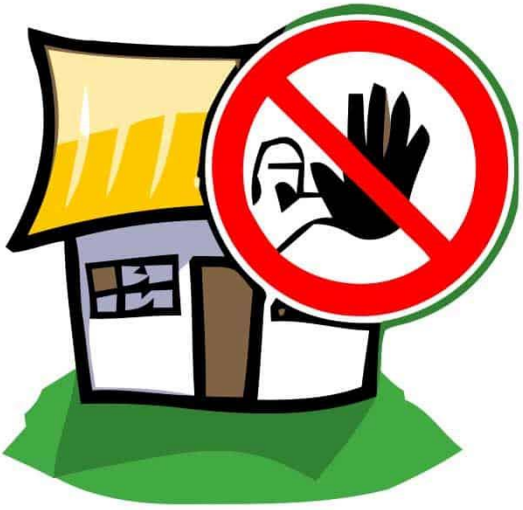

# FAIL2BAN

Óscar Osorio Giráldez

## Índice:


- Qué es.
- Cómo funciona.
- Instalación.
- Configuración:
    - Forma directa.
    - Forma ordenada.


## Qué es Fail2Ban:

Fail2Ban es una aplicación Python utilizada para prevenir intrusiones externas y actuar contra ellas en tiempo real penalizando o bloqueando conexiones con comportamiento sospechoso, especialmente para evitar ataques por fuerza bruta. Esta aplicación llama la atención especialmente por su simplicidad, ya que podemos definir qué programas monitorizar, cómo detectar intentos de ataque, y cómo actuar ante estos en muy pocos archivos de configuración.

## Cómo funciona Fail2Ban:

El funcionamiento se basa en una lectura periódica de los logs del sistema en busca de patrones o reglas para aplicar penalizaciones a la IP implicada.

Por ejemplo, podemos definir que Fail2Ban monitorice nuestro servidor SSH (los logs de ssh) y que cuándo haya más de 3 intentos de conexión desde una misma IP fallando la autenticación se bloquee esta IP de forma permanente o temporal. Esta IP se podría bloquear especificando la acción (o penalización) de que se actualice la tabla (iptable) de direcciones del firewall. Además del bloqueo de la IP podemos definirle otras acciones, por ejemplo, que nos envíe una notificación por correo con el log en cuestión que refleja el ataque.

## Instalación:

Como requisitos debemos tener Python 2.3 o superior y log4py (sourceforge.net/projects/log4py).

Lo siguiente será descargarnos e instalar la versión de fail2ban que queramos (disponible en código fuente, paquete .deb o .rpm)(también se puede descargar por apt: apt-get install fail2ban)

```
Fail2Ban v0.5.1 reads log file that contains password failure report
and bans the corresponding IP addresses using firewall rules.

-b start fail2ban in background.
-d start fail2ban in debug mode.
-c <FILE> read configuration file FILE.
-p <FILE> create PID lock in FILE.
-h display this help message.
-i <IP(s)> IP(s) to ignore.
-k kill a currently running Fail2Ban instance.
-r <VALUE> allow a max of VALUE password failure.
-t  ban IP for TIME seconds.
-v verbose. Use twice for greater effect.
-V print software version.
```

*Ejecuta fail2ban –h para ver en consola la ayuda del programa [3].*

## Configuración:

Una vez instalado tenemos que configurarlo. Puedes encontrar el archivo de configuración por defecto en /etc/fail2ban.conf.example, este archivo al tener extensión .example no es utilizado, por lo que tendrás que renombrarlo y dejarlo como fail2ban.conf.

Según he visto en diferentes referencias hay diferentes formas de configurar completamente tu fail2ban, la primera es definir toda la configuración en el archivo de configuración principal (fail2ban.conf) y otra es hacerlo de forma más ordenada separando cada parte de la configuración en diferentes archivos y directorios y con diferentes jerarquías.

