Mini Proyecto Guiado Paso a Paso: Asegurar un Servidor SSH con fail2ban

Autor: Santiago Abia
Fecha de Creación: 1 de Diciembre de 2025
Unidad Temática (UT): 3

Introducción y Objetivo del Proyecto

Situación Laboral Realista

Como administrador de sistemas, se ha desplegado un nuevo servidor Linux en Internet. Al revisar los registros de autenticación (`/var/log/auth.log`), se observa un flujo constante de intentos de inicio de sesión fallidos desde direcciones IP desconocidas. Esta actividad es característica de un **ataque de fuerza bruta automatizado**. La tarea crítica es instalar y configurar la herramienta `fail2ban` para bloquear automáticamente estas direcciones IP maliciosas y proteger el servidor.

Objetivo del Proyecto

El objetivo de este mini proyecto es **instalar y configurar `fail2ban`** en un servidor basado en Debian/Ubuntu para proteger el servicio SSH, bloqueando temporalmente las IPs que realicen múltiples intentos fallidos de autenticación.

Pasos para la Implementación de fail2ban

Paso 1: Instalación de fail2ban

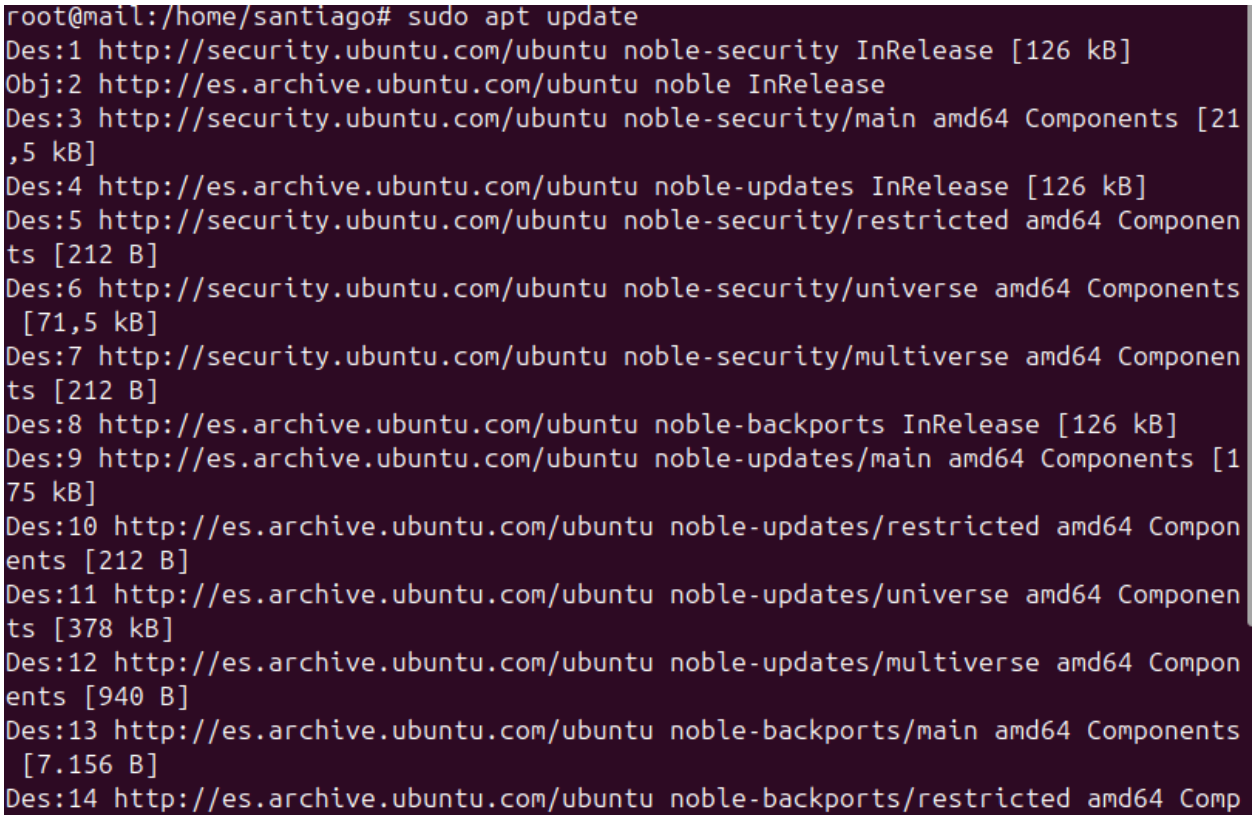
`fail2ban` es un software que monitoriza los archivos de log del sistema y actualiza dinámicamente las reglas del *firewall* (cortafuegos) para bloquear direcciones IP con actividad sospechosa.

