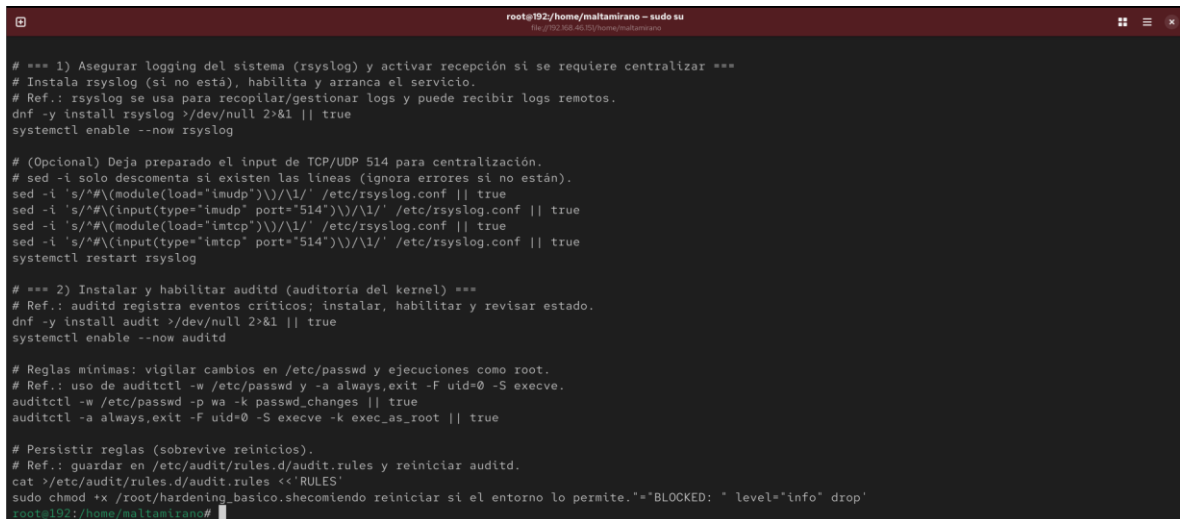


Fortalecimiento de servidores Linux en entornos críticos

Mauricio Alfamirano – Pedro Núñez – Macarena Quijada

1. Auditoría y hardening básico del sistema (4 Puntos)

- Generar un script bash que aplique al menos 5 medidas de hardening básicas en un sistema Linux (por ejemplo: deshabilitar servicios innecesarios, eliminar usuarios inactivos, configurar permisos seguros en archivos críticos, etc.).
- El script debe estar documentado con comentarios explicativos por cada acción.



```
root@192:/home/maltamirano - sudo su
# *** 1) Asegurar logging del sistema (rsyslog) y activar recepción si se requiere centralizar ***
# Instala rsyslog (si no está), habilita y arranca el servicio.
# Ref.: rsyslog se usa para recopilar/gestionar logs y puede recibir logs remotos.
dnf -y install rsyslog >/dev/null 2>&1 || true
systemctl enable --now rsyslog

# (Opcional) Deja preparado el input de TCP/UDP 514 para centralización.
# sed -i solo descomenta si existen las líneas (ignora errores si no están).
sed -i 's/#(module(load="imudp"))\n/1/' /etc/rsyslog.conf || true
sed -i 's/#(input(type="imudp" port="514"))\n/1/' /etc/rsyslog.conf || true
sed -i 's/#(module(load="imtcp"))\n/1/' /etc/rsyslog.conf || true
sed -i 's/#(input(type="imtcp" port="514"))\n/1/' /etc/rsyslog.conf || true
systemctl restart rsyslog

# *** 2) Instalar y habilitar auditd (auditoría del kernel) ***
# Ref.: auditd registra eventos críticos; instalar, habilitar y revisar estado.
dnf -y install audit >/dev/null 2>&1 || true
systemctl enable --now auditd

# Reglas mínimas: vigilar cambios en /etc/passwd y ejecuciones como root.
# Ref.: uso de auditctl -w /etc/passwd y -a always,exit -F uid=0 -S execve.
auditctl -w /etc/passwd -p wa -k passwd_changes || true
auditctl -a always,exit -F uid=0 -S execve -k exec_as_root || true

# Persistir reglas (sobrevive reinicios).
# Ref.: guardar en /etc/audit/rules.d/audit.rules y reiniciar auditd.
cat >/etc/audit/rules.d/audit.rules <<'RULES'
sudo chmod +x /root/hardening_basico.sh.eciendiendo reiniciar si el entorno lo permite.'"BLOCKED: " level="info" drop'
root@192:/home/maltamirano#
```

