

Cómo combatir los ataques de diccionario con Sshutout y Fail2ban

# LIBRO CERRADO

Los servicios que requieren para identificarse un nombre de usuario y una contraseña son objetivos potenciales para ataques de diccionario. Sshutout y Fail2ban introducen penalizaciones de tiempo para los intentos no válidos. **POR CHARLY KÜHNAST.**

**S**shutout [1] es un daemon escrito en C que comprueba, en intervalos regulares, un archivo de registro de las autenticaciones SSH no válidas. Si descubre un patrón de intentos fallidos de un solo cliente, lo bloquea siguiendo reglas de iptables. Tras un retraso configurable, la penalización se revoca automáticamente.

## Cómo Activar el Ban

Lo único que necesitamos para instalar el tarball de 32KB es el habitual `make; make install`.

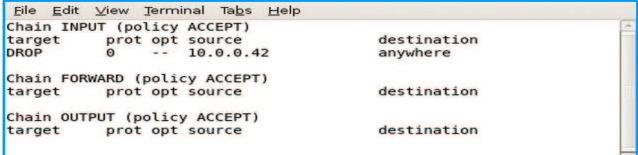
Tras la instalación, el daemon queda localizado en `/usr/local/sbin`; su archivo de configuración es `/etc/sshutout.conf`.

Este archivo de configuración nos permite personalizar un buen número de parámetros. Es extremadamente importante especi-

ficar el nombre adecuado para el archivo de registro que deseamos que Sshutout monitorice. El nombre predeterminado es `/var/log/messages`, pero la mayoría de distribuciones guardan su información de inicio de sesión en otros archivos de registro. En Ubuntu, por ejemplo, hará falta la siguiente:

```
sshd_log_file=/var/log/auth.log
```

La opción de umbral especifica el valor de LOS (LOS = “level of stupidity”, valor de estupidez). En otras palabras, define cuántos intentos fallidos puede hacer un cliente antes de que se le bloquee de forma temporal. Si se



```
File Edit View Terminal Tabs Help
Chain INPUT (policy ACCEPT)
target prot opt source destination
DROP 0 -- 10.0.0.42 anywhere
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

Figura 1: El comando de iptables -L muestra las reglas del firewall activo.

posee una buena cantidad de contraseñas o una memoria pobre, quizás sea conveniente aumentar el umbral (que por defecto es de 4).

## Tiempo de Cierre

El tiempo de cierre está relacionado con el valor de umbral. Por un lado, hay que prevenir los ataques de fuerza bruta tan efectivamente como sea posible; por el otro, no es una buena idea dejar fuera a algún usuario durante horas sólo porque le resulta difícil recordar sus contraseñas después de una noche de marcha. El valor por defecto, cinco minutos, es un acuerdo útil:

```
delay_penalty = 300.
```

Esto ofrece al usuario tiempo suficiente para beberse un café y refrescarse la memoria. Por otro lado, es posible excluir a los olvidadizos conocidos desde el principio. El archivo de configuración incluye una línea para crear una lista blanca en la cual introducir nombres o direcciones IP que no queremos dejar fuera.

Sshutout también redacta listas blancas de forma más o menos autónoma. Una vez lanzado, muestra una vista general de los parámetros actuales, incluyendo las líneas siguientes:

```
Whitelist:
213.133.98.97
```



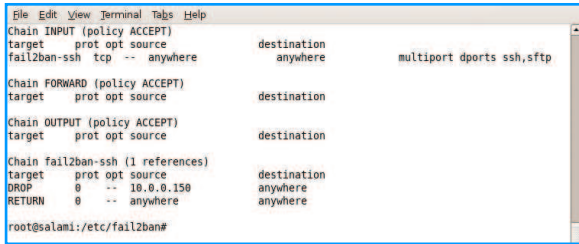


Figura 2: Fail2ban expulsa a un cliente que trate de establecer una conexión tras un número predefinido de intentos fallidos.

```
10.0.0.254
10.0.0.214
```

Esto significa que la lista blanca de Sshutout excluye automáticamente su propia dirección IP, la dirección de la puerta de acceso por defecto y el nombre del servidor. Esto no debería ser muy peligroso, pero algún administrador más cauteloso quizá quiera cambiar este comportamiento especificando `auto_whitelist = no`.

Por defecto, Sshutout guarda también registro de sus propias actividades en `/var/log/sshutout.log`. Si se bloquea la entrada a un cliente, se escribe una entrada en el archivo:

