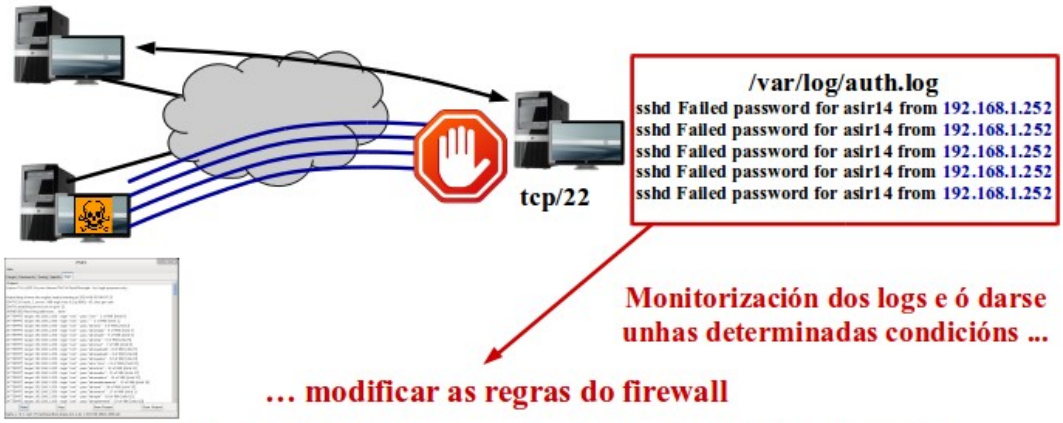
fail2ban es una aplicación que monitoriza los archivos de log en la búsqueda de información interesante, en especial de errores de autenticación. Analizando esa información y tomando como referencia los parámetros de configuración introducidos por el administrador, es capaz de distinguir entre un simple error de autenticación y un intento de ataque. Una vez detectado el ataque, fail2ban puede realizar diferentes acciones, en la mayor parte de los casos procederá a reconfigurar el firewall del equipo para bloquear al equipo atacante y/o notificarlo al administrador mediante un email.

Como se ya se vió en clase, los ataques de fuerza bruta o diccionario dejan huellas en el sistema de logs:

```
$ less /var/log/auth.log
...
Dec  4 11:43:20 ubuntu sshd[1003]: Failed password for root from 192.168.1.199 port 42410 ssh2
Dec  4 11:43:20 ubuntu sshd[994]: Failed password for root from 192.168.1.199 port 42401 ssh2
Dec  4 11:43:20 ubuntu sshd[1006]: Failed password for root from 192.168.1.199 port 42413 ssh2
Dec  4 11:43:20 ubuntu sshd[998]: Failed password for root from 192.168.1.199 port 42405 ssh2
Dec  4 11:43:20 ubuntu sshd[1001]: Failed password for root from 192.168.1.199 port 42408 ssh2
Dec  4 11:43:20 ubuntu sshd[1000]: Failed password for root from 192.168.1.199 port 42407 ssh2
Dec  4 11:43:20 ubuntu sshd[1007]: Failed password for root from 192.168.1.199 port 42414 ssh2
...
```

fail2ban detecta estas huellas y procede a actuar. En la siguiente imagen se da una visión general del proceso:



iptables -I INPUT 1 -p tcp –dport ssh -s *IP_PIRATA* -j DROP

Para instalar fail2ban ejecutamos los comandos:

```
$ sudo apt update
$ sudo apt install fail2ban
```

