*By Falko Timme*
Published: 2007-05-01 19:05

# Preventing Brute Force Attacks With Fail2ban On Debian Etch

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 04/24/2007

In this article I will show how to install and configure **fail2ban** on a Debian Etch system. Fail2ban is a tool that observes login attempts to various services, e.g. SSH, FTP, SMTP, Apache, etc., and if it finds failed login attempts again and again from the same IP address or host, fail2ban stops further login attempts from that IP address/host by blocking it with an iptables firewall rule.

This document comes without warranty of any kind! I want to say that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

## Preliminary Note

Fail2ban is similar to **DenyHosts** which I covered in this tutorial: **http://www.howtoforge.com/preventing_ssh_dictionary_attacks_with_denyhosts**, but unlike DenyHosts which focuses on SSH, fail2ban can be configured to monitor any service that writes login attempts to a log file, and instead of using `/etc/hosts.deny` to block IP addresses/hosts, fail2ban uses iptables.

In this example I will configure fail2ban to monitor login attempts to the SSH server, the Proftpd server, login attempts to .htaccess/.htpasswd protected web sites, to Courier POP3 and Courier IMAP, and to SASL (for sending emails). I will install the fail2ban package that is available for Debian Etch. It comes with a default configuration, but unfortunately that configuration doesn't quite work for most of the aforementioned services. Therefore I will create a customized fail2ban configuration that I have tested and that works for me.

## Installing fail2ban

Fail2ban can be installed as follows on Debian Etch:

```
apt-get install fail2ban
```

Afterwards, you will find all fail2ban configuration files in the */etc/fail2ban* directory.

## Configuring fail2ban

The default behaviour of fail2ban is configured in the file */etc/fail2ban/jail.conf*. Take a look at it, it's not hard to understand. There's a *[DEFAULT]* section that applies to all other sections unless the default options are overriden in the other sections.

I explain some of the configuration options here:

- *ignoreip*: This is a space-separated list of IP addresses that cannot be blocked by fail2ban. For example, if the computer from which you're connecting to the server has a static IP address, you might want to list it here.
- *bantime*: Time in seconds that a host is blocked if it was caught by fail2ban (600 seconds = 10 minutes).
- *maxretry*: Max. number of failed login attempts before a host is blocked by fail2ban.
- *filter*: Refers to the appropriate filter file in */etc/fail2ban/filter.d*.
- *logpath*: The log file that fail2ban checks for failed login attempts.

As suggested by a comment at the top of */etc/fail2ban/jail.conf*, we don't modify */etc/fail2ban/jail.conf* itself to adjust it to our needs, but override it by creating a new configuration file, */etc/fail2ban/jail.local*.

This is what my */etc/fail2ban/jail.local* file looks like:

```
vi /etc/fail2ban/jail.local
```

```
[DEFAULT]

# "ignoreip" can be an IP address, a CIDR mask or a DNS host
ignoreip = 127.0.0.1 192.168.0.99
bantime  = 600
```

```
maxretry = 3

# "backend" specifies the backend used to get files modification. Available
# options are "gamin", "polling" and "auto".
# yoh: For some reason Debian shipped python-gamin didn't work as expected
#      This issue left ToDo, so polling is default backend for now
backend = polling

#
# Destination email address used solely for the interpolations in
# jail.{conf,local} configuration files.
destemail = root@localhost

# Default action to take: ban only
action = iptables[name=%(__name__)s, port=%(port)s]



[ssh]

enabled = true
port    = ssh
filter  = sshd
logpath  = /var/log/auth.log
maxretry = 5



[apache]

enabled = true
port    = http
filter  = apache-auth
logpath = /var/log/apache*/*error.log
maxretry = 5
```

```
[apache-noscript]

enabled = false
port    = http
filter  = apache-noscript
logpath = /var/log/apache*/*error.log
maxretry = 5


[vsftpd]

enabled  = false
port     = ftp
filter   = vsftpd
logpath  = /var/log/auth.log
maxretry = 5


[proftpd]

enabled  = true
port     = ftp
filter   = proftpd
logpath  = /var/log/auth.log
failregex = proftpd: \(pam_unix\) authentication failure; .* rhost=<HOST>
maxretry = 5


[wuftpd]

enabled  = false
```

```
port    = ftp
filter   = wuftpd
logpath  = /var/log/auth.log
maxretry = 5



[postfix]

enabled  = false
port     = smtp
filter   = postfix
logpath  = /var/log/mail.log
maxretry = 5



[courierpop3]

enabled  = true
port     = pop3
filter   = courierlogin
failregex = courierpop3login: LOGIN FAILED.*ip=\[.*:<HOST>\]
logpath  = /var/log/mail.log
maxretry = 5



[courierimap]

enabled  = true
port     = imap2
filter   = courierlogin
failregex = imapd: LOGIN FAILED.*ip=\[.*:<HOST>\]
logpath  = /var/log/mail.log
maxretry = 5
```

```
[sasl]

enabled  = true
port     = smtp
filter   = sasl
failregex = warning: [-._\w]+\[<HOST>\]: SASL (?:LOGIN|PLAIN|(?:CRAM|DIGEST)-MD5) authentication failed
logpath  = /var/log/mail.log
maxretry = 5
```

My client computer has the static IP address *192.168.0.99*, and because I don't want to be locked out, I've added it to the *ignoreip* list. I've set the max. number of failed login attempts to 5 for all services, and I've created two new sections, *[courierpop3]* and *[courierimap]*, so that fail2ban can block login attempts to my Courier-POP3 and Courier-IMAP server.

I want to control login attempts to ssh, apache, proftpd, courierpop3, courierimap, and sasl, so I've set *enabled* to *true* for these services and to *false* for all other services.

If you compare the file with */etc/fail2ban/jail.conf*, you'll also notice that I've changed some log files because the log files in */etc/fail2ban/jail.conf* are not correct for Debian Etch. In addition to that, I've added a *failregex* line to some services because the regular expressions in the appropriate filter files in the */etc/fail2ban/filter.d* directory do not work for Debian Etch. The *failregex* line overrides the filter rule in the appropriate file in */etc/fail2ban/filter.d*.

Whenever we modify the fail2ban configuration, we must restart fail2ban, so this is what we do now:

```
/etc/init.d/fail2ban restart
```

That's it already. Fail2ban logs to */var/log/fail2ban.log*, so you can check that file to find out if/what hosts got blocked. If a host got blocked by fail2ban, it looks like this:

```
2007-04-24 17:49:09,466 fail2ban.actions: WARNING [apache] Ban 1.2.3.4
```

```
  2007-04-24 18:08:33,213 fail2ban.actions: WARNING [sasl] Ban 1.2.3.4
  2007-04-24 18:26:37,769 fail2ban.actions: WARNING [courierlogin] Ban 1.2.3.4
2007-04-24 18:39:06,765 fail2ban.actions: WARNING [courierimap] Ban 1.2.3.4
```

You can also check your firewall to see if any hosts are currently blocked. Simply run

```
iptables -L
```

## Links

- Fail2ban: **http://www.fail2ban.org**
- Debian: **http://www.debian.org**