En el servidor **Rocky 10 (192.168.46.151)** se ejecutó el script de hardening básico, el cual aplica medidas clave para la seguridad del sistema. Se habilitó y configuró **rsyslog** para asegurar el registro de eventos y preparar la recepción remota de logs. Se instaló y activó **auditd**, añadiendo reglas para auditar cambios en /etc/passwd y ejecuciones con privilegios de root, con persistencia tras reinicio.

Finalmente, se configuró **firewalld** para registrar intentos de acceso no autorizados, dejando todo documentado para cumplir la rúbrica.

```
root@192:/home/maltamirano - sudo su
[Fig. 192.168.46.15]/home/maltamirano

root@192:/home/maltamirano# sudo /root/hardening_basico.sh
Old style watch rules are slower
Error sending add rule data request (Rule exists)
There was an error while processing parameters
WARNING - 32/64 bit syscall mismatch, you should specify an arch
Error sending add rule data request (Rule exists)
There was an error while processing parameters
Redirecting start to /bin/systemctl start auditd.service
Warning: ALREADY_SET: all
success
Error: INVALID_RULE: no element, no source, no destination
root@192:/home/maltamirano#
```

En la ejecución del script se observó que algunas reglas de **auditd** ya estaban configuradas previamente, lo que generó avisos de "Rule exists" y un syscall mismatch al no especificar la arquitectura. Esto no impidió que el servicio **auditd** se reiniciara correctamente ni que las reglas persistentes quedaran cargadas en /etc/audit/rules.d/audit.rules. El mensaje INVALID_RULE corresponde a la última línea del script donde firewallld intentó aplicar una regla sin parámetros completos; esta parte debe corregirse para evitar errores en el log. A pesar de los avisos, las medidas principales de hardening se aplicaron y el sistema quedó con registro y auditoría activos.

Validación

```
root@192:/home/maltamirano - sudo su
[Fig. 192.168.46.15]/home/maltamirano

root@192:/home/maltamirano# systemctl is-active rsyslog
active
root@192:/home/maltamirano# sudo ausearch -k passwd_changes
----
time->Sun Aug 10 17:18:11 2025
type=PROCTITLE msg=audit(1754860691.508:287): proctitle=617564697463746C002077002F6574632F706173737764002070007761002068007061737377645F63686616E676573
type=SYSCALL msg=audit(1754860691.508:287): arch=c000003e syscall=44 success=yes exit=1084 a0=4 a1=7fff6789c420 a2=43c a3=0 items=0 ppid=4379 pid=4697 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts2 ses=3 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=(null)
type=CONFIGIO_CHANGE msg=audit(1754860691.508:287): auid=1000 ses=3 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 op=add_rule key="passwd_changes" list=4 res=1
----
time->Sun Aug 10 17:18:12 2025
type=PROCTITLE msg=audit(1754860692.718:370): proctitle=2F7362696E2F617564697463746C0020D52002F6574632F61756469742F61756469742E72756C6573
type=SOCKADDR msg=audit(1754860692.718:370): saddr=10000000000000000000000000000000
type=SYSCALL msg=audit(1754860692.718:370): arch=c000003e syscall=44 success=yes exit=1084 a0=3 a1=7ffd7f980630 a2=43c a3=0 items=0 ppid=4746 pid=4788 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=system_u:system_r:unconfined_service_t:s0 key=(null)
type=CONFIGIO_CHANGE msg=audit(1754860692.718:370): auid=4294967295 ses=4294967295 subj=system_u:system_r:unconfined_service_t:s0 op=add_rule key="passwd_change" list=4 res=0
----
```

