Fail2Ban

Fail2ban scans log files (e.g. /var/log/apache/error_log) and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc. Generally Fail2Ban is then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other action (e.g. sending an email) could also be configured

- How to Install and Configure Fail2Ban to work with NGINX Proxy Manager

How to Install and Configure Fail2Ban to work with NGINX Proxy Manager

In this post we'll look at how to get Fail2Ban set up to work with NGINX Proxy Manager. Additionally, we'll add a configuration that will show us the real (instead of the CloudFlare) IP addresses of anyone who is being a bad actor on your network and Fail2Ban will kick in and ban them on your network by communicating with CloudFlare and having CloudFlare ban them on their end for a determined period of time.

So. Let's jump into this. The following is a list of instructions without much explanation as it is the steps I went through (and later recounted to create this tutorial) to get Fail2Ban set up on my Docker setup with Nginx Proxy Manager.

Original source that got me started.

`cd /home`

`mkdir fail2ban`

`cd fail2ban`

Create the docker-compose.yml file

`nano docker-compose.yml`

```
version: "3.7"
services:
  fail2ban:
    image: crazymax/fail2ban:latest
    container_name: fail2ban_docker-pi
    network_mode: "host"
    cap_add:
      - NET_ADMIN
      - NET_RAW
    volumes:
```

```
      - "/home/docker/fail2ban/data:/data"
      - "/var/log/auth.log:/var/log/auth.log:ro"
      - "/home/docker/nginx-proxy-manager/data/logs/:/log/npm/:ro"
    restart: always
    environment:
  - TZ=Europe/Paris
  - F2B_LOG_TARGET=STDOUT
  - F2B_LOG_LEVEL=INFO
  - F2B_DB_PURGE_AGE=1d
  - SSMTP_HOST=smtp.example.com
  - SSMTP_PORT=587
  - SSMTP_HOSTNAME=example.com
  - SSMTP_USER=smtp@example.com
  - SSMTP_PASSWORD=
  - SSMTP_TLS=YES
```

Save and exit.

Start the docker container: `docker-compose up -d`

This should create the folder structure you need.

You're going to create 3 files:

- /home/docker/fail2ban/data/filter.d/npm-docker.conf
- /home/docker/fail2ban/data/jail.d/npm-docker.local
- /home/docker/fail2ban/data/action.d/cloudflare-apiv4.conf

npm-docker.conf

```
[INCLUDES]

[Definition]

failregex = ^<HOST>.+" (4\d\d|3\d\d) (\d\d\d|\d) .+$
            ^.+ 4\d\d \d\d\d - .+ \[Client <HOST>\] \[Length .+\] ".+" .+$
```

Save and exit.

npm-docker.local

```
[npm-docker]
enabled = true
ignoreip = 127.0.0.1/8 192.168.1.0/24 your.home.ip.address
action = cloudflare-apiv4
chain = INPUT
logpath = /log/npm/default-host_access.log
          /log/npm/proxy-host-*_access.log
          /log/npm/proxy-host-*_error.log
maxretry = 1
bantime  = -1
findtime = 86400
```

A couple of things:

Modify the 192.168.1.0/24 to match your network.

It's a good idea to add your home IP to the ignoreip line. Just add a space after the /24 and then enter your home's IP address.

Change the maxretry, bantime, and findtime to suit your needs.

> More information about the different settings can be found here

Save and exit.

