Blacklist IP's from a file

fail2ban is designed to dynamically watch logs and ban/unban IP's with bad reputations. However, with a little sneakiness, it can be abused to also load a list of permanently banned IP addresses. This is a drastic action, and can end up blocking legitimate users who gain IP's previously used for cracking attempts, so I tend to clean them up every once in a while. But, for me, it is a list of IP's that have done some extended hacking on my servers in the past, and this way, fail2ban doesn't have to monitor their attempts.

Basically, we create a custom action which creates its own chain (under fail2ban's control), loads the IP's, then that is all. The jail is there just to call the action the first time, and the filter is there because fail2ban requires one (and complains if you set it to null).

We'll create three files, and add a block to jail.local

1. list of IP's or subnets, one entry per line
2. action file to be stored in actions.d/
3. dummy filter file to be stored in filters.d/
4. modify jail.local

Blacklist File

This is a basic text file with one IP or subnet per line. I store mine in /etc/fail2ban on my Devuan Linux machine. An example is:

ip.blacklist

```
172.104.94.112
190.40.235.20
190.4.51.122
210.186.135.78
39.45.148.16
193.93.16.14
93.174.93.0/24
```

Action file

Now, we need to create an action, a file to be placed in action.d. I named it blacklistip.conf, and the name is important when you define the jail it goes in, ie action = blacklistip means *look in action.d for a file names blacklistip.conf and load it as the action*.

action.d/blacklistip.conf

```
# action file to allow loading of IP's from a text file to be blocked
```

```
# along with fail2ban.
# The file contains one IP or subnet per line, and may be placed
# anywhere on the system.
#
# This is a perversion of fail2ban's basic purpose, which is to
dynamically
# add/remove IP's from IPTables, but it allows us to permenantly ban
# some really, really bad people
#
# example of file
# 172.104.94.112
# 190.40.235.20
# 190.4.51.122
# 210.186.135.78
# 39.45.148.0/24
#
# NOTE: I did not set it up to ignore comments, so you can't put
comments
# into the file.
#
# Works on fail2ban v9


[INCLUDES]

before = iptables-common.conf


[Definition]

# what to do when fail2ban starts
# taken directly from the multiport ban script, with the last line
# inserted to load the IP file
# creates a chain, then loads all the IP's into it
actionstart = <iptables> -N f2b-<name>
              <iptables> -A f2b-<name> -j <returntype>
              <iptables> -I <chain> -p <protocol> -j f2b-<name>
              cat <filename> | while read IP; do <iptables> -I f2b-
<name> 1 -s $IP -j DROP; done


# these actions are taken when fail2ban is shut down
# basically, destroys the chain
actionstop = <iptables> -D <chain> -p <protocol> -j f2b-<name>
             <iptables> -F f2b-<name>
             <iptables> -X f2b-<name>


actioncheck =
actionban =
actionunban =
```

```
[Init]
```

Filter

Now, we need a filter, because we are abusing fail2ban. fail2ban assumes you're going to be parsing a log file to find bad guys attacking you, but we already know who we want to block. So, we create a dummy and store it in filter.d/blacklistip.conf. The name of the file is not arbitrary. It is the default based on the name of our jail definition (later). We could call it anything, but would need to add

```
filter = anything
```

to our jail if we do. This is simpler.

filter.d/blacklistip.conf

```
# dummy filter file for blacklistip jail. Expect a warning that
failregex is
# not defined, or, if you uncomment failregex, expect a warning that
there
# is no <Host> entry in it
#
# Since this is a static read, we don't actually parse any logs

[INCLUDES]


[Definition]

#failregex = ''

[Init]
```

Modify jail.local

jail.local (in the root of the fail2ban configuration directory) is the place to make local modifications. So, we need to add the following block to it. This basically defines a jail named *blacklistip*, which is enabled. Since we don't have a *filter* = line, the filter is assumed to be filter.d/blacklistip.conf.

The action is specifically defined to be blacklistip (ie, action.d/blacklistip.conf), and we are passing the name of the chain to create (name=blacklistip) and the file name to be read from (filename='/etc/fail2ban/ip.blacklist').

```
[blacklistip]
```

```
enabled = true
action = blacklistip[name=blacklistip,filename='/etc/fail2ban/ip.blacklist']
```

Test it

Restart fail2ban, then run the following command as root.

```
iptables -n -L f2b-blacklistip
```

You should see a list of all the banned IP's, with an action of drop. And, when you stop fail2ban, it will clean them up also.

Deficiencies

Actually, this should be done on the router, since it will use some memory and processor on your server. Also, there is no way to dynamically add/remove IP's. You must modify the file, then restart fail2ban.

Using fail2ban-client, you can add IP's (or remove them) from this list, so a simple script should be able to A) append/remove the IP from ip.blacklist B) append/remove the IP from the f2b-blacklistip chain but, I haven't written one yet.

Links

- https://www.fail2ban.org/wiki/index.php/Commands
- https://www.mauromascia.com/en/blog/fail2ban-set-permanent-ban-per-ip/

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/software/fail2ban/blacklist?rev=1565680453

Last update: **2019/08/13 07:14**