EBENEZER IT · GUÍAS PASO A PASO

GUÍA 11 · EPISODIOS 25 Y 26

Defensa: CrowdSec y AppArmor

Dos capas más para tu servidor. CrowdSec lee tus registros, reconoce ataques y los bloquea en el cortafuegos, y además te trae la lista de miles de atacantes que han visto otros servidores. AppArmor encierra a cada programa en lo que necesita: si alguien toma el control de uno, no puede salir de su jaula.

~2 horas
más una semana de observación

Requisito
Guías 01 y 02; mejor con 08 y 10

Ubuntu 24.04 LTS
CrowdSec · nftables · AppArmor

www.ebenezer.org.es · Alejandro Quevedo · Versión pública 1.0 · noviembre 2026

Índice

Antes de empezar

0 Qué vas a construir y valores de esta guía

Parte A · CrowdSec: detectar y bloquear

Paso 01 Cómo funciona CrowdSec

Paso 02 Instalar CrowdSec

Paso 03 El bouncer: el que bloquea

Paso 04 Probar un bloqueo

Parte B · Ajustar CrowdSec a tu servidor

Paso 05 Qué registros lee

Paso 06 Las colecciones de tus servicios

Paso 07 El correo, una sola vez

Paso 08 Listas blancas

Paso 09 La consola web (opcional)

Parte C · AppArmor: cada programa en su jaula

Paso 10 Cómo funciona AppArmor

Paso 11 Diagnosticar un perfil

Paso 12 Reglas locales

Paso 13 Observar y pasar a enforce

Paso 14 Comprobación final e instantánea

Anexos

A Problemas frecuentes

B CrowdSec y fail2ban: quién hace qué

C Chuleta de cscli y AppArmor

Cómo usar esta guía

Tienes aquí **todos los pasos y comandos**, en orden. La explicación de cada paso (el porqué de cada línea) está en el **vídeo del episodio**.

Para copiar los comandos sin errores, usa la versión web: **https://www.ebenezer.org.es/guia-11-defensa.html**. Al copiar desde un PDF se pueden romper los saltos de línea y las comillas.

ANTES DE EMPEZAR

Qué vas a construir

Tu servidor ya tiene cortafuegos (UFW y el de IONOS) y fail2ban. Esta guía añade dos capas que trabajan de otra manera:

Capa	Qué hace	Contra qué
CrowdSec	Lee los registros de SSH, Nginx, el correo… reconoce patrones de ataque y bloquea la IP en el cortafuegos. Además descarga la lista de IPs que otros servidores han visto atacar	Fuerza bruta, escáneres, intentos de explotar fallos web conocidos
AppArmor	Limita qué archivos, programas y permisos puede usar cada servicio, aunque se ejecute como root	Que un fallo en un programa le dé al atacante el resto del servidor

No sustituye a fail2ban

Los dos pueden convivir: CrowdSec cubre los servicios comunes y trae la inteligencia de la comunidad; fail2ban sigue con las jaulas específicas que ya tienes (ejabberd, Roundcube). El Anexo B lo explica.

Valores de esta guía

Valor	Qué es
`TU_DOMINIO`	Tu dominio
`TU_IP_CASA`	La IP pública de tu casa u oficina, **solo si es fija** (paso 8)

Recordatorio

- Haz una instantánea en IONOS antes de empezar.
- Ten abierta una segunda sesión SSH mientras tocas el cortafuegos: si algo te bloquea, la que ya está abierta sigue funcionando.
- La parte C necesita una semana de observación entre el paso 12 y el 13: no la hagas con prisa.

PARTE A

CrowdSec: detectar y bloquear

Instalamos el motor que detecta y el bouncer que bloquea, y probamos un bloqueo de verdad. Corresponde al **episodio 25**.

Paso 1 Episodio 25

Cómo funciona CrowdSec

Por qué: CrowdSec separa el trabajo en piezas: una que **detecta** y otra que **bloquea**.