### *Forma directa:*

Para la primera forma de las comentadas solo necesitamos las siguientes directivas de configuración, la configuración básica es igual en ambas formas(<ip> para utilizar la variable con la IP baneada):

Configuración básica (fail2ban.conf):

- Background : para ejecutar fail2ban como demonio, lo cual nos interesa si queremos que funcione en tiempo real y constantemente.

```
# Option: background.
# Notes.: start fail2ban as a daemon. Output is redirect to logfile..
# Values: [true | false] Default: false.
#.
background = true
```

*[3]*

- Logtargets : para definir donde guardar los logs de fail2ban.

```
# Option: logtargets.
# Notes.: log targets. Space separated list of logging targets..
# Values: STDERR SYSLOG file Default: /var/log/fail2ban.log.
#
logtargets = /var/log/fail2ban.log
```

*[3]*

- Maxretry : número de intentos para banear esa red.

```
# Option: maxretry.
# Notes.: number of retrys before IP gets banned..
# Values: NUM Default: 3.
#.
maxretry = 3
```

*[3]*

- Bantime: número de segundos que será baneado una IP.

```
# Option: bantime.
# Notes.: number of seconds an IP will be banned..
# Values: NUM Default: 600.
#.
bantime = 36000
```

*[3]*

- Ingoreip: IPs que serán ignoradas y no se banearán. (p.e. ignoreip = 192.168.1.0/24 para ignoras todas las de la red local).

```
# Option: ignoreip.
# Notes.: space separated list of IP's to be ignored by fail2ban..
# You can use CIDR mask in order to specify a range..
# Example: ignoreip = 192.168.0.1/24 123.45.235.65.
# Values: IP Default: 192.168.0.0/24.
#.
ignoreip = 192.168.1.0/24
```

*[3]*

- Cmdstart y cmdend: para definir una acción (comando) al inicio y cierre de fail2ban.

```
# Option: cmdstart.
# Notes.: command executed once at the start of Fail2Ban.
# Values: CMD Default:.
#.
cmdstart = echo "Se ha iniciado fail2ban" | mail -s "Fail2ban" manuelATtodo-linux.com..
```

```
# Option: cmdend.
# Notes.: command executed once at the end of Fail2Ban.
# Values: CMD Default:.
#.
cmdend =echo "Se ha detenido fail2ban" | mail -s "Fail2ban" manuelATtodo-linux.com<
```

*[3]*

Configuración de aviso por correo [Mail] (fail2ban.conf):

- Enabled: para que te avise por correo cuando se banee una IP.
- Host y port: para definir la dirección de nuestro servidor de correo.
- From y to: para definir las direcciones de correo.
- Subject y message: para definir asunto y mensaje del correo.

```
[MAIL]

# Option: enabled.
# Notes.: enable mail notification when banning an IP address..
# Values: [true | false] Default: false.
#.
enabled = true
```

```
# Option: host.
# Notes.: host running the mail server..
# Values: STR Default: localhost.
#.
host = localhost

# Option: port.
# Notes.: port of the mail server..
# Values: INT Default: 25.
#.
port = 25
```

```
# Option: from.
# Notes.: e-mail address of the sender..
# Values: MAIL Default: fail2ban.
#.
from = fail2ban

# Option: to.
# Notes.: e-mail addresses of the receiver. Addresses are space.
# separated..
# Values: MAIL Default: root.
#.
to = manuelATtodo-linux.com
```

```
# Option: subject.
# Notes.: subject of the e-mail..
# Tags: <ip> IP address.
# <failures> number of failures.
# <failtime> unix timestamp of the last failure.
# Values: TEXT Default: [Fail2Ban] Banned <ip>.
#.
subject = [Fail2Ban] Se ha baneado a <ip>..
```

```
# Option: message.
# Notes.: message of the e-mail..
# Tags: <ip> IP address.
# <failures> number of failures.
# <failtime> unix timestamp of the last failure
# <br> new line.
# Values: TEXT Default: .
#.
message = La ip <ip> ha sido baneada por Fail2Ban despues de
```

*Ejemplo de configuración para envío de correo al banear [3].*

Configuración para monitorizar una aplicación (fail2ban.conf):

