Fail2Ban

Installation and main configuration

Install Fail2Ban on Debian-based systems by running

```
sudo apt -y install fail2ban
```

Create `/etc/fail2ban/jail.local` and edit it.

Add the following configuration or replace existing lines

```
[DEFAULT]
# the IP address range we want to ignore
ignoreip = 127.0.0.1/8 [LAN SEGMENT]
# who to send e-mail to
destemail = [your e-mail]
# who is the email from
sender = [your e-mail]
mta = mail
#Enable Email alerts
action = %(action_mwl)s
```

That's pretty much it, now you can create jails for all the applications you use.

Checking for installed jails

```
sudo fail2ban-client status
```

Checking jail for banned IPs

```
sudo fail2ban-client status jailname
```

Unban client

```
fail2ban-client set <jail> unbanip <IP>
```

Script to check for Banned IPs

If you want to save yourself the trouble of checking every Fail2Ban jail yourself, you can use the following script, which moreover, runs a whois check, to tell you to whom the IP belongs. With some tweaks, you can also include this script in your monitoring (e.g. Nagios, Icinga).

```
#!/bin/bash
if [ $( id -u ) -ne 0 ]; then echo "$0 needs root to run"; exit 1; fi

LOG=fail2ban-whois.log
BOLD=$( tput bold )
SGR0=$( tput sgr0 )
SMUL=$( tput smul )
RMUL=$( tput rmul )

LIST=0
if [ $( echo $* | grep '\-h' ) ]; then echo -e "\n$0 [-h] help\n$0 [-s] output short listing
(default)\n$0 [-l] output long listing\n"; exit 0; fi
if [ $( echo $* | grep '\-s' ) ]; then LIST=0; fi
if [ $( echo $* | grep '\-l' ) ]; then LIST=1; fi
echo "$( date +%F\ %T ) running ${BOLD}$0${SGR0} and logging to ${SMUL}$LOG${RMUL}..." | tee -
a $LOG
for j in $( fail2ban-client status | grep --color=never "Jail list:" | sed -e
's/.*:\t*\(.*\)/\1/g' -e 's/,//g' ); do
  for ip in $( fail2ban-client status $j | grep "IP list:" ); do
    if [ "$( echo $ip | grep -E [0-9a-fA-F\.:]{4} )" ]; then
      echo -e "[JAIL] $j [IP] ${BOLD}$ip${SGR0} [WHOIS]\n$( whois $ip | \
        sed '0,/descr:/{s/descr:/descr_1:/}' | \
        grep -E '^((% Abuse contact for|abuse-mailbox:)|(inetnum:|CIDR:|IPv4
Address)|([nN]et[nN]ame:|Service Name|ownerid:)|([cC]ountry:|#
KOREAN\(UTF8\))|(descr_1:|Organization Name|owner:))' | \
        sed -e 's/^ *//g' -e 's/  */ /g' | sed -e 's/ - /-/g' \
            -e 's/% Abuse contact for.*is '\''\(.*\)'\''/abuse-c: \1/g' -e "s/\(abuse-
c:\|abuse-mailbox:\)/${SMUL}abuse:${RMUL}/g" \
            -e "s/\(inetnum:\|CIDR:\|IPv4 Address :\)/${SMUL}inetnum:${RMUL}/g" -e
"s/\([nN]et[nN]ame:\|Service Name :\|ownerid:\)/${SMUL}netname:${RMUL}/g" \
            -e "s/# KOREAN(UTF8)/${SMUL}country:${RMUL} ${BOLD}KR${SGR0}/" -e "s/[cC]ountry:
\([a-zA-Z][a-zA-Z]\)\(.*\)/${SMUL}country:${RMUL} ${BOLD}\U\1\E${SGR0}\2/g" \
```

```
            -e "s/\(Organization Name :\|owner:\)/${SMUL}descr:${RMUL}/g" | \
        sort -f | uniq -i | sed "s/descr_1:/${SMUL}descr:${RMUL}/g"
      )" | ( [ $LIST -eq 0 ] && sed ':a;N;$!ba;s/\(\n\|\r\n\)/ /g' || cat )
    fi
  done
done | tee -a $LOG
```