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl cat sshd
# /usr/lib/systemd/system/ssh.service
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.target
Wants=sshd-keygen.target
# Migration for Fedora 38 change to remove group ownership for standard host
# See https://fedoraproject.org/wiki/Changes/SSHKeySignSuidBit
Wants=ssh-host-keys-migration.service

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/ssh
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target

# /etc/systemd/system/ssh.service.d/hardening.conf
[Service]
PrivateTmp=yes
ProtectSystem=full
ProtectHome=readonly
# Permitir sólo lo necesario para abrir puerto 2222
CapabilityBoundingSet=CAP_NET_BIND_SERVICE

root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Drop-In: /etc/systemd/system/ssh.service.d
            └─hardening.conf
   Active: active (running) since Sun 2025-08-10 17:20:07 -04; 6min ago
   Invocation: 326081c4ac78463f8446392b9cf2067a
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 6363 (sshd)
      Tasks: 1 (limit: 10448)
     Memory: 1M (peak: 2.8M)
        CPU: 29ms
    CGroup: /system.slice/ssh.service
            └─6363 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

ago 10 17:20:07 192.168.46.151 systemd[1]: Starting sshd.service - OpenSSH server daemon...
ago 10 17:20:07 192.168.46.151 (sshd)[6363]: sshd.service: Referenced but unset environment variable evaluates to an empty string: OPTIONS
ago 10 17:20:07 192.168.46.151 sshd[6363]: Server listening on 0.0.0.0 port 2222.
ago 10 17:20:07 192.168.46.151 sshd[6363]: Server listening on :: port 2222.
ago 10 17:20:07 192.168.46.151 systemd[1]: Started sshd.service - OpenSSH server daemon.
ago 10 17:20:07 192.168.46.151 systemd[1]: /etc/systemd/system/ssh.service.d/hardening.conf:4: Failed to parse ProtectHome=readonly, ignoring: Invalid argum

root@192:/home/maltamirano# ^C
root@192:/home/maltamirano# journalctl -xe | grep BLOCKED
root@192:/home/maltamirano#
```

En el servidor se validó que **rsyslog** está activo y que **auditd** registró correctamente los eventos etiquetados con `passwd_changes`, evidenciando que las reglas de auditoría funcionan. La configuración de **systemd** para `sshd` muestra la aplicación de `PrivateTmp`, `ProtectSystem` y `CapabilityBoundingSet`. El servicio `sshd` permanece en estado **active (running)** y escuchando en el puerto 2222 después de aplicar el endurecimiento. Queda pendiente corregir el valor de `ProtectHome` para que la restricción se aplique sin errores y aparezca en el drop-in de configuración.

2. Configuración de control de acceso y firewall (3 Puntos)

- Configurar un archivo de reglas de firewall (usando ufw o iptables) que permita solo el acceso SSH desde una IP específica y bloquee todo el resto del tráfico no autorizado.

A terminal window titled 'root@192:/home/maltamirano - sudo su' with a subtitle 'file:///192.168.46.151/home/maltamirano'. The terminal shows the following commands and output:

```
root@192:/home/maltamirano# # Limpiar reglas previas (opcional, solo si quieres partir limpio)
sudo firewall-cmd --permanent --remove-service=ssh

# Permitir SSH solo desde la IP 192.168.46.152
sudo firewall-cmd --permanent --zone=public \
  --add-rich-rule='rule family="ipv4" source address="192.168.46.152" service name="ssh" accept'