cloudflare-apiv4.conf

Paste the following in:

```
#
# Author: Gilbn from https://technicalramblings.com
# Adapted Source:
https://github.com/fail2ban/fail2ban/blob/master/config/action.d/cloudflare.conf and
https://guides.wp-bullet.com/integrate-fail2ban-cloudflare-api-v4-guide/
#
# To get your Cloudflare API key: https://dash.cloudflare.com/profile use the Global API Key
#

[Definition]
```

```
# Option:  actionstart
# Notes.:  command executed once at the start of Fail2Ban.
# Values:  CMD
#
actionstart =

# Option:  actionstop
# Notes.:  command executed once at the end of Fail2Ban
# Values:  CMD
#
actionstop =

# Option:  actioncheck
# Notes.:  command executed once before each actionban command
# Values:  CMD
#
actioncheck =

# Option:  actionban
# Notes.:  command executed when banning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:      IP address
#            number of failures
#            unix timestamp of the ban time
# Values:  CMD

actionban = curl -s -X POST
"https://api.cloudflare.com/client/v4/user/firewall/access_rules/rules" \
            -H "X-Auth-Email: <cfuser>" \
            -H "X-Auth-Key: <cftoken>" \
            -H "Content-Type: application/json" \
            --data
'{"mode":"block","configuration":{"target":"ip","value":"<ip>"},"notes":"Fail2ban <name>"}'

# Option:  actionunban
# Notes.:  command executed when unbanning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:      IP address
#            number of failures
#            unix timestamp of the ban time
```

```
# Values:  CMD
#

actionunban = curl -s -X DELETE
"https://api.cloudflare.com/client/v4/user/firewall/access_rules/rules/$( \
              curl -s -X GET
"https://api.cloudflare.com/client/v4/user/firewall/access_rules/rules?mode=block&configuratio
n_target=ip&configuration_value=<ip>&page=1&per_page=1&match=all" \
             -H "X-Auth-Email: <cfuser>" \
             -H "X-Auth-Key: <cftoken>" \
             -H "Content-Type: application/json" | awk -F"[,:}]"
'{for(i=1;i<=NF;i++){if($i~/'id'\042/){print $(i+1);}}}' | tr -d '"' | sed -e 's/^[ \t]*//' |
head -n 1)" \
             -H "X-Auth-Email: <cfuser>" \
             -H "X-Auth-Key: <cftoken>" \
             -H "Content-Type: application/json"

[Init]

# Name of the jail in your jail.local file.  default = [jail name]
name = npm-docker

# Option: cfuser
# Notes.: Replaces <cfuser> in actionban and actionunban with cfuser value below
# Values: Your CloudFlare user account

cfuser = davidnburgess@gmail.com

# Option: cftoken (Global API Key)
# Notes.: Replaces <cftoken> in actionban and actionunban with cftoken value below
# Values: Your CloudFlare API key
cftoken = 1234567890abcdefghijklmnop
```

Be sure to replace `cfuser` with your actual CloudFlare account email address.

Also, change the `cftoken` to your CloudFlare Global API Key. To find this, login to your CloudFlare account and click the icon in the top right of the page. Then click "My Profile". Open the API Tokens on this page. Find the "Global API Key" and click "View".

Paste that token in for the cftoken.

Save and exit.

Next, we need to figure out where our Nginx Proxy Manager nginx.conf file is. To find all of our nginx.conf files, we'll want to go to our root directory and run a search command:

`cd /`

`find . -name nginx.conf`

That should show us something like this:

```
./root/Guacamole/guacamole-docker-compose/nginx/nginx.conf
./var/lib/docker/overlay2/9f9668f541949895adb713c3ab3965b6b376fec1f915dfb97bae4220ee3c5730/diff/etc/nginx/nginx.conf
./var/lib/docker/overlay2/fc5c253437720e420cb096d2832a4bda6413772c8531da3de0e8e6b67596bc81/diff/tmp/openresty/build/opm-0.0.5/web/conf/nginx.conf
./var/lib/docker/overlay2/fc5c253437720e420cb096d2832a4bda6413772c8531da3de0e8e6b67596bc81/diff/tmp/openresty/build/nginx-1.19.3/conf/nginx.conf
./var/lib/docker/overlay2/fc5c253437720e420cb096d2832a4bda6413772c8531da3de0e8e6b67596bc81/diff/tmp/openresty/bundle/opm-0.0.5/web/conf/nginx.conf
./var/lib/docker/overlay2/fc5c253437720e420cb096d2832a4bda6413772c8531da3de0e8e6b67596bc81/diff/tmp/openresty/bundle/nginx-1.19.3/conf/nginx.conf
./var/lib/docker/overlay2/793d7fa22cb7d1e29fa36ec6701d318f5b2320dfa592edf3972798e93fbc48c1/merged/etc/nginx/nginx.conf
./var/lib/docker/overlay2/793d7fa22cb7d1e29fa36ec6701d318f5b2320dfa592edf3972798e93fbc48c1/diff/etc/nginx/nginx.conf
./var/lib/docker/overlay2/6a34d5534e47724932d9474cc95a0cb7988f49cdacd4942b252864a31a8144d7/diff/etc/nginx/nginx.conf
./var/lib/docker/overlay2/d2c94b8db2ad500f5ef1ad5da3b0393e15bbb2c291e8059edb2a840f3e9d3b40/diff/etc/nginx/nginx.conf
./var/lib/docker/overlay2/915ad22c8860779776094e0712ee255b67523d2fbdb38f149459e5a908ad7980/merged/etc/nginx/nginx.conf
./var/lib/docker/overlay2/915ad22c8860779776094e0712ee255b67523d2fbdb38f149459e5a908ad7980/diff/etc/nginx/nginx.conf
./etc/nginx/nginx.conf
```