Pieza	Qué hace
Adquisición	Qué registros lee: `/var/log/auth.log`, `/var/log/nginx/*.log`, `/var/log/mail.log`…
Parsers	Entienden cada línea: «fallo de contraseña SSH desde 1.2.3.4»
Escenarios	Reglas de comportamiento: «más de 5 fallos de SSH en poco tiempo desde la misma IP es fuerza bruta»
Colecciones	Paquetes de parsers y escenarios para un servicio: `crowdsecurity/sshd`, `crowdsecurity/nginx`…
Decisiones	El resultado: «bloquear 1.2.3.4 durante 4 horas»
LAPI (API local)	Guarda las decisiones en tu servidor
Bouncer	Lee las decisiones de la LAPI y **bloquea**: el de nftables mete las IPs en el cortafuegos del núcleo
CAPI (API central)	Comparte tus detecciones (solo la IP atacante y el tipo de ataque) y te trae la lista de la comunidad

¿Qué se envía a CrowdSec?
La IP que te atacó, cuándo y qué escenario saltó. No se envían tus registros ni datos de tus usuarios. A cambio recibes la lista de IPs que la comunidad ha visto atacar: en un servidor pequeño suelen ser más de 10 000.

Comprueba antes de seguir
Sabes explicar la diferencia entre el motor (detecta) y el bouncer (bloquea), y qué es una colección.

Paso 2 Episodio 25

Instalar CrowdSec

Por qué: desde el repositorio oficial de CrowdSec: el de Ubuntu trae una versión muy antigua.

EN EL SERVIDOR

```
curl -fsSL https://install.crowdsec.net -o crowdsec-repo.sh
less crowdsec-repo.sh
sudo sh crowdsec-repo.sh && rm crowdsec-repo.sh
sudo apt update
sudo apt install crowdsec
sudo systemctl is-active crowdsec
sudo cscli version | head -n 3
```

Comprueba antes de seguir

`crowdsec` está `active` y `cscli version` responde.

Paso 3 Episodio 25

El bouncer: el que bloquea

Por qué: sin bouncer, CrowdSec detecta pero no bloquea.

EN EL SERVIDOR

```
sudo apt install crowdsec-firewall-bouncer-nftables
sudo systemctl is-active crowdsec-firewall-bouncer
sudo cscli bouncers list
```

SALIDA ESPERADA

```
Name                          IP Address  Valid  Last API pull          Type
cs-firewall-bouncer-XXXX      127.0.0.1   ✔️     2026-11-06T09:17:34Z   crowdsec-firewall-bouncer
```

Si el bouncer no aparece o su servicio dice *bouncer stream halted*, el registro automático falló. Se registra a mano así:

EN EL SERVIDOR

```
API_KEY=$(sudo cscli bouncers add cs-firewall-bouncer -o raw)
sudo sed -i "s|\${API_KEY}|${API_KEY}|" /etc/crowdsec/bouncers/crowdsec-firewall-bouncer.yaml
unset API_KEY
sudo systemctl restart crowdsec-firewall-bouncer
```

Comprueba antes de seguir

`cscli bouncers list` muestra el bouncer con ✔️ y una hora de hace unos segundos.

Paso 4 Episodio 25

Probar un bloqueo

Por qué: antes de fiarnos, comprobamos que una decisión llega de verdad al cortafuegos.

EN EL SERVIDOR

```
sudo cscli decisions add --ip 203.0.113.50 --duration 5m --reason "prueba de la guia"
sleep 15
sudo nft list ruleset | grep -c "203.0.113.50"
sudo cscli decisions delete --ip 203.0.113.50
sleep 15
sudo nft list ruleset | grep -c "203.0.113.50"
```

Comprueba antes de seguir

La IP de prueba aparece en nftables (1) y desaparece al borrar la decisión (0): CrowdSec bloquea de verdad.

PARTE B

Ajustar CrowdSec a tu servidor

La instalación automática acierta en casi todo, pero conviene revisar qué lee, quitar lo que sobra, que no cuente nada dos veces y que nunca te bloquee a ti. Corresponde al **episodio 25**.

Paso 5 Episodio 25

Qué registros lee

Por qué: CrowdSec solo protege lo que lee.

EN EL SERVIDOR

```
sudo ls /etc/crowdsec/acquis.d/
sudo cscli metrics show acquisition
```

Columna	Qué significa
Lines read	Líneas leídas del archivo
Lines parsed	Las que algún parser entendió. Si un archivo tiene todo en *unparsed*, ningún parser lo entiende: o falta su colección o el formato no es el esperado
Lines poured to bucket	Las que alimentaron algún escenario (posibles ataques)
Lines whitelisted	Las descartadas por una lista blanca (paso 8)

Es normal que `auth.log` o `syslog` tengan muchas líneas sin entender: contienen de todo (sudo, cron, el núcleo) y CrowdSec solo busca lo que le interesa. Lo que **no** es normal es que falte el registro de un servicio que tienes, o que un archivo de un servicio (por ejemplo `nginx/access.log`) tenga todo sin entender.

