fail2ban

CentOS 7 / Red Hat:

```
yum install fail2ban
```

Ubuntu / Debian:

```
apt install fail2ban
```

CentOS 6:

Добавляем репозиторий:

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

Устанавливаем пакет:

```
yum install fail2ban
```

Для запуска службы вводим следующие команды:

```
systemctl enable fail2ban
systemctl start fail2ban
```

* *Для старых систем без systemd это будут команды*

```
chkconfig fail2ban on
update-rc.d fail2ban defaults
service fail2ban start.
```

Базовая настройка

Процесс настройки fail2ban не зависит от дистрибутива Linux. Основной конфигурационный файл находится по пути **/etc/fail2ban/jail.conf**. Однако, его не рекомендуется менять и для настройки используют подключаемые файлы из

каталога **/etc/fail2ban/jail.d**.

Для начала создаем первый файл, в котором будут храниться настройки по умолчанию:

```
nano /etc/fail2ban/jail.d/default.conf
```

Приведем его к виду:

```
[DEFAULT]
maxretry = 4
findtime = 480
bantime = 720
action = firewallcmd-ipset
ignoreip = 127.0.0.1/8
```

** где:*

- ***maxretry –*** *количество действий, которые разрешено совершить до бана.*
- ***findtime –*** *время в секундах, в течение которого учитывается maxretry;*
- ***bantime –*** *время, на которое будет блокироваться IP-адрес;*
- ***action*** *– действия, которое будет выполняться, если Fail2ban обнаружит активность, соответствующую критериям поиска;*
- ***ignoreip –*** *игнорировать защиту, если запросы приходят с перечисленных адресов.*

** В данном примере, если в течение 8 минут (**480**) будет найдено 5 строк (**maxretry = 4**), содержащих критерий фильтра, Fail2ban заблокирует IP-адрес, с которого идет подключение на 12 минут (**720**);*
** В секции [DEFAULT] хранятся общие настройки для всех правил. Каждую из настроек можно переопределить при конфигурировании самого правила.*

Настройка правил

Для нового правила необходимо создать конфигурационный файл в каталоге **/etc/fail2ban/jail.d**, например:

```
nano /etc/fail2ban/jail.d/service.conf
```

```
[ssh]
enabled = true
port = ssh
filter = sshd
action = iptables[name=sshd, port=ssh, protocol=tcp]
logpath = /var/log/auth.log
maxretry = 10
findtime = 600
```

* *где:*

- ***ssh –*** *название для правила;*
- ***enabled*** *позволяет быстро включать (true) или отключать (false) правило;*
- ***port –*** *порт целевого сервиса. Принимается буквенное или цифирное обозначение;*
- ***filter –*** *фильтр (критерий поиска), который будет использоваться для поиска подозрительных действий. По сути, это имя файла из каталога* ***/etc/fail2ban/filter.d****без .conf на конце;*
- ***action –*** *действие, совершаемое в случае срабатывания правила. В квадратных скобках указаны название для правила, сетевой порт и протокол для блокирования;*
- ***logpath –*** *расположение лог-файла, в котором фильтр будет искать подозрительную активность на основе описанных критериев.*

* *обратите внимание, что мы переопределили параметры по умолчанию* ***maxretry****,* **findtime** *и* ***action****.*

Чтобы изменения вступили в силу, перезапускаем сервис:

systemctl restart fail2ban

* *в старых версиях*

service fail2ban restart.

Действия и фильтры

Файлы с настройкой действий находятся в каталоге **/etc/fail2ban/action.d**. Чтобы блокировать адрес, Fail2ban создает правило в брандмауэре netfilter. Для этого, чаще всего, используются утилиты iptables или firewall-cmd. Последняя применяется в последних версиях CentOS / Red Hat / Fedora. iptables более универсальная и может использоваться, почти, во всех системах Linux.