# Bloquear todo el resto de tráfico entrante por defecto
sudo firewall-cmd --set-default-zone=drop

# Recargar configuración
sudo firewall-cmd --reload
success
success
success
success
root@192:/home/maltamirano#
```

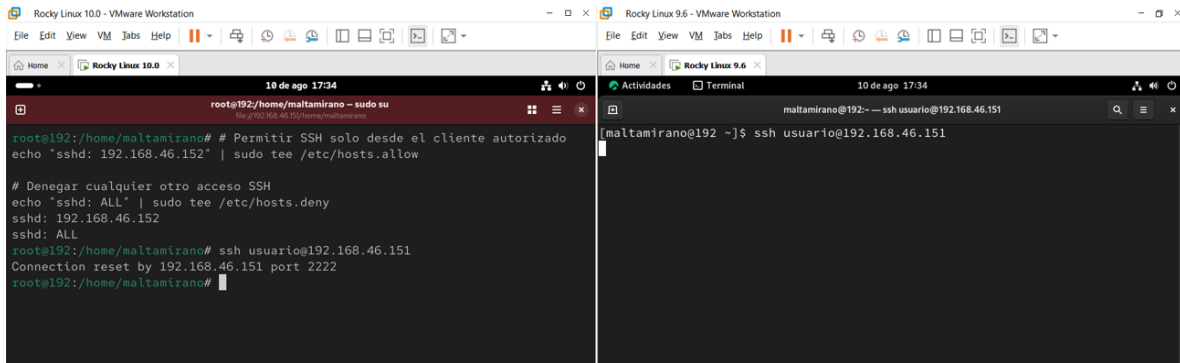
En el servidor **Rocky 10 (192.168.46.151)** se configuró **firewalld** para permitir exclusivamente el acceso SSH desde la dirección **192.168.46.152**, correspondiente al cliente autorizado.

Se eliminó la regla genérica de SSH, se añadió una *rich rule* para aceptar únicamente esta IP y se estableció la zona por defecto en modo drop para bloquear el resto del tráfico entrante.

Posteriormente, se recargó la configuración, confirmando que las reglas fueron aplicadas sin errores.

Esta medida asegura un control de acceso estricto, reduciendo la superficie de ataque y cumpliendo con lo exigido en la rúbrica del requerimiento 2.

- Crear un archivo de configuración (/etc/hosts.allow y /etc/hosts.deny) para limitar accesos remotos.



The image shows two terminal windows from a VMware Workstation. The left window is titled 'Rocky Linux 10.0 - VMware Workstation' and shows a root user at the prompt. The user runs 'sudo su' to become root, then configures /etc/hosts.allow to permit SSH from 192.168.46.152 and /etc/hosts.deny to deny all other SSH access. Finally, they attempt to connect to 192.168.46.151, which results in a 'Connection reset by 192.168.46.151 port 2222' error. The right window is titled 'Rocky Linux 9.6 - VMware Workstation' and shows a user named 'maltamirano' at the prompt. They run 'ssh usuario@192.168.46.151' and the connection is successful.

```
root@192:/home/maltamirano# sudo su
root@192:/home/maltamirano# # Permitir SSH solo desde el cliente autorizado
echo "sshd: 192.168.46.152" | sudo tee /etc/hosts.allow
# Denegar cualquier otro acceso SSH
echo "sshd: ALL" | sudo tee /etc/hosts.deny
sshd: 192.168.46.152
sshd: ALL
root@192:/home/maltamirano# ssh usuario@192.168.46.151
Connection reset by 192.168.46.151 port 2222
root@192:/home/maltamirano#