Comprueba antes de seguir

Aparecen `auth.log`, `mail.log` (si tienes correo) y los `access.log` y `error.log` de Nginx, con líneas entendidas.

Paso 6 Episodio 25

Las colecciones de tus servicios

Por qué: una colección por cada servicio que tienes, y ninguna de los que no.

EN EL SERVIDOR

```
sudo cscli collections list
```

Colección	Para qué	¿La quieres?
`crowdsecurity/linux` y `sshd`	Base del sistema y SSH	Siempre

Colección	Para qué	¿La quieres?
`crowdsecurity/nginx`, `base-http-scenarios`, `http-cve`	Escáneres web, fuerza bruta en formularios, fallos conocidos	Si tienes Nginx (Guía 02)
`crowdsecurity/postfix` y `dovecot`	Fuerza bruta y abusos del correo	Si tienes correo (Guía 08)
`crowdsecurity/wordpress`	Ataques típicos contra WordPress	Si tienes WordPress (Guía 05)
`crowdsecurity/whitelist-good-actors`	Nunca bloquear a buscadores y servicios conocidos (Google, Bing…)	Sí, si tienes web
`crowdsecurity/apache2`	Para Apache	No: usamos Nginx

EN EL SERVIDOR

```
sudo cscli collections install crowdsecurity/nginx crowdsecurity/base-http-scenarios
crowdsecurity/http-cve
sudo cscli collections install crowdsecurity/postfix crowdsecurity/dovecot
crowdsecurity/whitelist-good-actors
sudo cscli collections remove crowdsecurity/apache2
sudo systemctl reload crowdsec
sudo cscli collections list
```

Comprueba antes de seguir

En la lista están las colecciones de tus servicios y ninguna de los que no tienes.

Paso 7 Episodio 25

El correo, una sola vez

Por qué: Ubuntu escribe cada línea del correo en **dos** archivos: `/var/log/mail.log` y `/var/log/syslog`.

EN EL SERVIDOR

```
grep -n "^\*\.\*;auth,authpriv.none" /etc/rsyslog.d/50-default.conf
```

SALIDA ESPERADA

```
9:*.*;auth,authpriv.none        -/var/log/syslog
```

Si sale esa línea (sin `mail.none`), la cambiamos:

EN EL SERVIDOR

```
sudo cp /etc/rsyslog.d/50-default.conf /etc/rsyslog.d/50-default.conf.orig
sudo sed -i '/^\*\.\*;auth,authpriv\.none/ s/auth,authpriv\.none/auth,authpriv.none;mail.none/'
/etc/rsyslog.d/50-default.conf
grep -n "^\*\.\*;auth" /etc/rsyslog.d/50-default.conf
sudo systemctl restart rsyslog
logger -p mail.info "prueba-correo-una-vez"
sleep 2
grep -c "prueba-correo-una-vez" /var/log/mail.log /var/log/syslog
```

SALIDA ESPERADA

```
/var/log/mail.log:1
/var/log/syslog:0
```

Dovecot debe escribir en mail.log

En la Guía 08, Dovecot envía su registro al sistema y acaba en `mail.log`, donde lo leen CrowdSec y fail2ban. Si alguna vez configuras `log_path` en Dovecot para que escriba en su propio archivo, CrowdSec dejará de verlo.

Comprueba antes de seguir

La prueba aparece solo en `mail.log`, y unas horas después `cscli metrics show acquisition` ya no muestra las mismas cifras para `mail.log` y `syslog`.

Paso 8 Episodio 25

Listas blancas

Por qué: lo peor que puede hacer CrowdSec es bloquearte a ti.

EN EL SERVIDOR

```
sudo cat /etc/crowdsec/parsers/s02-enrich/whitelists.yaml
```

La lista que trae CrowdSec ya excluye `127.0.0.0/8`, `10.0.0.0/8`, `172.16.0.0/12` y `192.168.0.0/16`: la propia máquina, Docker y **tu VPN** (Guía 06). Si entras por SSH a través de la VPN, ya nunca te bloqueará.