Para instalar la herramienta, ejecute los siguientes comandos en la terminal del servidor:

```bash

# 1. Actualizar la lista de paquetes del sistema

sudo apt update
```



2. Instalar fail2ban (la opción -y acepta automáticamente la instalación)

sudo apt install fail2ban -y

```
root@mail:/home/santiago# sudo apt install fail2ban -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
fail2ban ya está en su versión más reciente (1.0.2-3ubuntu0.1).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
```

3. Verificar que el servicio esté en funcionamiento

sudo systemctl status fail2ban ```

```
root@mail:/home/santiago# sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enable>
     Active: active (running) since Mon 2025-12-01 13:11:17 CET; 2min 40s ago
       Docs: man:fail2ban(1)
   Main PID: 1346 (fail2ban-server)
      Tasks: 5 (limit: 4600)
     Memory: 23.9M (peak: 24.6M)
        CPU: 389ms
     CGroup: /system.slice/fail2ban.service
             └─1346 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

dic 01 13:11:17 mail.mi-startup.lan systemd[1]: Started fail2ban.service - Fail2Ban Se>
dic 01 13:11:17 mail.mi-startup.lan fail2ban-server[1346]: 2025-12-01 13:11:17,142 fai>
dic 01 13:11:17 mail.mi-startup.lan fail2ban-server[1346]: Server ready
lines 1-14/14 (END)
```

Nota: Una vez instalado, `fail2ban` ya ofrece protección básica para SSH con su configuración por defecto, pero se recomienda encarecidamente personalizarla para adaptarla a las necesidades específicas de seguridad.

Paso 2: Crear un Archivo de Configuración Local (`jail.local`)

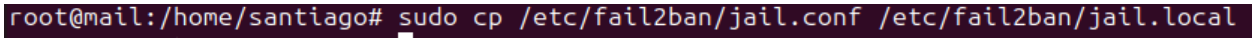
Se considera una **mejor práctica** no modificar el archivo de configuración principal (`/etc/fail2ban/jail.conf`). En su lugar, se debe crear una copia local llamada `jail.local` para personalizar la configuración. Esto asegura que las actualizaciones futuras del paquete `fail2ban` no sobrescriban las configuraciones personalizadas.

Ejecute los siguientes comandos para crear y abrir el archivo de configuración local:

```bash

# 1. Copiar jail.conf como plantilla local

sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```




```

# 2. Editar la configuración local con el editor de texto nano

sudo nano /etc/fail2ban/jail.local
```

Paso 3: Configurar la "Jaula" (Jail) para SSH

Dentro del archivo `/etc/fail2ban/jail.local`, busque la sección correspondiente a SSH, identificada como `[sshd]`, y ajuste los parámetros clave para definir el comportamiento de baneo deseado.

A continuación, se muestra la configuración recomendada para la sección `[sshd]`:

```ini [sshd] enabled = true # Activar la protección de SSH maxretry = 3   # Número de intentos fallidos permitidos antes del baneo bantime = 3600 # Tiempo de baneo en segundos (equivalente a 1 hora) findtime = 600 # Ventana de tiempo en segundos para contar los intentos (equivalente a 10 minutos) ```

```ini
[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
enabled = true # Activar la protección de SSH

maxretry = 3 # Número de intentos fallidos permitidos

bantime = 3600 # Tiempo de baneo en segundos (1 hora)

findtime = 600 # Ventana de tiempo en segundos (10 minutos)
```

Explicación de la Configuración:

Con esta configuración, si una dirección IP realiza **3 intentos fallidos** de autenticación dentro de una **ventana de 10 minutos** (`findtime`), quedará automáticamente **bloqueada durante una hora** (`bantime`).
```

Paso 4: Reiniciar fail2ban y Probar el Baneo

Para que los cambios realizados en el archivo `jail.local` surtan efecto, es necesario reiniciar el servicio `fail2ban`:

