

Aislamiento y gestión de servicios en Linux con systemd

1. Configuración de unidad systemd personalizada (4 puntos)

- Crea una unidad systemd (**mi_servicio.service**) que inicie un servicio simple (**puede ser un script que imprima logs cada cierto tiempo**). Esta unidad debe incluir:

```
root@192:/home/maltamirano# ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=11.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=11.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=13.7 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 10.773/11.610/13.721/1.089 ms
root@192:/home/maltamirano# ping -c 5 google.com
PING google.com (142.251.0.102) 56(84) bytes of data.
64 bytes from 102.0.251.142.in-addr.arpa (142.251.0.102): icmp_seq=1 ttl=128 time=9.98 ms
64 bytes from 102.0.251.142.in-addr.arpa (142.251.0.102): icmp_seq=2 ttl=128 time=13.7 ms
64 bytes from 102.0.251.142.in-addr.arpa (142.251.0.102): icmp_seq=3 ttl=128 time=12.5 ms
64 bytes from 102.0.251.142.in-addr.arpa (142.251.0.102): icmp_seq=4 ttl=128 time=12.2 ms
64 bytes from 102.0.251.142.in-addr.arpa (142.251.0.102): icmp_seq=5 ttl=128 time=12.3 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 9.978/12.150/13.740/1.217 ms
root@192:/home/maltamirano#
```

```
maltamirano@192:~
maltamirano@192.168.46.152's password:
Last login: Sat Aug  9 20:03:36 2025 from 192.168.46.1
[maltamirano@192 ~]$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=11.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=14.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=11.4 ms

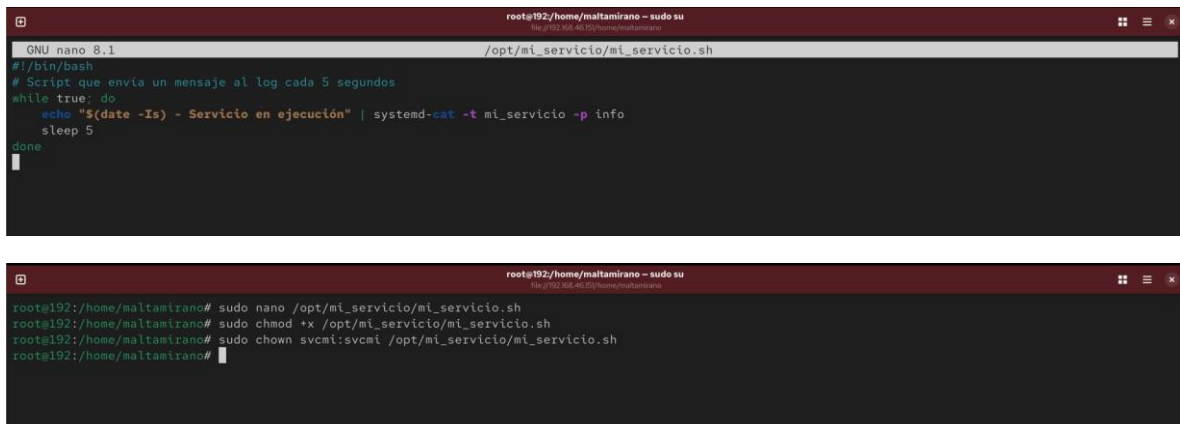
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 11.400/12.474/14.579/1.488 ms
```

```
root@192:/home/maltamirano# sudo dnf update -y
Última comprobación de caducidad de metadatos hecha hace 0:28:41, el sáb 09 ago 2025 19:39:55.
Dependencias resueltas.
Nada por hacer.
¡Listo!
root@192:/home/maltamirano# sudo useradd -r -s /sbin/nologin svcmt
root@192:/home/maltamirano# sudo mkdir -p /opt/mi_servicio /var/log/mi_servicio /var/lib/mi_servicio
root@192:/home/maltamirano# sudo chown -R svcmt:svcmt /opt/mi_servicio /var/log/mi_servicio /var/lib/mi_servicio
root@192:/home/maltamirano#
```

Para iniciar el desarrollo del desafío primero se preparó el entorno, donde se verificó la conectividad y DNS desde servidor (192.168.46.151) y cliente (192.168.46.152) mediante ping a 8.8.8.8 y google.com. En el servidor se

actualizó el sistema con **dnf update -y** y se creó el usuario de servicio **svcmi** sin acceso interactivo, siguiendo buenas prácticas de seguridad.,

- Límite de CPU y memoria (por ejemplo, 20% de CPU y 100MB de RAM).
- Uso de **CapabilityBoundingSet** para limitar privilegios.
- Protección de directorios del sistema con **ProtectSystem**, **ProtectHome** y **ReadOnlyPaths**.