Los archivos de configuración de fail2ban están ubicados en el directorio `/etc/fail2ban`.

```
$ ls -lahF /etc/fail2ban/
total 29K
drwxr-xr-x 2 root root   65 Apr 25 08:29 action.d/
-rw-r--r-- 1 root root 2.8K Jan 11  2020 fail2ban.conf
drwxr-xr-x 2 root root    2 Mar  2  2020 fail2ban.d/
drwxr-xr-x 3 root root   90 Apr 25 08:29 filter.d/
-rw-r--r-- 1 root root  26K Jan 11  2020 jail.conf
drwxr-xr-x 2 root root    3 Apr 25 08:29 jail.d/
-rw-r--r-- 1 root root  645 Jan 11  2020 paths-arch.conf
-rw-r--r-- 1 root root 2.8K Jan 11  2020 paths-common.conf
-rw-r--r-- 1 root root  573 Jan 11  2020 paths-debian.conf
-rw-r--r-- 1 root root  738 Jan 11  2020 paths-opensuse.conf
```

Hay tres conceptos básicos para entender la configuración y la forma de trabajar de fail2ban:

- **filter**: expresión regular que busca un patrón en un log (p.e. error de autenticación). En la carpeta `/etc/fail2ban/filter.d/` hay archivos con filtros para los servicios/aplicaciones más comunes y no únicamente para ssh:

```
$ ls -lhF /etc/fail2ban/filter.d/
total 135K
-rw-r--r-- 1 root root  467 Jan 11  2020 3proxy.conf
-rw-r--r-- 1 root root 3.2K Jan 11  2020 apache-auth.conf
-rw-r--r-- 1 root root 2.8K Jan 11  2020 apache-badbots.conf
-rw-r--r-- 1 root root 1.3K Jan 11  2020 apache-botsearch.conf
-rw-r--r-- 1 root root 1.6K Jan 11  2020 apache-common.conf
-rw-r--r-- 1 root root  324 Jan 11  2020 apache-fakegooglebot.conf
-rw-r--r-- 1 root root  511 Jan 11  2020 apache-modsecurity.conf
-rw-r--r-- 1 root root  596 Jan 11  2020 apache-nohome.conf
-rw-r--r-- 1 root root 1.3K Jan 11  2020 apache-noscript.conf
-rw-r--r-- 1 root root 2.2K Jan 11  2020 apache-overflows.conf
-rw-r--r-- 1 root root  362 Jan 11  2020 apache-pass.conf
-rw-r--r-- 1 root root 1020 Jan 11  2020 apache-shellshock.conf
-rw-r--r-- 1 root root 3.5K Jan 11  2020 assp.conf
-rw-r--r-- 1 root root 2.4K Jan 11  2020 asterisk.conf
...
-rw-r--r-- 1 root root  260 Jan 11  2020 squid.conf
-rw-r--r-- 1 root root  191 Jan 11  2020 squirrelmail.conf
-rw-r--r-- 1 root root 6.5K Jan 11  2020 sshd.conf
-rw-r--r-- 1 root root  363 Jan 11  2020 stunnel.conf
-rw-r--r-- 1 root root  649 Jan 11  2020 suhosin.conf
-rw-r--r-- 1 root root  890 Jan 11  2020 tine20.conf
-rw-r--r-- 1 root root 1.7K Jan 11  2020 traefik-auth.conf
-rw-r--r-- 1 root root  374 Jan 11  2020 uwimap-auth.conf
-rw-r--r-- 1 root root  637 Jan 11  2020 vsftpd.conf
-rw-r--r-- 1 root root  444 Jan 11  2020 webmin-auth.conf
-rw-r--r-- 1 root root  520 Jan 11  2020 wuftpd.conf
-rw-r--r-- 1 root root  521 Jan 11  2020 xinetd-fail.conf
-rw-r--r-- 1 root root  912 Jan 11  2020 znc-adminlog.conf
-rw-r--r-- 1 root root  524 Jan 11  2020 zoneminder.conf
```

