 debian kernel: core: CPUID marked event: 'branch misses' unavailable
oct 11 19:12:54 debian kernel: NMI watchdog: disabled (cpu0): hardware events not enabled
oct 11 19:12:54 debian kernel: piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
oct 11 19:12:54 debian kernel: LSI53C1030 B0:
oct 11 19:12:54 debian kernel: Capabilities={
oct 11 19:12:54 debian kernel: Initiator
oct 11 19:12:54 debian kernel: }
oct 11 19:12:54 debian kernel: sd 0:0:0:0: [sda] Assuming drive cache: write through
oct 11 19:18:04 debian dhclient[573]: send_packet: Network is unreachable
oct 11 19:18:04 debian dhclient[573]: send_packet: please consult README file regarding broadcast
oct 11 19:18:04 debian dhclient[573]: dhclient.c:2733: Failed to send 300 byte long packet over
lines 1-18/18 (END)
root@debian:/home/juanjo# journalctl -t sshd --priority=warning
-- No entries --
root@debian:/home/juanjo# journalctl -u ssh --priority=warning
-- No entries --
root@debian:/home/juanjo# _
```

Y por último vemos los errores de los logs para el servidor de ssh:

```
root@debian:/home/juanjo# journalctl -u ssh --priority=err
-- No entries --
root@debian:/home/juanjo# journalctl -t sshd --priority=err
-- No entries --
root@debian:/home/juanjo# journalctl --priority=err
-- Logs begin at Fri 2019-10-11 19:12:54 CEST, end at Fri 2019-10-11 21:17:01 CEST. --
oct 11 19:12:54 debian kernel: piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
oct 11 19:12:54 debian kernel: sd 0:0:0:0: [sda] Assuming drive cache: write through
oct 11 19:18:04 debian dhclient[573]: send_packet: Network is unreachable
oct 11 19:18:04 debian dhclient[573]: send_packet: please consult README file regarding broadcast
oct 11 19:18:04 debian dhclient[573]: dhclient.c:2733: Failed to send 300 byte long packet over
lines 1-6/6 (END)
```

En mi caso no hay ninguna entrada de error ya que no se ha dado el caso.