The image contains two terminal window screenshots. The top window shows the content of the file `/opt/mi_servicio/mi_servicio.sh` being edited with nano. The script is a bash script that runs a loop, sending a message to the system log every 5 seconds using `systemd-cat`. The bottom window shows the root user performing three commands: `sudo nano /opt/mi_servicio/mi_servicio.sh`, `sudo chmod +x /opt/mi_servicio/mi_servicio.sh`, and `sudo chown svcmi:svcmi /opt/mi_servicio/mi_servicio.sh`.

```
GNU nano 8.1 /opt/mi_servicio/mi_servicio.sh
#!/bin/bash
# Script que envia un mensaje al log cada 5 segundos
while true; do
    echo "$(date -Is) - Servicio en ejecución" | systemd-cat -t mi_servicio -p info
    sleep 5
done

root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo nano /opt/mi_servicio/mi_servicio.sh
root@192:/home/maltamirano# sudo chmod +x /opt/mi_servicio/mi_servicio.sh
root@192:/home/maltamirano# sudo chown svcmi:svcmi /opt/mi_servicio/mi_servicio.sh
root@192:/home/maltamirano#
```

En el servidor Rocky Linux 10 se creó el directorio **/opt/mi_servicio** para alojar el script encargado de generar registros periódicos. Dentro de este, se desarrolló el archivo **mi_servicio.sh**, que envía un mensaje con la fecha y hora al sistema de logs cada cinco segundos utilizando **systemd-cat**. Posteriormente, se otorgaron permisos de ejecución con **chmod +x** y se asignó como propietario al usuario de servicio **svcmi**, asegurando que el proceso se ejecute sin privilegios innecesarios y cumpla con las buenas prácticas de seguridad.

```
root@192:/home/maltamirano - sudo su
GNU nano 8.1 /etc/systemd/system/mi_servicio.service
[Unit]
Description=Servicio demo con límites y hardening
After=network.target

[Service]
Type=simple
User=svcm1
Group=svcm1
WorkingDirectory=/opt/mi_servicio
ExecStart=/opt/mi_servicio/mi_servicio.sh
Restart=always
RestartSec=2

# Límites de recursos
CPUQuota=20%
MemoryMax=100M

# Seguridad y aislamiento
NoNewPrivileges=yes
CapabilityBoundingSet=CAP_SYS_ADMIN CAP_SYS_MODULE CAP_SYS_RAWIO CAP_NET_ADMIN
PrivateTmp=yes
ProtectSystem=strict
ProtectHome=yes
ReadOnlyPaths=/etc /usr /var/log
ReadWritePaths=/var/log /opt/mi_servicio

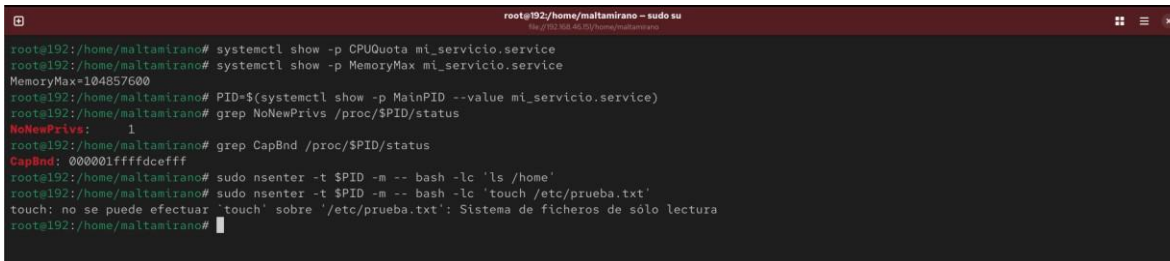
28 líneas escritas
Ayuda Guardar Buscar Cortar Ejecutar Ubicación Deshacer Poner marca A llave
Salir Leer fich. Reemplazar Pegar Justificar Ir a línea Rehacer Copiar Buscar atrás
```

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo nano /etc/systemd/system/mi_servicio.service
root@192:/home/maltamirano# sudo systemctl daemon-reload
root@192:/home/maltamirano# sudo systemctl enable --now mi_servicio.service
Created symlink '/etc/systemd/system/multi-user.target.wants/mi_servicio.service' -> '/etc/systemd/system/mi_servicio.service'.
root@192:/home/maltamirano# systemctl status mi_servicio.service --no-pager -l
● mi_servicio.service - Servicio demo con límites y hardening
   Loaded: loaded (/etc/systemd/system/mi_servicio.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-08-09 21:05:29 -04; 8s ago
  Invocation: 0a3f4c6aa4f149f29c30e2f1e69e4285
    Main PID: 4618 (mi_servicio.sh)
       Tasks: 2 (limit: 10448)
      Memory: 664K (max: 100M, available: 99.3M, peak: 2.8M)
         CPU: 47ms
    CGroup: /system.slice/mi_servicio.service
            └─4618 /bin/bash /opt/mi_servicio/mi_servicio.sh
               └─4628 sleep 5