  Por ejemplo, en el archivo `/etc/fail2ban/filter.d/sshd.conf` encontramos las siguientes expresiones regulares para encontrar incidencias en el servicio ssh:

```
cmnfailre = ^[aA]uthentication (?:failure|error|failed) for <F-USER>.*</F-USER> from <HOST>( via \S+)?%(__suff)s$
            ^User not known to the underlying authentication module for <F-USER>.*</F-USER> from <HOST>%(__suff)s$
            ^Failed publickey for invalid user <F-USER>(?P<cond_user>\S+)|(?:(?! from ).)*?</F-USER> from <HOST>%
(__on_port_opt)s(?: ssh\d*)?(?(cond_user): |(?:(?:(?! from ).)*)$)
            ^Failed \b(?!publickey)\S+ for (?P<cond_inv>invalid user )?<F-USER>(?P<cond_user>\S+)|(?(cond_inv)(?:
(?! from ).)*?|[^:]+)</F-USER> from <HOST>%(__on_port_opt)s(?: ssh\d*)?(?(cond_user): |(?:(?:(?! from ).)*)$)
            ^<F-USER>ROOT</F-USER> LOGIN REFUSED FROM <HOST>
            ^[iI](?:llegal|nvalid) user <F-USER>.*?</F-USER> from <HOST>%(__suff)s$
            ^User <F-USER>.+</F-USER> from <HOST> not allowed because not listed in AllowUsers%(__suff)s$
            ^User <F-USER>.+</F-USER> from <HOST> not allowed because listed in DenyUsers%(__suff)s$
            ^User <F-USER>.+</F-USER> from <HOST> not allowed because not in any group%(__suff)s$
            ^refused connect from \S+ \(<HOST>\)
            ^Received <F-MLFFORGET>disconnect</F-MLFFORGET> from <HOST>%(__on_port_opt)s:\s*3: .*: Auth fail%
(__suff)s$
            ^User <F-USER>.+</F-USER> from <HOST> not allowed because a group is listed in DenyGroups%(__suff)s$
            ^User <F-USER>.+</F-USER> from <HOST> not allowed because none of user's groups are listed in
AllowGroups%(__suff)s$
```

- **action**: define varios comandos a ejecutar en diferentes momentos. En la carpeta `/etc/fail2ban/action.d/` hay archivos con acciones para diferentes tipos de firewalls:

```
$ ls -lhF /etc/fail2ban/action.d/
total 104K
-rw-r--r-- 1 root root  587 Aug  1  2015 apf.conf
-rw-r--r-- 1 root root  640 Aug  1  2015 badips.conf
-rw-r--r-- 1 root root  11K Aug  1  2015 badips.py
-rw-r--r-- 1 root root 2.6K Aug  1  2015 blocklist_de.conf
-rw-r--r-- 1 root root 2.8K Aug  1  2015 bsd-ipfw.conf
```

```
-rw-r--r-- 1 root root 1.9K Aug  1  2015 cloudflare.conf
-rw-r--r-- 1 root root 4.0K Aug  1  2015 complain.conf
-rw-r--r-- 1 root root 7.4K Aug  1  2015 dshield.conf
-rw-r--r-- 1 root root 1.2K Aug  1  2015 dummy.conf
-rw-r--r-- 1 root root 1.6K Aug  1  2015 firewallcmd-allports.conf
-rw-r--r-- 1 root root 1.5K Aug  1  2015 firewallcmd-ipset.conf
-rw-r--r-- 1 root root 2.1K Aug  1  2015 firewallcmd-multiport.conf
-rw-r--r-- 1 root root 2.0K Aug  1  2015 firewallcmd-new.conf
-rw-r--r-- 1 root root 1.4K Aug  1  2015 hostsdeny.conf
-rw-r--r-- 1 root root 1.5K Aug  1  2015 ipfilter.conf
-rw-r--r-- 1 root root 1.4K Aug  1  2015 ipfw.conf
-rw-r--r-- 1 root root 1.5K Aug  1  2015 iptables-allports.conf
-rw-r--r-- 1 root root 1.9K Aug  1  2015 iptables-common.conf
-rw-r--r-- 1 root root 1.8K Aug  1  2015 iptables-ipset-proto4.conf
-rw-r--r-- 1 root root 1.8K Aug  1  2015 iptables-ipset-proto6-allports.conf
-rw-r--r-- 1 root root 1.8K Aug  1  2015 iptables-ipset-proto6.conf
-rw-r--r-- 1 root root 1.9K Aug  1  2015 iptables-multiport-log.conf
-rw-r--r-- 1 root root 1.4K Aug  1  2015 iptables-multiport.conf
-rw-r--r-- 1 root root 1.5K Aug  1  2015 iptables-new.conf
-rw-r--r-- 1 root root 2.3K Aug  1  2015 iptables-xt_recent-echo.conf
-rw-r--r-- 1 root root 1.4K Aug  1  2015 iptables.conf
-rw-r--r-- 1 root root 2.3K Aug  1  2015 mail-buffered.conf
-rw-r--r-- 1 root root 1.1K Aug  1  2015 mail-whois-common.conf
...
```