You can download it here, or for more convenience, just run the below command

```
curl -f TODO -o ${HOME}/fail2banwhois.sh && chmod +x ${HOME}/fail2banwhois.sh
```

You have to run the script with Sudo privileges. It will show you all currently banned IPs. The output looks like this (IP's are masked):

```
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: info@starcrecium.com country: RU inetnum:
100.100.100.100.0-100.100.100.100.255 netname: CY-STARCRECIUM descr: HOSTWAY route object
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] country: GB country: US inetnum:
100.100.100.100/13, 100.100.100.100.0.0/13, 100.100.100.100/12 inetnum: 100.100.100.100/15
netname: AMAZON-LHR netname: AT-88-Z
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@ito.gov.ir country: IR inetnum:
100.100.100.100-100.100.100.100 netname: TBZ-MED descr: Tabriz University of Medical Sciences
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@xyz.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: Science-LAN descr: Some Science Institution
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@versatel.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: DE-VERSATEL-20080807 descr: VT-Customer
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@xyz.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: Science-LAN descr: Some Science Institution
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@cdn77.com country: AT country: GB
inetnum: 100.100.100.100-100.100.100.100 netname: CDN77-VIE descr: CDN77-VIE POP
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@contabo.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: CONTABO
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@contabo.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: CONTABO descr: Contabo GmbH
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@pindc.ru country: RU inetnum:
100.100.100.100-100.100.100.100 netname: PINDC-public-vlans descr: PIN DC
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@contabo.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: CONTABO
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@telekom.de country: DE inetnum:
100.100.100.100-100.100.100.100 netname: DTAG-DIAL28 descr: Deutsche Telekom AG
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: ipas@cnnic.cn country: CN inetnum:
```

100.100.100.100-100.100.100.100 netname: TencentCloud descr: Tencent cloud computing (Beijing) Co., Ltd.
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: qcloud_net_duty@tencent.com country: AU country: CN country: ZZ inetnum: 100.100.100.100-100.100.100.100 inetnum: 100.100.100.100/16 netname: APNIC netname: TENCENT-CN descr: Tencent Cloud Computing (Beijing) Co., Ltd
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] country: US inetnum: 100.100.100.100/12, 100.100.100.100/14, 100.100.100.100/14, 100.100.100.100/17, 100.100.100.100/12, 100.100.100.100/15, 100.100.100.100/13, 100.100.100.100/16 netname: MSFT
[JAIL] nginx-dos [IP] 100.100.100.100 [WHOIS] abuse: abuse@versatel.de country: DE inetnum: 100.100.100.100-100.100.100.100 netname: VT-Customer-POOL descr: Versatel Deutschland

Filters

You can customize, improve the filters Fail2Ban uses. You can find the filters in `/etc/fail2ban/filter.d/`. Below is a **small** collection of filters for the services, I have already covered.

`nginx-http-auth.conf`

```
[Definition]



failregex = ^ \[error\] \d+#\d+: \*\d+ user "\S+":? (password mismatch|was not found in ".*"),
client: <HOST>, server: \S+, request: "\S+ \S+ HTTP/\d+\.\d+", host: "\S+"\s*$
            ^ \[error\] \d+#\d+: \*\d+ no user/password was provided for basic authentication,
client: <HOST>, server: \S+, request: "\S+ \S+ HTTP/\d+\.\d+", host: "\S+"\s*$

ignoreregex =
```

`nginx-badbots.conf`

```
sudo cp apache-badbots.conf nginx-badbots.conf
```

`nginx-noscript`

```
[Definition]

failregex = ^<HOST> -.*GET.*(\.php|\.asp|\.exe|\.pl|\.cgi|\.scgi)

ignoreregex =
```