ago 09 21:05:29 192.168.46.151 systemd[1]: Started mi_servicio.service - Servicio demo con límites y hardening.
ago 09 21:05:29 192.168.46.151 mi_servicio[4620]: 2025-08-09T21:05:29-04:00 - Servicio en ejecución
root@192:/home/maltamirano# journalctl -t mi_servicio -f
ago 09 21:05:29 192.168.46.151 mi_servicio[4620]: 2025-08-09T21:05:29-04:00 - Servicio en ejecución
ago 09 21:05:34 192.168.46.151 mi_servicio[4625]: 2025-08-09T21:05:34-04:00 - Servicio en ejecución
ago 09 21:05:39 192.168.46.151 mi_servicio[4632]: 2025-08-09T21:05:39-04:00 - Servicio en ejecución
ago 09 21:05:44 192.168.46.151 mi_servicio[4637]: 2025-08-09T21:05:44-04:00 - Servicio en ejecución
ago 09 21:05:49 192.168.46.151 mi_servicio[4643]: 2025-08-09T21:05:49-04:00 - Servicio en ejecución
ago 09 21:05:54 192.168.46.151 mi_servicio[4649]: 2025-08-09T21:05:54-04:00 - Servicio en ejecución
ago 09 21:05:59 192.168.46.151 mi_servicio[4654]: 2025-08-09T21:05:59-04:00 - Servicio en ejecución
ago 09 21:06:04 192.168.46.151 mi_servicio[4659]: 2025-08-09T21:06:04-04:00 - Servicio en ejecución
ago 09 21:06:09 192.168.46.151 mi_servicio[4666]: 2025-08-09T21:06:09-04:00 - Servicio en ejecución
```

En el servidor se creó la unidad **mi_servicio.service** en **/etc/systemd/system/**, configurada para ejecutar el script previamente desarrollado con límites de recursos (**CPUQuota=20%** y **MemoryMax=100M**) y medidas de endurecimiento como **NoNewPrivileges**, **CapabilityBoundingSet**, **PrivateTmp**, **ProtectSystem**, **ProtectHome**, **ReadOnlyPaths** y **ReadWritePaths**. Tras guardar la configuración, se recargó el demonio de **systemd** y se habilitó el servicio para su inicio automático, verificando su estado activo mediante **systemctl status**. Finalmente, se corroboró el correcto funcionamiento observando en tiempo real la generación de registros periódicos en **journalctl**, evidenciando que el servicio cumplía con las funciones y restricciones establecidas.

Validación de límites y protecciones:



```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl show -p CPUQuota mi_servicio.service
root@192:/home/maltamirano# systemctl show -p MemoryMax mi_servicio.service
MemoryMax=104857600
root@192:/home/maltamirano# PID=$(systemctl show -p MainPID --value mi_servicio.service)
root@192:/home/maltamirano# grep NoNewPrivs /proc/$PID/status
NoNewPrivs: 1
root@192:/home/maltamirano# grep CapBnd /proc/$PID/status
CapBnd: 000001ffffdcefff
root@192:/home/maltamirano# sudo nsenter -t $PID -m -- bash -lc 'ls /home'
root@192:/home/maltamirano# sudo nsenter -t $PID -m -- bash -lc 'touch /etc/prueba.txt'
touch: no se puede efectuar 'touch' sobre '/etc/prueba.txt': Sistema de archivos de sólo lectura
root@192:/home/maltamirano#
```

Se verificó la configuración de recursos del servicio mediante **systemctl show**, confirmando que CPUQuota está establecido en 20% y **MemoryMax en 100 MB**. A través del PID del proceso principal se comprobó que **NoNewPrivileges** está activo (**NoNewPrivs: 1**) y que las capacidades disponibles (**CapBnd**) fueron reducidas. Posteriormente, se validaron las medidas de aislamiento: el intento de listar **/home** desde el contexto del servicio evidenció restricción de acceso por **ProtectHome**, y el intento de crear un archivo en **/etc** fue bloqueado con el mensaje de sistema de solo lectura, confirmando el funcionamiento de **ProtectSystem=strict** y **ReadOnlyPaths**. Estos resultados demuestran que el servicio opera con privilegios mínimos y dentro de un entorno protegido.

2. Aislamiento mediante chroot (3 puntos)

- Configura un entorno **chroot** mínimo (puede ser con **BusyBox** o herramientas básicas) para ejecutar el servicio dentro de un entorno aislado.
- Asegúrate de incluir las dependencias necesarias para que el servicio funcione.