Остановимся на описании самых используемых действий:

- iptables – создание простого правила в netfilter с помощью одноименной утилиты;
- iptables-multiport – использование модуля multiports, позволяющий добавлять диапазоны портов для блокировки;
- iptables-ipset – использование ipset для придания более лаконичного вида правилам;
- iptables-allports – блокирует для адреса все порты;
- firewallcmd-new – создание простого правила в netfilter с помощью firewall-cmd;
- firewallcmd-ipset – добавляет правила с помощью утилиты firewall-cmd, используя ipset;
- firewallcmd-rich-rules – создает rich-rules при помощи firewall-cmd.

Подробнее, как создаются правила в netfilter при помощи iptables и firewalld.

Фильтры, в основном, представляют набор регулярных выражений для поиска ключевых слов в log-файлах. Они находятся в каталоге **/etc/fail2ban/filter.d**.

Для создания и настройки своих фильтров, можно использовать

имеющиеся файлы в качестве шпаргалки.

Примеры правил

В данных примерах блокировка IP-адреса будет происходить на 12 минут после 4-х попыток ввода пароля в течение 8 минут. Эти параметры берутся из настроек [DEFAULT]. Если их нужно переопределить, просто добавляем их при описании правила.

SSH

CentOS

```
vi /etc/fail2ban/jail.d/ssh.conf
```

```
[ssh]
enabled = true
port = ssh
filter = sshd
action = firewallcmd-new[name=sshd]
logpath = /var/log/secure
```

Ubuntu

```
vi /etc/fail2ban/jail.d/ssh.conf
```

```
[ssh]
enabled = true
port = ssh
filter = sshd
action = iptables[name=sshd]
logpath = /var/log/auth.log
```

Asterisk

```
vi /etc/fail2ban/jail.d/asterisk.conf
```

```
[asterisk]
enabled = true
filter = asterisk
action = iptables-allports[name=asterisk, protocol=all]
logpath = /var/log/asterisk/messages
```

NGINX

```
vi /etc/fail2ban/jail.d/nginx.conf
```

```
[nginx]
enabled = true
port = http,https
filter = nginx-http-auth
action = iptables-multiport[name=nginx, port="http,https",
protocol=tcp]
logpath = /var/log/nginx/error.log
```

NGINX DDoS (req limit)

Данное правило поможет защитить веб-сервер nginx от DDoS-атак. В некоторых сборках, для данного правило может не оказаться готового фильтра, поэтому в данном примере, мы его создадим вручную.

Создаем фильтр:

```
vi /etc/fail2ban/filter.d/nginx-limit-req.conf
```

```
[Definition]

ngx_limit_req_zones = [^"]+
failregex = ^\s*\[error\] \d+#\d+: \*\d+ limiting requests,
excess: [\d\.]+ by zone "(?:%(ngx_limit_req_zones)s)", client:
<HOST>
ignoreregex =
```

Создаем правило в Fail2ban:

```
vi /etc/fail2ban/jail.d/nginx-ddos.conf

[nginx-ddos]
enabled = true
port = http,https
filter = nginx-limit-req
action = iptables-multiport[name=nginxddos, port="http,https",
protocol=tcp]
logpath = /var/log/nginx/error.log
```

OWNCLOWD

```
vi /etc/fail2ban/filter.d/owncloud.conf

[Definition]
failregex={.*Login failed: \'.*\' \(Remote IP: \'<HOST>\'\)"}
ignoreregex =
```

```
vi /etc/fail2ban/jail.d/owncloud.conf

[owncloud]
enabled = true
filter = owncloud
ignoreip  =  127.0.0.1/8  192.168.1.47  194.44.219.161
194.44.219.163 194.44.219.164
#action                    =                    smeserver-
iptables[port="$port",protocol=tcp,bantime=$bantime]
logpath = /var/www/html/owncloud/data/owncloud.log
maxretry = 3
port = 80,443
bantime = 10800
protocol = tcp
```

WORDPRESS

```
vi /etc/fail2ban/filter.d/wp-login.conf

[Definition]
failregex = ^<HOST> .* "POST .*wp-login.php
            ^<HOST> .* "POST .*xmlrpc.php
ignoreregex =
```

```
vi /etc/fail2ban/jail.d/wp-login.conf

[wp-login]
enabled = true
ignoreip = 127.0.0.1/8 192.168.0.0/24
action = iptables-multiport[name=WP, port="http,https"
protocol=tcp]
# включаем отправку оповещения на почту, если вам это
необходимо
#   sendmail[name=wp-login,   dest=zeroxzed@gmail.com,
sender=fail2ban@serveradmin.ru]
filter = wp-login
logpath = /var/log/nginx/access.log
bantime = 10800
maxretry = 3
```