Por ejemplo, en el archivo `/etc/fail2ban/action.d/iptables.conf` encontramos unos ejemplos de acciones. Revisándolas podemos ver :

- `actionstart`: al arrancar fail2ban se crea una cadena nueva *f2b-<name>* hacia la que se dirigen los paquetes de un determinado tipo (indicado por el protocolo y puerto/s).
- `actionban`: crea una regla para bloquear el tráfico del tipo gestionado por la cadena *f2b-<name>* con origen el equipo indicado en *<ip>*.
- `actionunban`: borrar la regla que bloquea el tráfico del tipo gestionado por la cadena *f2b-<name>* con origen el equipo indicado en *<ip>*.
- `actionstop`: borrar la cadena creada por `actionstart` y las reglas asociadas.

```
# 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>
```

- **jail**: es una combinación de un filter y una o varias actions. Para evitar problemas con futuras actualizaciones que puedan modificar el contenido del ficheiro jail.conf se recomienda crear un fichero jail.local donde pondremos nuestra configuración particular.

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

En el archivo `/etc/fail2ban/jail.local` se definen algunos parámetros de configuración globales y se configuran los jails con los que trabajará fail2ban (puede manejar varias jails a la vez).

```
[vsftpd]
# or overwrite it in jails.local to be
# logpath = %(syslog_authpriv)s
# if you want to rely on PAM failed login attempts
# vsftpd's failregex should match both of those formats
port     = ftp,ftp-data,ftps,ftps-data
logpath  = %(vsftpd_log)s
```

Revisando los parámetros globales y el jail para ssh en `/etc/fail2ban/jail.local` y `/etc/fail2ban/jail.d/defaults-debian.conf` tenemos que:

Si al monitorizar el archivo /var/log/auth.log (`logpath`) durante los últimos 600 segundos (`findtime`) aparecen 5 (`maxretry`) entradas asociadas al mesmo equipo origen con errores indicados en el filter /etc/fail2ban/filter.d/sshd.conf (`filter`), se procederá a 'banear' (`action`) ese equipo durante 600 segundos (`bantime`). Por 'banear' nos referimos a ejecutar la acción `iptables-multiport` con los parámetros puerto 22/tcp (`protocol` e `port`) y la IP del equipo problemático; es decir, se crea una regla para bloquear al equipo problemático y a los 600 segundos proceder a borrarla para desbloquearlo. Fijarse que lo que está fuera de la sección `[sshd]` es un parámetro global y dentro de la `[sshd]` (tanto en `jail.local` como en `defaults-debian.conf`) es local y tiene prioridade sobre el global; es decir, se pueden tener unos valores por defecto para esos parámetros y en los jails que nos interesen unos valores específicos.