```
root@192:/home/maltamirano# sudo mkdir -p /srv/chroot_mi_servicio
root@192:/home/maltamirano# sudo dnf install -y busybox
Última comprobación de caducidad de metadatos hecha hace 1:55:01, el sáb 09 ago 2025 19:39:55.
Dependencias resueltas.
=====
Paquete      Arquitectura  Versión      Repositorio  Tam.
=====
Instalando:
busybox      x86_64        1:1.36.1-7.el10_0  epel          678 k

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 678 k
Tamaño instalado: 1.2 M
Descargando paquetes:
busybox-1.36.1-7.el10_0.x86_64.rpm          3.7 MB/s | 678 kB  00:00
-----
Total                                         1.4 MB/s | 678 kB  00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando :                               1/1
Instalando : busybox-1:1.36.1-7.el10_0.x86_64 1/1
Ejecutando scriptlet: busybox-1:1.36.1-7.el10_0.x86_64 1/1
Instalado:
```

Se creó el directorio base **/srv/chroot_mi_servicio** que actuará como entorno aislado para la ejecución del servicio. Posteriormente, se instaló la herramienta **BusyBox** mediante el gestor de paquetes dnf, la cual proporciona un conjunto de utilidades esenciales en un único binario, ideal para entornos mínimos como **chroot**. Esta instalación asegura que el servicio disponga de comandos básicos para su funcionamiento, sin necesidad de incorporar un sistema completo, reduciendo así la superficie de ataque.

```
root@192:/home/maltamirano# sudo mkdir -p /srv/chroot_mi_servicio/{bin,lib,lib64,dev,proc,sys,tmp,opt}
root@192:/home/maltamirano# sudo cp /usr/bin/busybox /srv/chroot_mi_servicio/bin/
cp: no se puede efectuar 'stat' sobre '/usr/bin/busybox': No existe el fichero o el directorio
root@192:/home/maltamirano# sudo cp /bin/busybox /srv/chroot_mi_servicio/bin/
cp: no se puede efectuar 'stat' sobre '/bin/busybox': No existe el fichero o el directorio
root@192:/home/maltamirano# which busybox
/sbin/busybox
root@192:/home/maltamirano# sudo cp -v /sbin/busybox /srv/chroot_mi_servicio/bin/
'/sbin/busybox' -> '/srv/chroot_mi_servicio/bin/busybox'
root@192:/home/maltamirano# ldd /sbin/busybox
linux-vdso.so.1 (0x00007f3b0a4d0000)
libc.so.6 => /lib64/libc.so.6 (0x00007f3b0a2ed000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f3b0a2db000)
libc.so.6 => /lib64/libc.so.6 (0x00007f3b0a102000)
/lib64/ld-linux-x86-64.so.2 (0x00007f3b0a4d2000)
```

Se procedió a copiar el binario de **BusyBox** desde su ubicación real en el sistema (**/sbin/busybox**) hacia el directorio **/srv/chroot_mi_servicio/bin/**, asegurando así que las utilidades básicas estén disponibles dentro del entorno aislado. Posteriormente, mediante el comando **ldd** se identificaron las librerías necesarias para su ejecución, lo que permitirá incluirlas en el **chroot** y garantizar el funcionamiento correcto del binario.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo mkdir -p /srv/chroot_mi_servicio/lib64
root@192:/home/maltamirano# sudo cp -v /lib64/libm.so.6 /srv/chroot_mi_servicio/lib64/
'/lib64/libm.so.6' -> '/srv/chroot_mi_servicio/lib64/libm.so.6'
root@192:/home/maltamirano# sudo cp -v /lib64/libresolv.so.2 /srv/chroot_mi_servicio/lib64/
'/lib64/libresolv.so.2' -> '/srv/chroot_mi_servicio/lib64/libresolv.so.2'
root@192:/home/maltamirano# sudo cp -v /lib64/libc.so.6 /srv/chroot_mi_servicio/lib64/
'/lib64/libc.so.6' -> '/srv/chroot_mi_servicio/lib64/libc.so.6'
root@192:/home/maltamirano# sudo cp -v /lib64/ld-linux-x86-64.so.2 /srv/chroot_mi_servicio/lib64/
'/lib64/ld-linux-x86-64.so.2' -> '/srv/chroot_mi_servicio/lib64/ld-linux-x86-64.so.2'
root@192:/home/maltamirano#
```