```bash
# Comando para aplicar los cambios y reiniciar el servicio
sudo systemctl restart fail2ban
```

```
root@mail:/home/santiago# sudo systemctl restart fail2ban
```

Prueba del Sistema:

Para verificar que la configuración funciona correctamente, intente simular un ataque de fuerza bruta desde otro equipo:

1. Desde un ordenador diferente, intente conectarse por SSH al servidor: ```bash ssh tu_usuario@<IP_DEL_SERVIDOR> ```
2. **Introduzca una contraseña incorrecta tres veces consecutivas.**

Tras el tercer intento fallido, la dirección IP de su equipo de prueba debería quedar bloqueada por el *firewall*, y cualquier intento de conexión posterior será rechazado o resultará en un *timeout*.

```
root@mail:/home/santiago# ssh santiago@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ED25519 key fingerprint is SHA256:NHkppLFb9q/97HfIvYbuyuhEaM+QPttw6FdRl983f0w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.15' (ED25519) to the list of known hosts.
santiago@10.0.2.15's password:
Permission denied, please try again.
santiago@10.0.2.15's password:
Permission denied, please try again.
santiago@10.0.2.15's password:
santiago@10.0.2.15: Permission denied (publickey,password).
root@mail:/home/santiago#
```

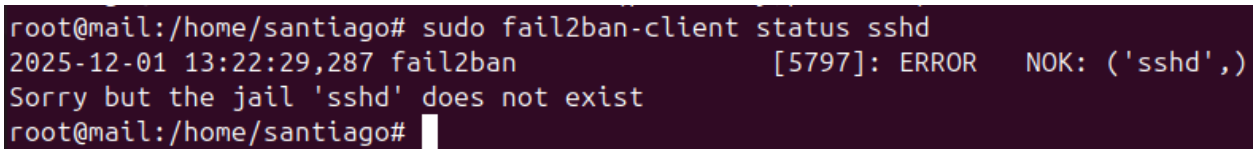
Paso 5: Verificar el Estado de los Baneos

Para comprobar el estado actual de las "jaulas" y las reglas del *firewall* aplicadas por `fail2ban`, utilice los siguientes comandos:

```bash

# 1. Ver el estado de la jaula 'sshd' y las IPs actualmente baneadas

sudo fail2ban-client status sshd


```

2. Comprobar las reglas de baneo que se han añadido al firewall (iptables)

sudo iptables -L -n ```

El primer comando (`fail2ban-client status sshd`) es la forma más directa de ver qué direcciones IP están actualmente bloqueadas por la jaula de SSH. El segundo comando (`iptables -L -n`) muestra la cadena del *firewall* con las reglas de baneo que `fail2ban` ha insertado.

```
root@mail:/home/santiago# sudo iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
ufw-before-logging-input  0    --  0.0.0.0/0            0.0.0.0/0
ufw-before-input  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-input  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-logging-input  0    --  0.0.0.0/0            0.0.0.0/0
ufw-reject-input  0    --  0.0.0.0/0            0.0.0.0/0
ufw-track-input  0    --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source               destination
ufw-before-logging-forward  0    --  0.0.0.0/0            0.0.0.0/0
ufw-before-forward  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-forward  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-logging-forward  0    --  0.0.0.0/0            0.0.0.0/0
ufw-reject-forward  0    --  0.0.0.0/0            0.0.0.0/0
ufw-track-forward  0    --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ufw-before-logging-output  0    --  0.0.0.0/0            0.0.0.0/0
ufw-before-output  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-output  0    --  0.0.0.0/0            0.0.0.0/0
ufw-after-logging-output  0    --  0.0.0.0/0            0.0.0.0/0
ufw-reject-output  0    --  0.0.0.0/0            0.0.0.0/0
ufw-track-output  0    --  0.0.0.0/0            0.0.0.0/0

Chain ufw-after-forward (1 references)
target     prot opt source               destination

Chain ufw-after-input (1 references)
target     prot opt source               destination
ufw-skip-to-policy-input  17   --  0.0.0.0/0            0.0.0.0/0           udp dpt:13
7
ufw-skip-to-policy-input  17   --  0.0.0.0/0            0.0.0.0/0           udp dpt:13
8
ufw-skip-to-policy-input  6    --  0.0.0.0/0            0.0.0.0/0           tcp dpt:13
```