/etc/fail2ban/jail.local

```
ignoreip = 127.0.0.1/8
bantime  = 600
findtime = 600
maxretry = 5
# "enabled" enables the jails.
#  By default all jails are disabled, and it should stay this way.
#  Enable only relevant to your setup jails in your .local or jail.d/*.conf
#
# true:  jail will be enabled and log files will get monitored for changes
# false: jail is not enabled
enabled = false
banaction = iptables-multiport
protocol = tcp
chain = INPUT
# The simplest action to take: ban only
action_    =    %(banaction)s[name=%(__name__)s,    bantime="%(bantime)s",    port="%(port)s",
protocol="%(protocol)s", chain="%(chain)s"]
action = %(action_)s
# SSH servers
```

```
#
[sshd]
port    = ssh
logpath = %(sshd_log)s
----
/etc/fail2ban/jail.d/defaults-debian.conf
[sshd]
enabled = true
```

fail2ban puede ser gestionado a través de:

```
$ sudo systemctl {start | stop | restart | status } fail2ban.service
```

o a través de

```
$ sudo fail2ban-client
add          flushlogs  get          help         ping         reload      set          start
status     stop       version
```

fail2ban e iptables

Para ver como fail2ban manipula el ruleset del equipo, partimos con el firewall del servidor aceptando todo tipo de tráfico:

```
$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Después de instalar fail2ban tenemos una regla que recoge todo el tráfico ssh con destino al servidor (cadena INPUT) y lo envía a la cadena f2b-ssh. En ese momento en la cadena f2b-ssh la única regla que hay es un RETURN que devuelve los paquetes ssh a la cadena INPUT para seguir procesándolos; y por lo tanto llegarían al final de INPUT, donde se les aplicaría la policy (ACCEPT).

```
$ sudo iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    f2b-sshd   tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 22

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain f2b-sshd (1 references)
num  target     prot opt source               destination
1    RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Si lanzamos un ataque de diccionario contra el servidor, se vemos que se inicia pero que al poco tiempo deja de progresar:

```
root@kali:~# medusa -h 10.11.224.7 -u ubuntu -P wordlist_breve.txt -M ssh -f  -t 3
ACCOUNT CHECK: [ssh] Host: 10.11.224.7 (1 of 1, 0 complete) User: ubuntu (1 of 1, 0 complete)
Password: abrumar (1 of 984 complete)
ACCOUNT CHECK: [ssh] Host: 10.11.224.7 (1 of 1, 0 complete) User: ubuntu (1 of 1, 0 complete)
Password: abroncar (2 of 984 complete)
ACCOUNT CHECK: [ssh] Host: 10.11.224.7 (1 of 1, 0 complete) User: ubuntu (1 of 1, 0 complete)
Password: abroto[C3][B1]ar (3 of 984 complete)
...
NOTICE: ssh.mod: failed to connect, port 22 was not open on 10.11.224.7
NOTICE: ssh.mod: failed to connect, port 22 was not open on 10.11.224.7
NOTICE: ssh.mod: failed to connect, port 22 was not open on 10.11.224.7
```

Si revisamos el firewall en el servidor vemos que aparece una nueva regla en la cadena f2b-ssh bloqueando la IP correspondiente al equipo atacante. fail2ban se encargó de ejecutar la actionban: `iptables -I f2b-sshd 1 -s 10.11.224.112 -j REJECT --reject-with icmp-port-unreachable`

```
root@fail2ban:~# iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    f2b-sshd   tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 22

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain f2b-sshd (1 references)
num  target     prot opt source               destination
1    REJECT     all  --  10.11.224.112        0.0.0.0/0            reject-with icmp-port-
unreachable
2    RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

En el archivo de log de fail2ban `/var/log/fail2ban.log` podemos ver como el equipo fue 'baneado' y tras 600 segundos es 'unbanned':