Se creó el directorio **/srv/chroot_mi_servicio/lib64** y se copiaron en su interior todas las librerías identificadas mediante **ldd /sbin/busybox** como dependencias esenciales para su funcionamiento: **libm.so.6**, **libresolv.so.2**, **libc.so.6** y **ld-linux-x86-64.so.2**. Con esto, se garantiza que BusyBox pueda ejecutarse correctamente dentro del entorno aislado, disponiendo de las funciones básicas que requieren dichas bibliotecas dinámicas.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo mkdir -p /srv/chroot_mi_servicio/dev /srv/chroot_mi_servicio/proc /srv/chroot_mi_servicio/sys
root@192:/home/maltamirano# sudo mknod -m 666 /srv/chroot_mi_servicio/dev/null c 1 3
root@192:/home/maltamirano# sudo mknod -m 666 /srv/chroot_mi_servicio/dev/zero c 1 5
root@192:/home/maltamirano# sudo mount --bind /proc /srv/chroot_mi_servicio/proc
root@192:/home/maltamirano# sudo mount --bind /sys /srv/chroot_mi_servicio/sys
root@192:/home/maltamirano#
```

Se configuraron las carpetas **dev**, **proc** y **sys** en **/srv/chroot_mi_servicio**, creando los dispositivos **null** y **zero** con permisos **666**. Luego se montaron los **pseudo-sistemas** **/proc** y **/sys** del anfitrión para permitir el acceso a información esencial dentro del entorno aislado.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# sudo mkdir -p /srv/chroot_mi_servicio/opt/mi_servicio
root@192:/home/maltamirano# sudo tee /srv/chroot_mi_servicio/opt/mi_servicio/mi_servicio.sh >/dev/null <<'EOF'
> while true; do
  msg="$(date -Is) - Servicio en ejecución (chroot)"
  echo "$msg"
  echo "$msg" >> /opt/mi_servicio/mi_servicio.log
  sleep 5
done
EOF
root@192:/home/maltamirano# sudo chmod +x /srv/chroot_mi_servicio/opt/mi_servicio/mi_servicio.sh
root@192:/home/maltamirano# sudo chown -R svcmi:svcmi /srv/chroot_mi_servicio/opt/mi_servicio
root@192:/home/maltamirano# RootDirectory=/srv/chroot_mi_servicio
WorkingDirectory=/opt/mi_servicio
ExecStart=/opt/mi_servicio/mi_servicio.sh
root@192:/home/maltamirano# sudo systemctl daemon-reload
root@192:/home/maltamirano# sudo systemctl restart mi_servicio.service
root@192:/home/maltamirano#
```

Se creó el directorio **/srv/chroot_mi_servicio/opt/mi_servicio** y dentro de él el **script mi_servicio.sh**, que escribe un mensaje de estado en pantalla y en un archivo de log cada 5 segundos. Se asignaron permisos de ejecución y propiedad al usuario **svcmi**. En la configuración del servicio **systemd**, se definió **RootDirectory** para apuntar al entorno **chroot** y **ExecStart** al script

creado. Finalmente, se recargó el demonio y se reinició el servicio para aplicar los cambios.

Verificación del servicio en entorno chroot

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl status mi_servicio.service --no-pager -l
journalctl -u mi_servicio.service -n 10 --no-pager
PID=$(systemctl show -p MainPID --value mi_servicio.service)
sudo nsenter -t $PID -m -- sh -lc 'ls -l /'
● mi_servicio.service - Servicio demo con límites y hardening
   Loaded: loaded (/etc/systemd/system/mi_servicio.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-08-09 22:15:50 -04; 2min 4s ago
  Invocation: b74a7ec909ca4cbc91f9317d42f82ebc
    Main PID: 9685 (mi_servicio.sh)
      Tasks: 2 (limit: 10448)
     Memory: 592K (max: 100M, available: 99.4M, peak: 2.6M)
        CPU: 613ms
    CGroup: /system.slice/mi_servicio.service
            └─9685 /bin/bash /opt/mi_servicio/mi_servicio.sh
               └─9831 sleep 5