Проверяем:

```
fail2ban-regex             /var/log/nginx/access.log
/etc/fail2ban/filter.d/wp-login.conf

Results
=======

Failregex: 4 total
|- #) [# of hits] regular expression
| 1) [3] ^<HOST> .* "POST .*wp-login.php
| 2) [1] ^<HOST> .* "POST .*xmlrpc.php
`-
Ignoreregex: 0 total

Date template hits:
|- [# of hits] date format
|      [302]      Day(?P<_sep>[-/])MON(?P=_sep)Year[
:]?24hour:Minute:Second(?:\.Microseconds)?(?: Zone offset)?
`-
Lines: 302 lines, 0 ignored, 4 matched, 298 missed
[processed in 0.06 sec]
```

После настройки не забываем перезапустить fail2ban:

```
systemctl restart fail2ban
```

SEAFILE

```
# vi /etc/fail2ban/jail.d/seafile.conf

# All standard jails are in the file configuration located
# /etc/fail2ban/jail.conf

# Warning you may override any other parameter (e.g. banaction,
# action, port, logpath, etc) in that section within jail.local

# Change logpath with your file log used by seafile (e.g. seahub.log)
# Also you can change the max retry var (3 attemps = 1 line written in the
# seafile log)
# So with this maxrety to 1, the user can try 3 times before his IP is banned

[seafile]

enabled = true
port = http,https
filter = seafile-auth
logpath = /home/www/seafile/logs/seahub.log
maxretry = 3
```

```
# vi /etc/fail2ban/filter.d/seafile-auth.conf

# Fail2Ban filter for seafile
#

[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
before = common.conf

[Definition]
```

```
_daemon = seaf-server

failregex = Login attempt limit reached.*, ip: <HOST>

ignoreregex =

# DEV Notes:
#
#   pattern   :   2015-10-20   15:20:32,402   [WARNING]
seahub.auth.views:155 login Login attempt limit reached,
username: <user>, ip: 1.2.3.4, attemps: 3
#  2015-10-20  17:04:32,235  [WARNING]  seahub.auth.views:163
login Login attempt limit reached, ip: 1.2.3.4, attempts: 3
```

Перечитываем конфиг и проверяем:

```
# fail2ban-client reload
# fail2ban-client status
#    fail2ban-regex    /home/www/seafile/logs/seahub.log
/etc/fail2ban/filter.d/seafile-auth.conf
```

Работа со списком заблокированных адресов

Просмотр

Получить статистику заблокированных адресов можно следующей командой:

```
fail2ban-client status <имя правила>
```

Получить список правил можно командой:

```
fail2ban-client status
```

При наличие заблокированных IP-адресов мы увидим, примерно, следующее:

```
`- action
   |- Currently banned: 2
```

```
   |  `- IP list:       31.207.47.55 10.212.245.29
```

С помощью iptables:

```
iptables -L -n --line
```

С помощью firewall-cmd:

```
firewall-cmd --direct --get-all-rules
```

Удаление

Средствами fail2ban:

Для удаление адреса из списка вводим:

```
fail2ban-client set <имя правила> unbanip <IP-адрес>
```

например:

```
fail2ban-client set ssh unbanip 31.207.47.55
```

С помощью iptables:

```
iptables -D <цепочка правил> -s IP-адрес
```

например:

```
iptables -D fail2ban-ssh -s 10.212.245.29
```

С помощью firewall-cmd:

```
firewall-cmd --direct --permanent --remove-rule <правило>
```

например:

```
firewall-cmd --direct --permanent --remove-rule ipv4 filter
f2b-sshd 0 -s 188.134.7.221
```

После необходимо перечитать правила:

```
firewall-cmd --reload
```

https://www.dmosk.ru/instruktions.php?object=fail2ban