`nginx-noproxy`

```
[Definition]

failregex = ^<HOST> -.*GET http.*

ignoreregex =
```

MongoDB

`mongo-auth.conf`

```
[INCLUDES]

before = common.conf

[Definition]

_daemon = mongodb

failregex = ^.*[aA]uthentication [fF]ail(ed|ure) for \w+ on \w+ from client <HOST>:[0-9].*|$

ignoreregex =

# Author: luiseok (https://github.com/luiseok)
```

SSH

Create an SSH jail by creating the file `/etc/fail2ban/jail.d/ssh.local`. Paste in the following config, adapt it to your needs

```
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
findtime = 300
bantime = 3600
ignoreip = 127.0.0.1
```

You can only use the ban action ufw, if you have it installed, configured, and enabled.

Start and reload SSHD, and then add your jail

```
sudo fail2ban-client start
sudo fail2ban-client reload
sudo fail2ban-client add sshd
```

NGINX

We can use Fail2ban to

1. Block too many HTTP Authentication attempts
2. Prevent clients from searching for scripts
3. Stop malicious requests from bots
4. Ban clients trying to use NGINX as an open proxy

1. Block too many HTTP Authentication attempts

Create the file `/etc/fail2ban/jail.d/nginx-http.local`. Add the following content:

```
[nginx-http-auth]

enabled  = true
filter   = nginx-http-auth
port     = http,https
logpath  = /var/log/nginx/error.log
```

Keep in mind, that you change the error log, for every site you use. E.G., you only want to monitor failed authentication attempts for your custom log path, `error-bookstack.log`.

2. Prevent clients from searching for scripts

> Only use this function, if you have no scripts and no PHP on your Website!

Create the file `/etc/fail2ban/jail.d/nginx-noscript.local`. Add the following content:

```
[nginx-noscript]

enabled  = true
port     = http,https
filter   = nginx-noscript
logpath  = /var/log/nginx/access.log
maxretry = 6
```

Again, you might want to/have to change the log path.

3. Stop malicious requests from Bots

Create the file `/etc/fail2ban/jail.d/nginx-nobots.local`. Add the following content:

```
[nginx-badbots]

enabled  = true
port     = http,https
filter   = nginx-badbots
logpath  = /var/log/nginx/access.log
maxretry = 2
```

Again, you might want to/have to change the log path.

4.Ban clients trying to use NGINX as open proxy

Create the file `/etc/fail2ban/jail.d/nginx-noproxy.local`. Add the following content:

```
[nginx-noproxy]

enabled  = true
port     = http,https
filter   = nginx-noproxy
logpath  = /var/log/nginx/access.log
maxretry = 2
```

In this case, you should actually leave it at the global access logfile.

Start and reload SSHD, and then add your jail

```
sudo fail2ban-client start
sudo fail2ban-client reload
sudo fail2ban-client add <yournginxjail>
```

MySQL

Create an SSH jail by creating the file `/etc/fail2ban/jail.d/mysql.local`. Paste in the following config, adapt it to your needs

```
[mysqld]
log = /var/log/mysql/access.log
log_error = /var/log/mysql/error.log
log_warnings = 2
```

Start and reload SSHD, and then add your jail

```
sudo fail2ban-client start
sudo fail2ban-client reload
sudo fail2ban-client add mysql
```

MongoDB

Create an SSH jail by creating the file `/etc/fail2ban/jail.d/mongodb.local`. Paste in the following config, adapt it to your needs

```
[mongo-auth]

enabled  = true
filter  = mongo-auth
logpath  = /var/log/mongodb/mongod.log
maxretry = 3
port  = 27017
banaction = iptables-multiport[name="mongo", port="27017"]

bantime = 86400
findtime = 300
```

Start and reload SSHD, and then add your jail

```
sudo fail2ban-client start
sudo fail2ban-client reload
sudo fail2ban-client add mongodb
```