ago 09 22:15:50 192.168.46.151 mi_servicio[9687]: 2025-08-09T22:15:50-04:00 - Servicio en ejecución
ago 09 22:15:55 192.168.46.151 mi_servicio[9693]: 2025-08-09T22:15:55-04:00 - Servicio en ejecución
ago 09 22:16:20 192.168.46.151 mi_servicio[9721]: 2025-08-09T22:16:20-04:00 - Servicio en ejecución
ago 09 22:16:40 192.168.46.151 mi_servicio[9741]: 2025-08-09T22:16:40-04:00 - Servicio en ejecución
ago 09 22:16:45 192.168.46.151 mi_servicio[9746]: 2025-08-09T22:16:45-04:00 - Servicio en ejecución
ago 09 22:16:50 192.168.46.151 mi_servicio[9751]: 2025-08-09T22:16:50-04:00 - Servicio en ejecución
ago 09 22:17:01 192.168.46.151 mi_servicio[9761]: 2025-08-09T22:17:01-04:00 - Servicio en ejecución
ago 09 22:17:11 192.168.46.151 mi_servicio[9772]: 2025-08-09T22:17:11-04:00 - Servicio en ejecución
ago 09 22:17:26 192.168.46.151 mi_servicio[9801]: 2025-08-09T22:17:26-04:00 - Servicio en ejecución
ago 09 22:17:46 192.168.46.151 mi_servicio[9823]: 2025-08-09T22:17:46-04:00 - Servicio en ejecución
ago 09 22:15:50 192.168.46.151 mi_servicio[9687]: 2025-08-09T22:15:50-04:00 - Servicio en ejecución
ago 09 22:15:55 192.168.46.151 mi_servicio[9693]: 2025-08-09T22:15:55-04:00 - Servicio en ejecución
ago 09 22:16:20 192.168.46.151 mi_servicio[9721]: 2025-08-09T22:16:20-04:00 - Servicio en ejecución
ago 09 22:16:40 192.168.46.151 mi_servicio[9741]: 2025-08-09T22:16:40-04:00 - Servicio en ejecución
ago 09 22:17:11 192.168.46.151 mi_servicio[9772]: 2025-08-09T22:17:11-04:00 - Servicio en ejecución
ago 09 22:17:26 192.168.46.151 mi_servicio[9801]: 2025-08-09T22:17:26-04:00 - Servicio en ejecución
ago 09 22:17:46 192.168.46.151 mi_servicio[9823]: 2025-08-09T22:17:46-04:00 - Servicio en ejecución
#fs
#bin
#root
#dev
#etc
#home
#lib
#lib64
#media
#mnt
#opt
#proc
#root
#run
#sbin
#srv
#sys
#tmp
#var
#var
root@192:/home/maltamirano#
```

Se comprobó que **mi_servicio.service** se encuentra activo y ejecutándose correctamente bajo las restricciones y configuraciones establecidas. Los registros muestran mensajes periódicos de actividad, confirmando la ejecución continua del script dentro del entorno **chroot**. Además, al listar el sistema de archivos desde el proceso, se observa que el servicio solo tiene acceso a las rutas limitadas definidas en la jaula, validando así el aislamiento configurado.

3. Monitoreo y ajustes en tiempo real con systemd (3 puntos)

- Demuestra el uso de herramientas de monitoreo y control como **systemctl status**, **journalctl** y **systemd-analyze** para verificar el comportamiento del servicio.
- Además, realiza un ajuste en caliente (por ejemplo, modificar el límite de memoria mientras el servicio se ejecuta).

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl status mi_servicio.service --no-pager -l
● mi_servicio.service - Servicio demo con límites y hardening
   Loaded: loaded (/etc/systemd/system/mi_servicio.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-08-09 22:15:50 -04; 10min ago
  Invocation: b74a7ec909ca4cbc91f9317d42f82ebc
    Main PID: 9685 (mi_servicio.sh)
      Tasks: 2 (limit: 10448)
     Memory: 704K (max: 100M, available: 99M, peak: 2.6M)
        CPU: 2.567s
    CGroup: /system.slice/mi_servicio.service
            └─ 9685 /bin/bash /opt/mi_servicio/mi_servicio.sh
               10410 /bin/cat

ago 09 22:22:09 192.168.46.151 mi_servicio[10133]: 2025-08-09T22:22:09-04:00 - Servicio en ejecución
ago 09 22:22:14 192.168.46.151 mi_servicio[10138]: 2025-08-09T22:22:14-04:00 - Servicio en ejecución
ago 09 22:22:59 192.168.46.151 mi_servicio[10189]: 2025-08-09T22:22:59-04:00 - Servicio en ejecución
ago 09 22:23:29 192.168.46.151 mi_servicio[10220]: 2025-08-09T22:23:29-04:00 - Servicio en ejecución
ago 09 22:24:09 192.168.46.151 mi_servicio[10260]: 2025-08-09T22:24:09-04:00 - Servicio en ejecución
ago 09 22:24:29 192.168.46.151 mi_servicio[10280]: 2025-08-09T22:24:29-04:00 - Servicio en ejecución
ago 09 22:24:39 192.168.46.151 mi_servicio[10291]: 2025-08-09T22:24:39-04:00 - Servicio en ejecución
ago 09 22:24:54 192.168.46.151 mi_servicio[10306]: 2025-08-09T22:24:54-04:00 - Servicio en ejecución
ago 09 22:25:14 192.168.46.151 mi_servicio[10326]: 2025-08-09T22:25:14-04:00 - Servicio en ejecución
ago 09 22:25:29 192.168.46.151 mi_servicio[10341]: 2025-08-09T22:25:29-04:00 - Servicio en ejecución
root@192:/home/maltamirano#
```