Si además tienes en casa o en la oficina una **IP pública fija** desde la que administras, añádela. Si tu IP cambia (lo normal en casa), **no** lo hagas: protegerías la IP de otra persona el día que cambie. Usa la VPN.

EN EL SERVIDOR

```
F=/etc/crowdsec/parsers/s02-enrich/mis-redes.yaml
[ -e "$F" ] && echo "YA EXISTE, no lo toco" || sudo tee "$F" > /dev/null <<'EOF'
name: ebenezer/mis-redes
description: "IPs desde las que administro: nunca bloquear"
whitelist:
  reason: "administracion"
  ip:
    - "TU_IP_CASA"
EOF
sudo systemctl reload crowdsec
sudo cscli parsers list | grep mis-redes
```

Comprueba antes de seguir

Entiendes qué redes ya están protegidas y, si añadiste tu IP fija, `mis-redes` aparece en la lista de parsers.

Paso 9 Episodio 25

La consola web (opcional)

Por qué: `app.crowdsec.net` muestra en una web las alertas de tu servidor, de dónde vienen y qué escenarios saltan más.

1. Crea una cuenta en `app.crowdsec.net` y, en **Security Engines**, copia la orden de enrolamiento que te da.
2. En el servidor, escríbela con `read` para que la clave no quede en el historial:

```
EN EL SERVIDOR
```

```
read -rs -p "Clave de enrolamiento: " K; echo
sudo cscli console enroll "$K"
unset K
sudo systemctl restart crowdsec
```

1. Vuelve a la web y **acepta** el servidor que aparece como pendiente.

Comprueba antes de seguir

`sudo cscli console status` dice `Enrolled` ✅.

PARTE C

AppArmor: cada programa en su jaula

AppArmor ya está activo en Ubuntu. Aprendemos a leerlo, a ajustar un perfil que se queda corto y a ponerlo a bloquear sin romper el servicio. Usamos de ejemplo el de ejabberd (Guía 10), pero el método vale para cualquiera. Corresponde al **episodio 26**.

Paso 10 Episodio 26

Cómo funciona AppArmor

Por qué: un perfil es una lista de lo que un programa **puede** hacer: qué archivos lee o escribe, qué programas ejecuta, qué permisos especiales usa.

EN EL SERVIDOR

```
sudo apt install apparmor-utils
sudo aa-status | grep -E "profiles are|processes are"
```

Modo	Qué hace
enforce	Aplica el perfil: lo que no está permitido se **bloquea** y queda anotado como `DENIED`
complain	Solo **anota** lo que se saldría del perfil (`ALLOWED`), pero lo deja hacer. Sirve para ajustar un perfil antes de aplicarlo
unconfined	Perfil cargado que no restringe nada. Ubuntu 24.04 trae muchos así: es normal

Dónde están los perfiles

En `/etc/apparmor.d/`, un archivo por programa, con el nombre de su ruta: el de `/usr/sbin/ejabberdctl` es `usr.sbin.ejabberdctl`. Los trae cada paquete y se sobrescriben al actualizar, así que **no se editan**: los cambios propios van en `/etc/apparmor.d/local/` (paso 12).

Comprueba antes de seguir

Sabes qué perfiles están en `enforce` y cuáles en `complain`. El de ejabberd viene en `complain`.

Paso 11 Episodio 26

Diagnosticar un perfil

Por qué: antes de pasar un perfil a `enforce`, hay que saber qué se habría bloqueado.

EN EL SERVIDOR

```
sudo journalctl -k --since "7 days ago" | grep -c 'apparmor="ALLOWED"'
sudo journalctl -k --since "7 days ago" | grep 'apparmor="ALLOWED"' | grep -o 'profile="[^"]*"'
| sort | uniq -c | sort -rn | head
sudo journalctl -k --since "7 days ago" | grep 'apparmor="ALLOWED"' | grep
'profile="/usr/sbin/ejabberdctl"' \
  | grep -o 'operation="[^"]*"\|name="[^"]*"' | paste - - | sed -E 's/[0-9]+/N/g' | sort | uniq
-c | sort -rn | head -n 20
```

Con ejabberd instalado como en la Guía 10, lo habitual es ver esto:

Avisos	Operación y recurso	Qué es
Miles	`sendmsg` `/run/systemd/notify`	Cada pocos segundos ejabberd le dice a systemd «sigo vivo». El perfil del paquete no lo contempla
Unos pocos	`exec /usr/bin/dirname`	El script `ejabberdctl` lo usa al arrancar
Unos pocos	`capable sys_resource`	Subir sus propios límites (archivos abiertos)

Si las subidas o el certificado están en otra carpeta

El perfil espera los datos en `/var/lib/ejabberd` y la configuración en `/etc/ejabberd`. Si guardas las subidas en otro sitio (por ejemplo `/var/www/upload`), aparecerán avisos de `open`, `mknod` y `mkdir` sobre esa ruta: también hay que permitirla.

