MASTER OF SCIENCE
IN ENGINEERING

fail2ban - an effective use of iptables

Luca Haab

December 01, 2025

swissuniversities

fail2ban

Fail2Ban scans logs like `/var/log/auth.log` (*/var/log/messages in our case*)

- bans IP addresses conducting too many failed login attempts
- does this by updating system frewall rules
- comes out-of-the-box ready to read many standard logs
- is easily configured to read any log file of your choosing for any error you wish.

fail2ban - How it all began

In 2004 the original author was

- eager to learn a new programming language (*Python*)
- getting broadband access at home thus
  - time to get a linux box on the *Internet*
  - with ssh remote access

... and immediately after he had....

⚡ **Danger**

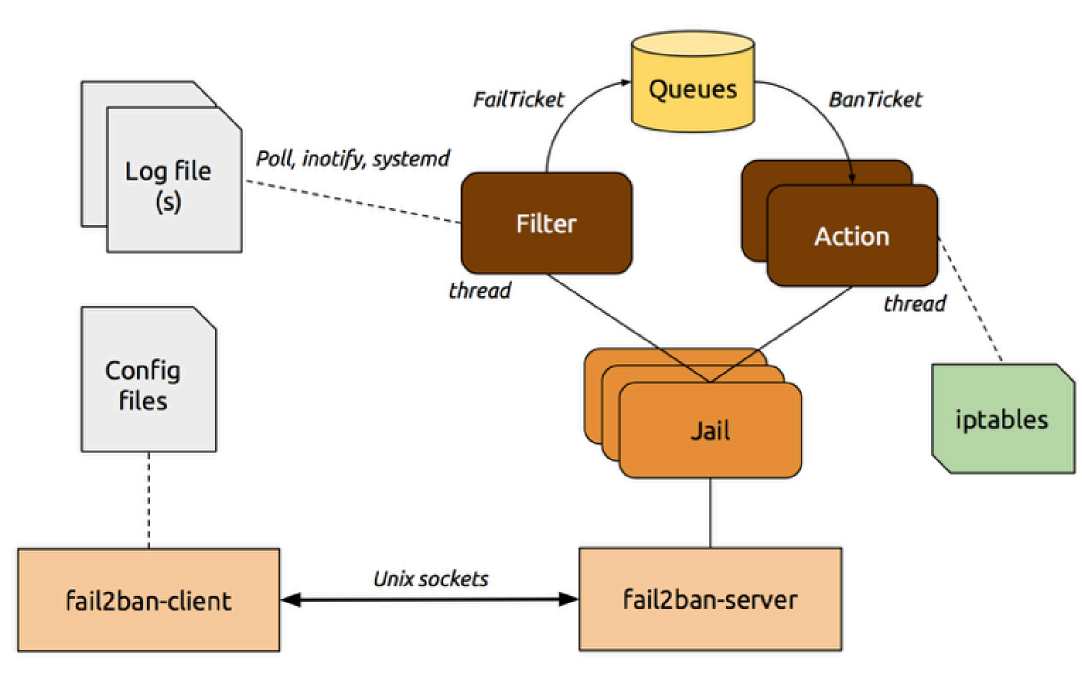
Script kiddies trying to log into his *Linux* box (that is, many (failed) log-in attempts over ssh).

```
/var/log/sshd.log
Jun 2 16:47:48 i sshd[1]: Failed password for X from 1.2.3.4 port 59926
Jun 2 16:47:49 i sshd[1]: Failed password for X from 1.2.3.4 port 59926
Jun 2 16:47:50 i sshd[1]: Failed password for X from 1.2.3.4 port 59926
```

fail2ban Architecture

`fail2ban` has 5 main parts:

- *Server* refers to the script `fail2ban-server` and it is the part that offers the service
- *Client* refers to the script `fail2ban-client`, which interacts with the server and config files
- *Jail* is a combination of one filter and one or several actions. `fail2ban` can handle several jails at the same time. *Jails* are defined in `/etc/fail2ban/jail.conf` or `jail.local`
- *Filter* defines a regular expression which must match a pattern corresponding to a *log-in* failure or any other expression. Filters are defined in `/etc/fail2ban/filters.d`
- *Action* contains several commands that are executed at different moments. Actions are defined in `/etc/fail2ban/action.d`



`fail2ban` *Filters*

Filters are `fail2ban` inputs and

- contain regular expressions matching offending patterns in logs
- are defined in `config/filter.d`
- will try to match and find `<HOST>`

Info

```
config/filter.d/sshd.conf
failregex = ^%(__prefix_line)s[iI]nvalid user .* from <HOST>\s*$
```

Tip

Use fail2ban-regex to test your regex

fail2ban *Actions*

Actions are the outputs of a match and

- result in several commands executed at different times
  - actionstart, actionstop
  - actioncheck
  - actionban, actionunban
- are defined in config/action.d
- actionban executed whenever a must be banned

Info

```
config/action.d/iptables.conf
actionban = <iptables> -I f2b-<name> 1 -s <ip> -j <blocktype>
```

`fail2ban` *Jail*

A *Jail* is a configuration unit that ties together:

- a *Filter* - the rules (usually regex patterns) for detecting unwanted behavior in log files
- an *Action* - what `fail2ban` should do when a rule is triggered (e.g., add a firewall ban)
- *Jail* settings - such as log paths, ban time, find time, and max retries.

Conclusion

A jail = filter + action + settings.