This is a list of the Docker overlays being used.

After we find where oure nginx.conf files are, we'll want to make sense of what all those overlays are associate with. To find out what overlay is associate with which container, we'll run the following:

`docker inspect $(docker ps -qa) | jq -r 'map([.Name, .GraphDriver.Data.MergedDir]) | .[] | "\(.[0])\t\(.[1])"'`

It should return something like this:

```
/portainer
/var/lib/docker/overlay2/deb596691739d7e75fcd1e7751082d33a0e82fb273685519d4771088f7db38a4/merged
/authelia
/var/lib/docker/overlay2/5ddef2e06a851bb519a743b53d72bdfea601e028730bc6fd05a2f990b8e76591/merged
/nginx_guacamole_compose
/var/lib/docker/overlay2/915ad22c8860779776094e0712ee255b67523d2fbdb38f149459e5a908ad7980/merged
/guacamole_compose
/var/lib/docker/overlay2/00eaae208982f0413aab19523f77299b07401159012839e2ef72e1b1e10852b1/merged
/postgres_guacamole_compose
/var/lib/docker/overlay2/df976d515810127c3e2bf28f80843cdbf2299e746c87c83cd7e59de62be5ea55/merged
/guacd_compose
/var/lib/docker/overlay2/702379337be210e8e48b101ff79e39eaa038b00c30989b092c66f06eb50ed29e/merged
/npm_app_1
/var/lib/docker/overlay2/793d7fa22cb7d1e29fa36ec6701d318f5b2320dfa592edf3972798e93fbc48c1/merged
```

In this case, we're looking for the "npm_app_1" overlay since we know that it is for NGINX Proxy Manager.

That tells us the nginx.conf file we're looking for is here:

`/var/lib/docker/overlay2/793d7fa22cb7d1e29fa36ec6701d318f5b2320dfa592edf3972798e93fbc48c1/merged/etc/nginx/nginx.conf`

So we're going to edit that file:

`sudo nano /var/lib/docker/overlay2/793d7fa22cb7d1e29fa36ec6701d318f5b2320dfa592edf3972798e93fbc48c1/merged/etc/nginx/nginx.conf`

Find the section that starts with `http {` . Scroll down in that section and find `# Real IP Determination`.

Paste the following right below that:

```
#CF IPs
        set_real_ip_from 103.21.244.0/22;
        set_real_ip_from 103.22.200.0/22;
        set_real_ip_from 103.31.4.0/22;
        set_real_ip_from 104.16.0.0/12;
        set_real_ip_from 108.162.192.0/18;
        set_real_ip_from 131.0.72.0/22;
        set_real_ip_from 141.101.64.0/18;
        set_real_ip_from 162.158.0.0/15;
        set_real_ip_from 172.64.0.0/13;
        set_real_ip_from 173.245.48.0/20;
        set_real_ip_from 188.114.96.0/20;
        set_real_ip_from 190.93.240.0/20;
        set_real_ip_from 197.234.240.0/22;
        set_real_ip_from 198.41.128.0/17;
        set_real_ip_from 2400:cb00::/32;
        set_real_ip_from 2606:4700::/32;
        set_real_ip_from 2803:f800::/32;
        set_real_ip_from 2405:b500::/32;
        set_real_ip_from 2405:8100::/32;
        set_real_ip_from 2c0f:f248::/32;
        set_real_ip_from 2a06:98c0::/29;

        real_ip_header X-Forwarded-For;
```

> This list of IP addresses may change, so please refer to this list of current CloudFlares IP addresses.