```
10.0.0.42 blocked on Sat
Feb 02 15:32:32 2008
```

`iptables -L` permite ver las normas del firewall subyacente (Figura 1). Finalmente, una función de excepción de manejo se encarga de los casos en los que un cliente inicia una conexión SSH pero no realiza una entrada (lo que suele ocurrir en ataques de denegación de servicio). Sshutout ignora este tipo de conexiones por defecto. La entrada `illegal user = yes` indica al programa que debe tratarlas igual que cualquier otro intento fallido.

## Fail2ban: Protección General

Fail2ban [2] utiliza básicamente el mismo método que Sshutout; sin embargo, no está

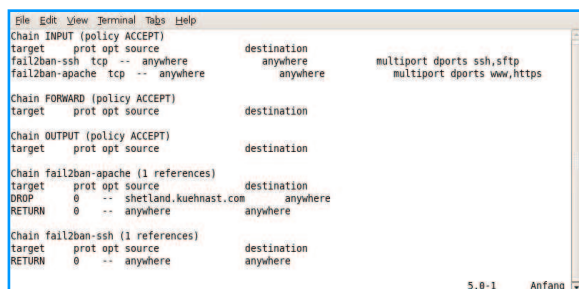


Figura 3: Reglas para Apache: Fail2ban puede proteger bas-tantes aplicaciones distintas.

restringido a SSH. De hecho, puede proteger más o menos cualquier servicio que requiera una autenticación del usuario. Fail2ban toma sus decisiones a partir de las entradas del archivo de registro, aunque emplea un método técnico diferente.

La herramienta está dividida en dos componentes: un servidor y un cliente. El primero monitoriza los archivos de registro y las normas de iptables. El administrador puede utilizar el cliente para dar instrucciones al servidor para cambiar, por ejemplo, el nivel del login.

El archivo de configuración `jail.conf` define el número de servicios que Fail2ban puede proteger, con `ssh` encabezando la lista:

```
[ssh]
enabled = true
port = ssh,sftp
filter = sshd
logpath = /var/log/auth.log
maxretry = 6
```

Las secciones que siguen contienen parámetros para otros servicios. Cada una de estas secciones es una “jaula” (“jail” en inglés) en la jerga de Fail2ban. La entrada `filter = sshd` es equivalente a un archivo en el directorio `/etc/fail2ban/filter.d`. El archivo contiene una expresión regular que el servidor de Fail2ban busca en el archivo de registro. En este punto no se puede configurar el tiempo de cierre, dado que este valor está configurado en 600 segundos en la sección global `[DEFAULT]`. Si resulta demasiado largo, puede añadirse una entrada `bantime = 300` en la sección `[ssh]`.

## Protección Contra DoS

SSH tiene dos jaulas: la que acabamos de mencionar, y `sshd-ddos`. Esta última no está diseñada para prevenir un intento de adivinar contraseñas, sino de contrarrestar los ataques DoS (“denial-of-service” o denegación de servicio), que abren conexiones con el daemon SSH sin introducir una contraseña. En caso de ataque DoS, el archivo de registro contendrá mensajes como éste:

```
sshd: Did not receive 2
identification string 2
from
10.0.0.150
```

Aunque es posible configurar múltiples expresiones regulares por jaula, muchos administradores prefieren asignar distintos tiempos de expulsión para ataques DoS distribuidos (DdoS) en oposición a los intentos fallidos de login. Así pues, crear las dos categorías SSH y `sshd-ddos` tiene mucho sentido.

La jaula SSH es la única configurada en `enabled = true` por defecto; todas las demás (incluyendo `sshd-ddos`) hay que activarlas manualmente.

## Esquema de Protección

Si un usuario introduce la contraseña equivocada varias veces, los resultados de Fail2ban son similares a los de Sshutout: se activa una regla de iptable y se cierran todas las conexiones del ordenador “ofensor” durante los siguientes cinco minutos (Figura 2).

La protección para otros servicios sigue el mismo patrón (Figura 3). Si tenemos algunas páginas web protegidas de login en nuestro servidor Apache, Fail2ban ofrece una jaula como ésta:

```
[apache]
enabled = false
port = http,https
filter = apache-auth
logpath = 2
/var/log/apache*/access.log
maxretry = 3
```

que sólo tendremos que modificar ligeramente. La versión Apache que yo empleo escribe los mensajes de error en un archivo `error.log` separado, y no en `access.log`. Tras configurar `enabled = true`, podemos activar la jaula.

Consejo: hay un método más elegante que reiniciar el daemon de Fail2ban (lo cual podría desactivar las normas activas de iptables). Consiste en enviar el siguiente comando del cliente Fail2ban al servidor:

```
fail2ban-client start apache
```

el cual le indica al servidor que añada una entrada `[apache]` a la lista de jaulas activas. Para probar todo esto, introduce unas cuantas contraseñas inválidas, y se activó una nueva regla de iptables.

## RECURSOS

- [1] Sshutout: <http://www.techfinesse.com/sshutout/sshutout.html>
- [2] Fail2ban: <http://www.fail2ban.org>