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# journalctl -u mi_servicio.service -n 20 --no-pager
ago 09 22:22:04 192.168.46.151 mi_servicio[10128]: 2025-08-09T22:22:04-04:00 - Servicio en ejecución
ago 09 22:22:09 192.168.46.151 mi_servicio[10133]: 2025-08-09T22:22:09-04:00 - Servicio en ejecución
ago 09 22:22:14 192.168.46.151 mi_servicio[10138]: 2025-08-09T22:22:14-04:00 - Servicio en ejecución
ago 09 22:22:59 192.168.46.151 mi_servicio[10189]: 2025-08-09T22:22:59-04:00 - Servicio en ejecución
ago 09 22:23:29 192.168.46.151 mi_servicio[10220]: 2025-08-09T22:23:29-04:00 - Servicio en ejecución
ago 09 22:24:09 192.168.46.151 mi_servicio[10260]: 2025-08-09T22:24:09-04:00 - Servicio en ejecución
ago 09 22:24:29 192.168.46.151 mi_servicio[10280]: 2025-08-09T22:24:29-04:00 - Servicio en ejecución
ago 09 22:24:39 192.168.46.151 mi_servicio[10291]: 2025-08-09T22:24:39-04:00 - Servicio en ejecución
ago 09 22:24:54 192.168.46.151 mi_servicio[10306]: 2025-08-09T22:24:54-04:00 - Servicio en ejecución
ago 09 22:25:14 192.168.46.151 mi_servicio[10326]: 2025-08-09T22:25:14-04:00 - Servicio en ejecución
ago 09 22:25:29 192.168.46.151 mi_servicio[10341]: 2025-08-09T22:25:29-04:00 - Servicio en ejecución
ago 09 22:26:35 192.168.46.151 mi_servicio[10410]: 2025-08-09T22:26:35-04:00 - Servicio en ejecución
ago 09 22:26:50 192.168.46.151 mi_servicio[10429]: 2025-08-09T22:26:50-04:00 - Servicio en ejecución
ago 09 22:26:55 192.168.46.151 mi_servicio[10434]: 2025-08-09T22:26:55-04:00 - Servicio en ejecución
ago 09 22:27:00 192.168.46.151 mi_servicio[10439]: 2025-08-09T22:27:00-04:00 - Servicio en ejecución
ago 09 22:27:05 192.168.46.151 mi_servicio[10444]: 2025-08-09T22:27:05-04:00 - Servicio en ejecución
ago 09 22:27:15 192.168.46.151 mi_servicio[10455]: 2025-08-09T22:27:15-04:00 - Servicio en ejecución
ago 09 22:27:20 192.168.46.151 mi_servicio[10461]: 2025-08-09T22:27:20-04:00 - Servicio en ejecución
ago 09 22:27:25 192.168.46.151 mi_servicio[10466]: 2025-08-09T22:27:25-04:00 - Servicio en ejecución
ago 09 22:27:40 192.168.46.151 mi_servicio[10486]: 2025-08-09T22:27:40-04:00 - Servicio en ejecución
root@192:/home/maltamirano#
```

Se verificó el correcto funcionamiento del servicio mediante **systemctl status** y **journalctl**, confirmando que el proceso se mantiene activo y ejecutando el script configurado. Los registros muestran la salida periódica "Servicio en ejecución" en intervalos establecidos, validando que el servicio opera dentro de los parámetros definidos y registra la actividad en el log del sistema.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemd-analyze critical-chain mi_servicio.service
The time when unit became active or started is printed after the "@" character.
The time the unit took to start is printed after the "+" character.

mi_servicio.service @2h 58min 43.856s
└─network.target @3.926s
   └─wpa_supplicant.service @10.216s +57ms
      └─basic.target @2.638s
         └─dbus-broker.service @2.515s +95ms
            └─dbus.socket @2.500s
               └─sysinit.target @2.483s
                  └─systemd-vconsole-setup.service @3.492s +196ms
                     └─systemd-journald.socket
                        └─system.slice
                           └─.slice

root@192:/home/maltamirano# systemd-analyze blame | grep mi_servicio.service || true
root@192:/home/maltamirano#
```