`fail2ban` *Jail*

⚠ Example

`/etc/fail2ban/jail.conf` or `jail.local`

```
[sshd]
enabled = true
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 1h
action = iptables-multiport
```

How fail2ban works

Example with sshd: on our NanoPi, `sshd` daemon logs authentications in `/var/log/messages`

```
Jan  1 00:01:28 buildroot auth.info sshd[201]: Failed password for foo from 192.168.0.1 port 55032 ssh2
Jan  1 00:01:35 buildroot auth.info sshd[201]: Failed password for foo from 192.168.0.1 port 55032 ssh2
Jan  1 00:01:37 buildroot auth.info sshd[201]: Failed password for foo from 192.168.0.1 port 55032 ssh2
Jan  1 00:01:37 buildroot auth.info sshd[201]: Connection closed by authenticating user foo 192.168.0.1 port 55032 [preauth]
Jan  1 00:02:08 buildroot auth.info sshd[224]: Accepted password for foo from 192.168.0.1 port 55197 ssh2
```

- 1-3 lines: authentication of user *foo* is incorrect
- 4th line: user *foo* is disconnected
- 5th line: authentication of user *foo* is correct.

How `fail2ban` works (previous slide):

- *Jail* configuration results in scans of `/var/log/messages` and `sshd` lines
- *Filter* detects the incorrect authentications and, if a threshold is reached, an action is started
- *Action* configures `iptables`

fail2ban Configuration

- By default, fail2ban keeps all configuration files in /etc/fail2ban/ directory.
- Configurations are contained in *.conf files. It is recommended not to modify these files but override these configurations by creating new configuration files *.local inside the /etc/fail2ban directory.
- fail2ban has four configuration file types in /etc/fail2ban directory:

```
jail.conf         Jails defining combinations of Filters with Actions.
fail2ban.conf     fail2ban global configuration (such as logging)
filter.d/*.conf   Filters specifying how to detect authentication failures
action.d/*.conf   Actions  defining  the  commands for banning and unbanning of IP address
```

fail2ban Configuration

General configuration:

```
cat /etc/fail2ban/fail2ban.conf
# Values: [CRITICAL | ERROR | WARNING | NOTICE | INFO
| DEBUG]
loglevel = DEBUG        // Important for debug
# Values: [ STDOUT | STDERR | SYSLOG | SYSOUT |
FILE ]  Default: STDERR
logtarget = /var/log/fail2ban.log   // Important for
debug

socket = /var/run/fail2ban/fail2ban.sock
pidfile = /var/run/fail2ban/fail2ban.pid
dbfile = /var/lib/fail2ban/fail2ban.sqlite3
dbpurgeage = 1d
```

Preconfigured filters:

```
ls /etc/fail2ban/filter.d/
apache-auth.conf
dropbear.conf
sshd.conf
…

ls /etc/fail2ban/action.d/
iptables.conf
```

fail2ban: how to check the rules (*regex*)

The `fail2ban-regex` command can be used to check regular expressions.

```
fail2ban-regex -v aLogFile aFilter
```

For instance:

```
fail2ban-regex -vvvvv /var/log/messages /etc/
fail2ban/filter.d/sshd.conf
```

or

```
fail2ban-regex -v 'Sep 29 17:15:02 Failed password
for user from 127.0.0.1 port 20000 ssh1: ruser from
1.2.3.4' '^ Failed \S+ for .* from <HOST>( port \d*)?
( ssh\d+)?(: ruser .*)?$'
```

```
fail2ban-regex -vvvvv /var/log/messages "Failed password for [-\w]+.* from <HOST>”
Failregex: 5 total
|-  #) [# of hits] regular expression
|   1) [3] Failed password for [-\w]+ from
<HOST>
|      192.168.0.4  Sat Jan 01 00:04:20 2020
|      192.168.0.4  Sat Jan 01 00:04:20 2020
|      192.168.0.4  Sat Jan 01 00:04:21 2020
`-
Ignoreregex: 0 total
Date template hits:
|- [# of hits] date format
|  [1073] {^LN-BEG}(?:DAY )?MON Day %k:Minute:Second(?:\.Microseconds)?(?:
ExYear)?
```

The term `[-\w]+` identifies the login name. It means matches 1 or more occurrences of the characters `-,a-z,A-Z,0-9` Every line of `fail2ban-regex` must have the term ‘“’, which identifies the IP address.

fail2ban: making it permanent

Warning

The code below is just a short example - not a proper implementation.

```
#!/bin/sh
umask 077
start() {
        printf "Starting fail2ban: "
        mkdir /var/run/fail2ban
        /usr/bin/fail2ban-client start
        touch /var/lock/fail2ban
        echo "OK"
}
stop() {
        printf "Stopping fail2ban: "
        /usr/bin/fail2ban-client stop
        rm -rf /var/run/fail2ban
        echo "OK"
}
restart() {
        stop
        start
}
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        restart
        ;;
  *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

In order to make it permanent, one needs to add it to `init` system. That is, one would need to create a dedicated file `S45fail2ban` in the `/etc/init.d` folder

Tip

Do remember to change its attributes (e.g. `chmod ugo+rx`)

`fail2ban`: let’s make use of it!

Resources

- `fail2ban` documentation: https://github.com/fail2ban/fail2ban/wiki and repository https://github.com/fail2ban/fail2ban
- `fail2ban` presentation by the authour, *Linux Fribourg Seminar*: https://gitlab.forge.hefr.ch/fribourg-linux-seminar/seminars/-/raw/master/19.11_handout_17th_seminar/03_fail2ban.pdf