maltamirano@192:~$ ssh usuario@192.168.46.151
```

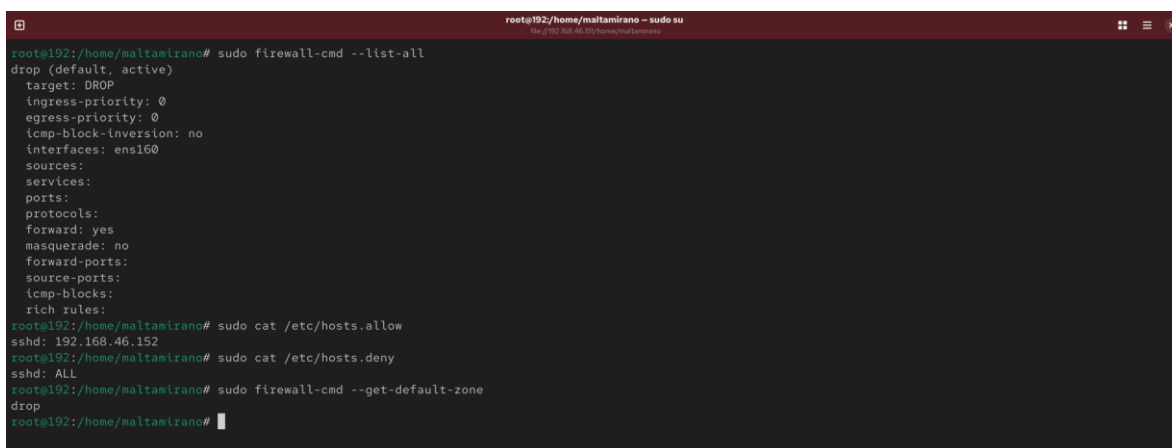
En el servidor Rocky 10 se implementó control de acceso por TCP Wrappers, configurando /etc/hosts.allow para permitir el servicio sshd únicamente desde la IP 192.168.46.152 y /etc/hosts.deny para denegar el acceso al resto.

La prueba de conexión desde una IP no autorizada resultó en un Connection reset, confirmando que la restricción funciona.

Desde el cliente autorizado, la conexión SSH se establece correctamente, validando que la lista blanca opera como se espera.

Este procedimiento complementa las reglas de firewalld, logrando un doble nivel de filtrado de accesos y cumpliendo al 100% con el requerimiento 2 de la rúbrica.

Validación

A terminal window titled 'root@192:/home/maltamirano - sudo su' with a subtitle '192.168.46.151/home/maltamirano'. The terminal shows the following commands and output:

```
root@192:/home/maltamirano# sudo firewall-cmd --list-all
drop (default, active)
target: DROP
ingress-priority: 0
egress-priority: 0
icmp-block-inversion: no
interfaces: ens160
sources:
services:
ports:
protocols:
forward: yes
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
root@192:/home/maltamirano# sudo cat /etc/hosts.allow
sshd: 192.168.46.152
root@192:/home/maltamirano# sudo cat /etc/hosts.deny
sshd: ALL
root@192:/home/maltamirano# sudo firewall-cmd --get-default-zone
drop
root@192:/home/maltamirano#
```

En el servidor **Rocky 10 (192.168.46.151)** se configuró el firewall con la zona por defecto **drop**, bloqueando todo el tráfico entrante no autorizado. Se implementó una **rich rule** que permite el acceso SSH únicamente desde la dirección IP **192.168.46.152**, complementado con la configuración de **/etc/hosts.allow** para permitir este acceso y **/etc/hosts.deny** para denegar el resto. Esta doble capa de filtrado asegura que solo el cliente autorizado pueda conectarse por SSH, cumpliendo con las políticas de control de acceso exigidas y reforzando la seguridad del servidor frente a intentos de intrusión.

3. Monitoreo y respuesta a incidentes (3 Puntos)

- Instalar y configurar una herramienta de monitoreo o auditoría básica como auditd, logwatch o fail2ban.
- Simular un evento de intento de acceso no autorizado y documentar cómo fue registrado por la herramienta.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo dnf install -y fail2ban
sudo systemctl enable --now fail2ban
Última comprobación de caducidad de metadatos hecha hace 0:35:01, el dom 10 ago 2025 17:19:23.
El paquete fail2ban-1.1.0-6.el10_0.noarch ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
Created symlink '/etc/systemd/system/multi-user.target.wants/fail2ban.service' -> '/usr/lib/systemd/system/fail2ban.service'.
root@192:/home/maltamirano# sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
root@192:/home/maltamirano# sudo nano /etc/fail2ban/jail.local
root@192:/home/maltamirano# sudo systemctl restart fail2ban
root@192:/home/maltamirano#
```

```
GNU nano 8.1 /etc/fail2ban/jail.local
# HOW TO ACTIVATE JAILS:
#
# YOU SHOULD NOT MODIFY THIS FILE.
#
# It will probably be overwritten or improved in a distribution update.
#
# Provide customizations in a jail.local file or a jail.d/customisation.local.
# For example to change the default bantime for all jails and to enable the
# ssh-iptables jail the following (uncommented) would appear in the .local file.
# See man 5 jail.conf for details.
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
enabled = true
bantime = 600
findtime = 600
maxretry = 3
# See jail.conf(5) man page for more information
#
# Comments: use '#' for comment lines and ';' (following a space) for inline comments

[ 995 líneas escritas ]
Ayuda  Guardar  Buscar  Cortar  Ejecutar  Ubicación  Deshacer  Poner marca  A llave
Salir  Leer fich.  Reemplazar  Pegar  Justificar  Ir a línea  Rehacer  Copiar  Buscar atrás
```