A little way below that you should see `# Local subnets`. In that section, look for `real_ip_header X-Real-IP`

Comment out `real_ip_header X-Real-IP` by placing a # in front of it so that it looks like `#real_ip_header X-Real-IP`

Save and exit.

> You may also want to change the permissions of the nginx.conf file from `644` to `604` to remove the write permission from the file to keep it from being overwritten when/if the

container is updated.

You can change the permissions of the nginx.conf file by running this command while still in the overlay directory:

```
chmod 604 nginx.conf
```

Restart the Fail2Ban container.

The container should come up and you should see something like this if everything went correctly:

```
2022-01-16 01:08:54,950 fail2ban.server         [1]: INFO
--------------------------------------------------
2022-01-16 01:08:54,950 fail2ban.server         [1]: INFO    Starting Fail2ban v0.11.2
2022-01-16 01:08:54,951 fail2ban.observer       [1]: INFO    Observer start...
2022-01-16 01:08:54,961 fail2ban.database       [1]: INFO    Connected to fail2ban persistent
database '/data/db/fail2ban.sqlite3'
2022-01-16 01:08:54,962 fail2ban.jail           [1]: INFO    Creating new jail 'npm-docker'
2022-01-16 01:08:54,974 fail2ban.jail           [1]: INFO    Jail 'npm-docker' uses pyinotify
{}
2022-01-16 01:08:54,976 fail2ban.jail           [1]: INFO    Initiated 'pyinotify' backend
2022-01-16 01:08:54,979 fail2ban.filter         [1]: INFO      maxRetry: 2
2022-01-16 01:08:54,979 fail2ban.filter         [1]: INFO      findtime: 600
2022-01-16 01:08:54,980 fail2ban.actions        [1]: INFO      banTime: 3600
2022-01-16 01:08:54,980 fail2ban.filter         [1]: INFO      encoding: UTF-8
2022-01-16 01:08:54,980 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/default-
host_access.log' (pos = 740, hash = b98d578fa16b9fab8732f5a0de03f6a71c69039b)
2022-01-16 01:08:54,981 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-17_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,981 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-26_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,982 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-11_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,982 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-10_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,982 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-13_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,983 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-33_access.log' (pos = 269556, hash = 0a6442840af52e2f0c0c0e5d0b4f691d2211ca2f)
2022-01-16 01:08:54,983 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-
host-8_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
```

```
2022-01-16 01:08:54,983 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-28_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,984 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-3_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,984 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-21_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,984 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-23_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,985 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-14_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,985 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-6_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,986 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-1_access.log' (pos = 61533, hash = 8928523abdadfe1ba9699888f64273cb6178d890)
2022-01-16 01:08:54,989 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-25_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,990 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-24_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,990 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-31_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,993 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-30_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,993 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-7_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,994 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-29_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,994 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-18_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,994 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-9_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,994 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-32_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,995 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-19_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,995 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-5_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,995 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-4_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,996 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-22_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
```

```
2022-01-16 01:08:54,996 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-2_access.log' (pos = 51260, hash = 30133d5d464bce57bf00b304322991dcd738b118)
2022-01-16 01:08:54,996 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-27_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,996 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-16_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,997 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-15_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:54,997 fail2ban.filter         [1]: INFO    Added logfile: '/log/npm/proxy-host-12_access.log' (pos = 0, hash = da39a3ee5e6b4b0d3255bfef95601890afd80709)
2022-01-16 01:08:55,002 fail2ban.jail           [1]: INFO    Jail 'npm-docker' started

Server ready
```