- Enabled: para activarlo.
- Logfile: para decirle donde está el log de la aplicación que tiene que monitorizar.
- Fwban y fwunban: para especificar el comando a ejecutar cuando se banee una IP y cuando deje de estar baneada.
- Failregex: expresión regular para detectar en el log una acción sospechosa y en la que haya que banear la IP.

```
[Apache]

# Option: enabled.
# Notes.: enable monitoring for this section..
# Values: [true | false] Default: false.
#.
enabled = true
```

```
# Option: logfile.
# Notes.: logfile to monitor..
# Values: FILE Default: /var/log/httpd/access_log.
#.
logfile = /var/log/httpd/access_log
```

```
# Option: fwban.
# Notes.: command executed when banning an IP. Take care that the
# command is executed with Fail2Ban user rights..
# Tags: <ip> IP address.
# <failures> number of failures.
# <failtime> unix timestamp of the last failure.
# <bantime> unix timestamp of the ban time.
# Values: CMD.
# Default: iptables -I INPUT 1 -i eth0 -s <ip> -j DROP.
#.
fwban = iptables -I INPUT -s <ip> -j DROP

# Option: fwunban.
# Notes.: command executed when unbanning an IP. Take care that the
# command is executed with Fail2Ban user rights..
# Tags: <ip> IP address.
# <bantime> unix timestamp of the ban time.
# <unbantime> unix timestamp of the unban time.
# Values: CMD.
# Default: iptables -D INPUT -i eth0 -s <ip> -j DROP.
#.
fwunban = iptables -D INPUT -s <ip> -j DROP
```

```
# Option: failregex.
# Notes.: regex to match the password failure messages in the logfile..
# Values: TEXT Default: authentication failure|user .* not found.
#.
failregex = authentication failure|user .* not found|*. User notfound.
```

*Ejemplo de configuración para Apache [3].*

*Forma ordenada:*

Para la segunda forma tenemos el archivo fail2ban.conf y fail2ban.local para la configuración del programa (básica) (los archivos .conf prevalecen sobre los .local, es decir, lo que se defina en un .local solo podrá complementar la configuración definida en un .conf, no cambiarla), jail.conf y jail.local para la configuración de monitorización de cada programa, donde se define la parte de *Configuración para monitorizar una aplicación* vista anteriormente, directorio filter.d con archivos .conf para cada programa a monitorizar donde se definen las expresiones regulares utilizadas para detectar patrones en los logs y el directorio action.d con archivos .conf para cada acción a llevar a cabo como puede ser banear una ip u mandar un correo, en estos están los scripts necesarios para llevar a cabo la acción.

```
/etc/fail2ban/
├── action.d
│   ├── dummy.conf
│   ├── hostsdeny.conf
│   ├── iptables.conf
│   ├── mail-whois.conf
│   ├── mail.conf
│   └── shorewall.conf
├── fail2ban.conf
├── fail2ban.local
├── filter.d
│   ├── apache-auth.conf
│   ├── apache-noscript.conf
│   ├── couriersmtp.conf
│   ├── postfix.conf
│   ├── proftpd.conf
│   ├── qmail.conf
│   ├── sasl.conf
│   ├── sshd.conf
│   └── vsftpd.conf
├── jail.conf
└── jail.local
```

*Esquema de directorios y archivos de fail2ban [1].*

Directivas de configuración para esta segunda forma por cada archivo explicado:

- Fail2ban.conf y fail2ban.local como las anteriores directivas.
- Jail.conf y jail.local:
    - [nombre]: definimos un nombre, es como abrir una función, en este caso es la configuración de monitorización de un programa.
    - Enabled: para activar esta "función".
    - Filter: nombre del archivo (sin el .conf) en filter.d donde están definidas las expresiones regulares para buscar patrones en los logs.
    - Action: ponemos las diferentes acciones con las variables que necesiten para ejecutar los scripts de la forma nombre[var1=valor1,var2=valor2], el nombre es el del archivo en action.d sin el .conf.
    - Logpath: para definir la ubicación del archivo de logs de esta aplicación (la que vamos a monitorizar).
    - Maxretry: número de detecciones máximas para banear.
    - Bantime: número de segundos que durará el baneo (número negativo para hacerlo permanente)

```
[ssh-iptables]

enabled  = true
filter   = sshd
action   = iptables[name=SSH, port=ssh, protocol=tcp]
           sendmail-whois[name=SSH, dest=you@example.com, sender=fail2ban@example.com...
logpath  = /var/log/secure
maxretry = 5
```