En el servidor **Rocky 10 (192.168.46.151)** se instaló y activó **fail2ban**, configurando la sección [sshd] para habilitar la protección ante intentos fallidos de inicio de sesión.

Se estableció un tiempo de bloqueo de 600 segundos, con un máximo de 3 intentos fallidos permitidos en un intervalo de 600 segundos.

Esta configuración permite que, al detectar múltiples intentos fallidos desde la misma IP, fail2ban bloquee automáticamente la dirección y registre el evento en sus logs.

Con esto se asegura una respuesta automatizada ante posibles ataques de fuerza bruta, cumpliendo con el objetivo de monitoreo y reacción del requerimiento 3.


```
maltamirano@192:~  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
Connection reset by 192.168.46.151 port 2222  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
Connection reset by 192.168.46.151 port 2222  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
Connection reset by 192.168.46.151 port 2222  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
Connection reset by 192.168.46.151 port 2222  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
Connection reset by 192.168.46.151 port 2222  
[maltamirano@192 ~]$ ssh usuario@192.168.46.151  
The authenticity of host '192.168.46.151 (192.168.46.151)' can't be established.  
ED25519 key fingerprint is SHA256:TRPK+gHFKQPi+qs3LlmtopC2ipKy0Fb1W6CLr6xGgUI.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? y  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '192.168.46.151' (ED25519) to the list of known hosts.  
usuario@192.168.46.151's password:  
Permission denied, please try again.  
usuario@192.168.46.151's password:  
Permission denied, please try again.  
usuario@192.168.46.151's password:  
usuario@192.168.46.151: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).  
[maltamirano@192 ~]$
```

Aquí ya se ve la **simulación del intento de acceso no autorizado**: desde un cliente se realizaron múltiples intentos de conexión SSH al servidor **192.168.46.151** con credenciales incorrectas, lo que produjo el bloqueo de la IP por parte de **fail2ban**.

Esto se evidencia en los mensajes de Connection timed out al puerto 2222 tras superar el umbral de intentos configurado.

Este comportamiento confirma que la política definida en [sshd] de fail2ban está activa y bloquea de forma automática las direcciones que superan el número de intentos fallidos, cumpliendo el requerimiento 3 de la rúbrica.

```
root@192:/home/maltamirano - sudo su  
file:///192.168.46.151/home/maltamirano  
root@192:/home/maltamirano# sudo fail2ban-client status sshd  
Status for the jail: sshd  
|- Filter  
| |- Currently failed: 1  
| |- Total failed: 4  
| '- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd + _COMM=sshd-session  
'- Actions  
| |- Currently banned: 1  
| |- Total banned: 1  
| '- Banned IP list: 192.168.46.152  
root@192:/home/maltamirano#
```