Comprueba antes de seguir

Tienes la lista de lo que el perfil no permite, agrupada y con su explicación.

Paso 12 Episodio 26

Reglas locales

Por qué: las reglas que faltan se añaden en el archivo local del perfil, que el paquete no toca al actualizarse.

EN EL SERVIDOR

```
grep -n "local/usr.sbin.ejabberdctl" /etc/apparmor.d/usr.sbin.ejabberdctl
sudo cat /etc/apparmor.d/local/usr.sbin.ejabberdctl 2>/dev/null; echo "---"
```

Debe salir una línea `#include <local/usr.sbin.ejabberdctl>` justo antes de la `}` que cierra el perfil principal. Entonces añadimos las reglas:

EN EL SERVIDOR

```
F=/etc/apparmor.d/local/usr.sbin.ejabberdctl
grep -q "Ebenezer IT" "$F" 2>/dev/null && echo "YA ESTÁN, no las repito" || sudo tee -a "$F" >
/dev/null <<'EOF'
# Ebenezer IT: lo que ejabberd necesita y el perfil del paquete no trae
/run/systemd/notify w,
/usr/bin/dirname rix,
capability sys_resource,
EOF
sudo cat "$F"
sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.ejabberdctl
```

Letra	Permiso
`r` / `w`	Leer / escribir (escribir incluye crear y borrar)
`k`	Bloquear el archivo mientras se usa (*lock*)

Letra	Permiso
`ix`	Ejecutar con el mismo perfil
`*` / `**`	Cualquier nombre en esa carpeta / cualquier cosa debajo, a cualquier profundidad

Si tus subidas están en otra carpeta (paso 11), añade también, por ejemplo, `/var/www/upload/ r,` y `/var/www/upload/** rwk,`.

Unos minutos después, el contador debe quedarse a cero:

EN EL SERVIDOR

```
sudo journalctl -k --since "5 min ago" | grep 'apparmor="ALLOWED"' | grep -c ejabberdctl
```

Comprueba antes de seguir

En los últimos 5 minutos hay **0** avisos del perfil de ejabberd.

Paso 13 Episodio 26

Observar y pasar a enforce

Por qué: que no haya avisos en 5 minutos no basta: algunas cosas pasan una vez al día o a la semana (la renovación del certificado, la copia nocturna, la limpieza de mensajes del domingo).

EN EL SERVIDOR

```
sudo journalctl -k --since "7 days ago" | grep 'apparmor="ALLOWED"' | grep ejabberdctl \
  | grep -o 'operation="[^"]*"\|name="[^"]*"' | paste - - | sort | uniq -c | sort -rn
```

Si sale algo nuevo, vuelve al paso 12 y añade esa regla. Cuando la semana salga **vacía**:

EN EL SERVIDOR

```
sudo aa-enforce /usr/sbin/ejabberdctl
sudo aa-status | grep -A3 "enforce mode" | head -n 4
sudo systemctl restart ejabberd
sleep 15
sudo ejabberdctl status
```

Prueba todo lo del chat: un mensaje, una foto, una llamada, y que `ejabberdctl` responda. Luego, busca bloqueos:

EN EL SERVIDOR

```
sudo journalctl -k --since "30 min ago" | grep 'apparmor="DENIED"' | grep ejabberdctl
```

Si algo deja de funcionar

Vuelve a `complain` con `sudo aa-complain /usr/sbin/ejabberdctl`, busca el `DENIED` con el comando de arriba, añade la regla en el paso 12 y repite la semana de observación.

Comprueba antes de seguir

El perfil está en `enforce`, el chat funciona entero y no hay ningún `DENIED` de ejabberd.

Paso 14 Episodio 26

Comprobación final e instantánea

Por qué: un repaso de las dos capas.