*Ejemplo de configuración para ssh [1].*

- Filtro.conf en filter.d (nombreDelFiltro.conf):
    - Failregex: para definir todas las expresiones regulares a usar en la detección de irregularidades en los logs.

```
failregex = ^%(__prefix_line)s(?:error: PAM: )?[aA]uthentication (?:failure|error) for .* from <HOST>( via \S+)?\s*$
            ^%(__prefix_line)s(?:error: PAM: )?User not known to the underlying authentication module for .* from <HOST>\s*$
            ^%(__prefix_line)sFailed \S+ for .*? from <HOST>(?: port \d*)?(?: ssh\d*)?(: (ruser .*|(\S+ ID \S+ \(serial \d+\) C
            ^%(__prefix_line)sROOT LOGIN REFUSED.* FROM <HOST>\s*$
            ^%(__prefix_line)s[iI](?:llegal|nvalid) user .* from <HOST>\s*$
            ^%(__prefix_line)sUser .+ from <HOST> not allowed because not listed in AllowUsers\s*$
            ^%(__prefix_line)sUser .+ from <HOST> not allowed because listed in DenyUsers\s*$
            ^%(__prefix_line)sUser .+ from <HOST> not allowed because not in any group\s*$
            ^%(__prefix_line)srefused connect from \S+ \(<HOST>\)\s*$
            ^%(__prefix_line)sReceived disconnect from <HOST>: 3: \S+: Auth fail$
            ^%(__prefix_line)sUser .+ from <HOST> not allowed because a group is listed in DenyGroups\s*$
            ^%(__prefix_line)sUser .+ from <HOST> not allowed because none of user's groups are listed in AllowGroups\s*$
```

*Ejemplo de expresiones regulares para log de ssh [1].*

- Acción.conf en action.d (nombreDeLaAcción.conf):
    - Actionstart y actionstop: script al iniciar y finalizar fail2ban.
    - Actioncheck: script de checkeo tras el script de banear una IP.
    - Actionban: script para banear una IP.
    - Actionunban: script para quitar el baneo a una IP.

```
# Option:  actionstart
# Notes.:  command executed once at the start of Fail2Ban.
# Values:  CMD
#
actionstart = <iptables> -N f2b-<name>
              <iptables> -A f2b-<name> -j <returntype>
              <iptables> -I <chain> -p <protocol> --dport <port> -j f2b-<name>

# Option:  actionstop
# Notes.:  command executed once at the end of Fail2Ban
# Values:  CMD
#
actionstop = <iptables> -D <chain> -p <protocol> --dport <port> -j f2b-<name>
             <iptables> -F f2b-<name>
             <iptables> -X f2b-<name>

# Option:  actioncheck
# Notes.:  command executed once before each actionban command
# Values:  CMD
#
actioncheck = <iptables> -n -L <chain> | grep -q 'f2b-<name>[ \t]'

# Option:  actionban
# Notes.:  command executed when banning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:    See jail.conf(5) man page
# Values:  CMD
#
actionban = <iptables> -I f2b-<name> 1 -s <ip> -j <blocktype>

# Option:  actionunban
# Notes.:  command executed when unbanning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:    See jail.conf(5) man page
# Values:  CMD
#
actionunban = <iptables> -D f2b-<name> -s <ip> -j <blocktype>
```

*Ejemplo de script en acción para banear IP con el firewall (iptable) [1].*

BIBLIOGRAFÍA:

[1] https://raiolanetworks.es/blog/bloquear-ataques-dos-con-fail2ban-en-linux/

[2] https://es.wikipedia.org/wiki/Fail2ban

[3] https://www.fail2ban.org/wiki ->

https://www.fail2ban.org/wiki/index.php/HOWTO_fail2ban_spanish