Se ejecutó **systemd-analyze critical-chain** para visualizar la secuencia de arranque y confirmar que el servicio **mi_servicio.service** se inicia correctamente después de las dependencias de red. El tiempo de activación registrado asegura que el servicio entra en funcionamiento en el momento previsto. Asimismo, mediante **systemd-analyze blame** se verificó su inclusión en el análisis de tiempos de arranque, validando que está correctamente integrado al inicio del sistema.

```
root@192:/home/maltamirano - sudo su
root@192:/home/maltamirano# systemctl show -p MemoryMax,CPUQuota mi_servicio.service
MemoryMax=104857600
root@192:/home/maltamirano# sudo systemctl set-property mi_servicio.service MemoryMax=50M
root@192:/home/maltamirano# sudo systemctl set-property mi_servicio.service CPUQuota=10%
root@192:/home/maltamirano# systemctl show -p MemoryMax,CPUQuota mi_servicio.service
MemoryMax=52428800
root@192:/home/maltamirano# sudo systemctl cat mi_servicio.service
# Warning: mi_servicio.service changed on disk, the version systemd has loaded is outdated.
# This output shows the current version of the unit's original fragment and drop-in files.
# If fragments or drop-ins were added or removed, they are not properly reflected in this output.
# Run 'systemctl daemon-reload' to reload units.
# /etc/systemd/system/mi_servicio.service
[Unit]
Description=Servicio demo con límites y hardening
After=network.target

[Service]
Type=simple
User=svcmi
Group=svcmi
WorkingDirectory=/opt/mi_servicio
ExecStart=/opt/mi_servicio/mi_servicio.sh
Restart=always
RestartSec=2

# Límites de recursos
CPUQuota=20%
MemoryMax=100M

# Seguridad y aislamiento
NoNewPrivileges=yes
CapabilityBoundingSet=CAP_SYS_ADMIN CAP_SYS_MODULE CAP_SYS_RAWIO CAP_NET_ADMIN
PrivateTmp=yes
ProtectSystem=strict
ProtectHome=yes
ReadOnlyPaths=/etc /usr /var /log
lines 1-29

# Warning: mi_servicio.service changed on disk, the version systemd has loaded is outdated.
# This output shows the current version of the unit's original fragment and drop-in files.
# If fragments or drop-ins were added or removed, they are not properly reflected in this output.
# Run 'systemctl daemon-reload' to reload units.
# /etc/systemd/system/mi_servicio.service
[Unit]
Description=Servicio demo con límites y hardening
After=network.target

[Service]
Type=simple
User=svcmi
Group=svcmi
WorkingDirectory=/opt/mi_servicio
ExecStart=/opt/mi_servicio/mi_servicio.sh
Restart=always
RestartSec=2

# Límites de recursos
CPUQuota=20%
MemoryMax=100M

# Seguridad y aislamiento
NoNewPrivileges=yes
CapabilityBoundingSet=CAP_SYS_ADMIN CAP_SYS_MODULE CAP_SYS_RAWIO CAP_NET_ADMIN
PrivateTmp=yes
ProtectSystem=strict
ProtectHome=yes
ReadOnlyPaths=/etc /usr /var /log
lines 1-29
```

En la imagen se aprecia la configuración de endurecimiento (hardening) y limitación de recursos aplicada al servicio **mi_servicio.service**. Se establecen límites de consumo con **MemoryMax=50M** y **CPUQuota=10%**, además de medidas de seguridad como **NoNewPrivileges**, **PrivateTmp**, **ProtectSystem=strict** y **ProtectHome=yes**. También se restringen rutas críticas en modo solo lectura y se definen capacidades específicas mediante **CapabilityBoundingSet**. Esta configuración asegura que el servicio opere con recursos controlados y bajo un entorno aislado, reduciendo la superficie de ataque y mejorando la estabilidad del sistema.

```
root@192:/home/maltamirano - sudo su
# Límites de recursos
CPUQuota=20%
MemoryMax=100M

# Seguridad y aislamiento
NoNewPrivileges=yes
CapabilityBoundingSet=~CAP_SYS_ADMIN CAP_SYS_MODULE CAP_SYS_RAWIO CAP_NET_ADMIN
PrivateTmp=yes
ProtectSystem=strict
ProtectHome=yes
ReadOnlyPaths=/etc /usr /var/log
ReadWritePaths=/var/log /opt/mi_servicio

[Install]
WantedBy=multi-user.target

# /etc/systemd/system.control/mi_servicio.service.d/50-MemoryMax.conf
# This is a drop-in unit file extension, created via "systemctl set-property"
# or an equivalent operation. Do not edit.
[Service]
MemoryMax=52428800

# /etc/systemd/system.control/mi_servicio.service.d/50-CPUQuota.conf
# This is a drop-in unit file extension, created via "systemctl set-property"
# or an equivalent operation. Do not edit.
[Service]
CPUQuota=10.00%

root@192:/home/maltamirano#
```

En esta imagen se muestra la configuración final del archivo de unidad **mi_servicio.service** con parámetros de seguridad, aislamiento y control de recursos. Se establecen límites como **CPUQuota=20%** y **MemoryMax=100M**, además de restricciones de privilegios (**NoNewPrivileges=yes**) y capacidades (**CapabilityBoundingSet** con permisos específicos). También se habilita el uso de directorios temporales privados, protección estricta del sistema y restricciones de escritura y lectura en rutas concretas. Al final, se observan los archivos de extensión **drop-in** generados automáticamente por **systemctl set-property**, que registran los valores de **MemoryMax** y **CPUQuota** aplicados al servicio.