EN EL SERVIDOR

```
sudo systemctl is-active crowdsec crowdsec-firewall-bouncer
sudo cscli bouncers list
sudo cscli metrics show decisions
sudo aa-status | grep -E "profiles are in (enforce|complain) mode"
```

Comprobación	Resultado correcto
Servicios	`active` y `active`
Bouncer	✔️ con una hora reciente
Decisiones	Las de `CAPI` (la comunidad) son miles; las de `crowdsec` son las tuyas
AppArmor	El perfil de ejabberd, en `enforce` (tras la semana del paso 13)

Al chequeo diario

Añade `crowdsec crowdsec-firewall-bouncer` a la lista `SERVICES` de tu script de chequeo. Si el bouncer se para, CrowdSec seguiría detectando pero nadie bloquearía.

Después, en IONOS, la instantánea `11-defensa`.

Comprueba antes de seguir

Tu servidor reconoce y bloquea ataques solo, recibe la lista de atacantes de la comunidad, y sus servicios están encerrados en lo que necesitan.

ANEXOS

Para cuando algo no sale

Los registros que lo cuentan todo: `/var/log/crowdsec.log`, `/var/log/crowdsec-firewall-bouncer.log` y, para AppArmor, `sudo journalctl -k | grep apparmor`.

Anexo A · Problemas frecuentes

Lo que ves	Qué pasa	Qué hacer
`cscli bouncers list` vacío o sin ✔️	El bouncer no se registró	Registro manual del paso 3
Te has bloqueado a ti mismo	Tu IP saltó un escenario (muchos fallos de contraseña, por ejemplo)	Desde la consola web de IONOS o por la VPN: `sudo cscli decisions list` y `sudo cscli decisions delete --ip TU_IP`. Después, paso 8
Un servicio no aparece en `metrics show acquisition`	Falta su archivo en `acquis.d` o escribe en otro sitio	Crea un archivo en `/etc/crowdsec/acquis.d/` con `filenames:` (la ruta del registro) y `labels: type:` (normalmente `syslog`), igual que los que ya hay, y recarga CrowdSec
Las mismas cifras en `mail.log` y `syslog`	El correo se escribe en los dos	Paso 7
Un servicio deja de funcionar tras `aa-enforce`	Le falta una regla al perfil	`aa-complain`, busca el `DENIED` y vuelve al paso 12
`aa-enforce: command not found`	Falta `apparmor-utils`	`sudo apt install apparmor-utils`

Anexo B · CrowdSec y fail2ban: quién hace qué

Los dos leen registros y bloquean IPs. No se estorban (cada uno usa su propia tabla en el cortafuegos), pero conviene saber qué hace cada uno:

	CrowdSec	fail2ban
Detecta	Con colecciones mantenidas por la comunidad	Con filtros que configuras tú (jaulas)
Inteligencia compartida	Sí: bloquea IPs que atacaron a otros antes de que te ataquen a ti	No
Servicios cubiertos aquí	SSH, Nginx, WordPress, Postfix, Dovecot	Los mismos, más ejabberd y Roundcube

Mantener los dos tiene sentido: CrowdSec aporta la lista de la comunidad y los escenarios web; fail2ban, las jaulas de servicios que CrowdSec no conoce. Si algún día quieres quedarte solo con uno, apaga antes las jaulas de fail2ban que CrowdSec ya cubre.

Anexo C · Chuleta de cscli y AppArmor

Para...	Cómo
Ver bloqueos activos (los tuyos)	`sudo cscli decisions list`
Contar bloqueos por origen	`sudo cscli metrics show decisions`
Ver las alertas recientes	`sudo cscli alerts list | head -n 20`
Desbloquear una IP	`sudo cscli decisions delete --ip 1.2.3.4`
Bloquear una IP a mano	`sudo cscli decisions add --ip 1.2.3.4 --duration 24h --reason "..."`
Qué haría CrowdSec con un registro	`sudo cscli explain --file /var/log/auth.log --type syslog | tail`
Actualizar colecciones y escenarios	`sudo cscli hub update && sudo cscli hub upgrade && sudo systemctl reload crowdsec`
Resumen de AppArmor	`sudo aa-status`
Pasar un perfil a observar / a bloquear	`sudo aa-complain RUTA` · `sudo aa-enforce RUTA`
Recargar un perfil	`sudo apparmor_parser -r /etc/apparmor.d/PERFIL`
Ver bloqueos de AppArmor	`sudo journalctl -k --since today | grep 'apparmor="DENIED"'`