```
$ cat /var/log/fail2ban.log
2022-04-25 09:01:08,347 fail2ban.server  [1805]: INFO    Changed logging target to
/var/log/fail2ban.log for Fail2ban v0.9.3
2022-04-25 09:01:08,348 fail2ban.database    [1805]: INFO    Connected to fail2ban persistent
database '/var/lib/fail2ban/fail2ban.sqlite3'
2022-04-25 09:01:08,422 fail2ban.database       [1805]: WARNING New database created. Version '2'
2022-04-25 09:01:08,424 fail2ban.jail           [1805]: INFO    Creating new jail 'sshd'
2022-04-25 09:01:08,463 fail2ban.jail           [1805]: INFO    Jail 'sshd' uses pyinotify
2022-04-25 09:01:08,475 fail2ban.filter         [1805]: INFO    Set jail log file encoding to
ANSI_X3.4-1968
2022-04-25 09:01:08,477 fail2ban.jail           [1805]: INFO    Initiated 'pyinotify' backend
2022-04-25 09:01:08,486 fail2ban.filter         [1805]: INFO    Set findtime = 600
2022-04-25 09:01:08,486 fail2ban.filter         [1805]: INFO    Set jail log file encoding to
ANSI_X3.4-1968
2022-04-25 09:01:08,486 fail2ban.filter         [1805]: INFO    Set maxRetry = 5
2022-04-25 09:01:08,494 fail2ban.filter         [1805]: INFO    Added logfile = /var/log/auth.log
2022-04-25 09:01:08,504 fail2ban.actions        [1805]: INFO    Set banTime = 600
2022-04-25 09:01:08,504 fail2ban.filter         [1805]: INFO    Set maxlines = 10
2022-04-25 09:01:08,601 fail2ban.server         [1805]: INFO    Jail sshd is not a JournalFilter
instance
2022-04-25 09:01:08,611 fail2ban.jail           [1805]: INFO    Jail 'sshd' started
2022-04-25 09:14:18,639 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:18,647 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:18,648 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:20,869 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:20,877 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:20,877 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:21,572 fail2ban.actions        [1805]: NOTICE  [sshd] Ban 10.11.224.112
2022-04-25 09:14:22,384 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:22,385 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:14:22,386 fail2ban.filter         [1805]: INFO    [sshd] Found 10.11.224.112
2022-04-25 09:24:22,443 fail2ban.actions        [1805]: NOTICE  [sshd] Unban 10.11.224.112
```

Es posible desbloquear manualmente un equipo baneado: primero buscamos el nombre exacto del jail, revisamos los equipos baneados y después mandamos la orden de `unban` sobre la IP de interés:

```
$ sudo fail2ban-client status
Status
|- Number of jail:      1
```

```
`- Jail list:     sshd
root@fail2ban:~# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 1
|  |- Total failed:     9
|  `- File list:  /var/log/auth.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   10.11.224.112
$ sudo fail2ban-client set sshd unbanip 10.11.224.112
10.11.224.112
```

maxretry y "Last message repeated N times"

Puede suceder que durante las pruebas de funcionamiento haya más de 5 fallos de contraseña y fail2ban no bloquea al equipo. Este 'error' a la hora de contar las entradas en auth.log puede ocurrir por la configuración del daemon syslog, si la reducción de mensajes repetidos está activada. Si el parámetro de syslog RepeatedMsgReduction está activado, un mensaje que aparezca varias veces en un período de tempo muy corto no se repetirá en el archivo de log, sino que aparecerá una línea indicando que el mensaje se repitió x veces:

```
May 15 00:25:27 ubuntu sshd[1477]: Failed password for root from 192.168.1.37 port 35924 ssh2
May 15 00:25:34 ubuntu sshd[1442]: message repeated 4 times: [ Failed password for root from 192.168.1.37 port 35896 ssh2]
```

Una solución es poner el parámetro de syslog RepeatedMsgReduction a off:

```
$ sudo nano /etc/rsyslog.conf

...
# Filter duplicated messages
$RepeatedMsgReduction off
...

$ sudo systemctl restart rsyslog.service
$ sudo systemctl restart fail2ban.service
```