FreeBSD Administration - Support #564

Install Fail2ban on FreeBSD

02/15/2015 06:09 PM - Daniel Curtis

Status:	Closed	**Start date:**	02/15/2015
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Intrusion Detection/Prevention	**Estimated time:**	0.50 hour
Target version:	FreeBSD 9	**Spent time:**	3.50 hours

Description

Fail2ban scans log files and bans IPs that show the malicious signs like too many password failures, seeking for exploits, and such. It can be useful to ban bots who try to bruteforce your ssh and flood your logs (another solution is to restrict allowed IP or change sshd port). This is a simple guide on setting up fail2ban on FreeBSD, in combination with ipfw.

~~Install IPFW~~

ipfw is now built into FreeBSD.

- ~~Install ipfw:~~

```
pkg install ipfw
```

Configure IPFW

- Start and enable ipfw at boot:

```
echo 'firewall_enable="YES"' >> /etc/rc.conf
echo 'firewall_script="/usr/local/etc/ipfw.rules"' >> /etc/rc.conf
service ipfw start
```

Install Fail2ban

- Install py27-fail2ban

```
pkg install py27-fail2ban
```

- Then create the ssh-ipfw.local file

```
vi /usr/local/etc/fail2ban/jail.d/ssh-ipfw.local
```

  - And add the following

```
[ssh-ipfw]
enabled  = true
filter   = sshd
action   = ipfw
#          sendmail-whois[name=SSH, dest=root@localhost, sender=noreply@localhost]
logpath  = /var/log/auth.log
findtime  = 600
maxretry = 3
bantime  = 3600
```

- Edit the ipfw action file:

```
vi /usr/local/etc/fail2ban/action.d/ipfw.conf
```

  - And modify the localhost parameter to the IP address of the server:

```
localhost = 192.168.1.100
```

- Start and enable fail2ban at boot:

```
echo 'fail2ban_enable="YES"' >> /etc/rc.conf
service fail2ban start
```

Now you can look in /var/log/fail2ban.log to see detected IP and applied ban.

- To list current banned IP:

```
ipfw list
```

Securing Web Services

ownCloud

This example uses the owncloud package available from the ports tree.

- Create the owncloud filter definition for fail2ban

```
/usr/local/etc/fail2ban/filter.d/owncloud.conf
```

  - And add the following

```
[Definition]
failregex={"app":"core","message":"Login failed: user '.*' , wrong password, IP:<HOST>","l
evel":2,"time":".*"}
          {"app":"core","message":"Login failed: '.*' \(Remote IP: '<HOST>', X-Forwarded-F
or: '.*'\)","level":2,"time":".*"}
          {"reqId":".*","remoteAddr":"<HOST>","app":"core","message":"Login failed: .*","l
evel":2,"time":".*"}
```

The first line is for owncloud <= 7.0.1. The second for owncloud 7.0.2-7.05 and the bottom one for owncloud 8.

- Create the owncloud service definition:

```
vi /usr/local/etc/fail2ban/jail.d/owncloud-auth.conf
```

  - And add the following:

```
[owncloud]
enabled = true
filter  = owncloud
port    = http,https
logpath = /usr/local/www/owncloud/data/owncloud.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Redmine

This example uses the redmine package available from the ports tree.

- Create the redmine filter definition for fail2ban

  `/usr/local/etc/fail2ban/filter.d/redmine.conf`

  - And add the following

    ```
    [Definition]
    failregex = Failed [-/\w]+ for .* from <HOST>
    ```

- Create the redmine service definition:

  `vi /usr/local/etc/fail2ban/jail.d/redmine-auth.conf`

  - And add the following:

    ```
    [redmine]
    enabled = true
    filter  = redmine
    port    = http,https
    logpath = /usr/local/www/redmine/log/production.log
    ```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Piwik

This example uses the piwik package available from the ports tree.

- Create the piwik filter definition for fail2ban

  `/usr/local/etc/fail2ban/filter.d/piwik.conf`

  - And add the following

    ```
    [Definition]
    failregex = ^<HOST> -.*"POST /
    ```

- Create the piwik service definition:

  `vi /usr/local/etc/fail2ban/jail.d/piwik-auth.conf`

  - And add the following:

    ```
    [piwik]
    enabled = true
    filter  = piwik
    port    = http,https
    ```

```
logpath = /var/log/piwik.example.com-access.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

WordPress

This example uses the piwik package available from the ports tree.

- Create the wordpress filter definition for fail2ban

```
/usr/local/etc/fail2ban/filter.d/wordpress.conf
```

  - And add the following

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

- Create the wordpress service definition:

```
vi /usr/local/etc/fail2ban/jail.d/wordpress-auth.conf
```

  - And add the following:

```
[wordpress]
enabled = true
filter  = wordpress
port    = http,https
logpath = /var/log/wordpress.example.com-access.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

GitLab

- Create the gitlab filter definition for fail2ban

```
/usr/local/etc/fail2ban/filter.d/gitlab.conf
```

  - And add the following

```
[Definition]
failregex = ^<HOST> -.*"POST /users/sign_in HTTP.*$
```

- Create the gitlab service definition:

```
vi /usr/local/etc/fail2ban/jail.d/gitlab-auth.conf
```

  - And add the following:

```
[gitlab]
enabled = true
filter  = gitlab
port    = http,https
```

```
logpath = /var/log/gitlab.example.com-access.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Dovecot

- The fail2ban package already has a dovecot filter:

```
[Definition]

_daemon = (auth|dovecot(-auth)?|auth-worker)

failregex = ^%(__prefix_line)s(pam_unix(\(dovecot:auth\))?:)?\s+authentication failure; logname=\S* uid=\S* euid=\S* tty=dovecot ruser=\S* rhost=<HOST>(\s+user=\S*)?\s*$
            ^%(__prefix_line)s(pop3|imap)-login: (Info: )?(Aborted login|Disconnected)(: Inactivity)? \(((auth failed, \d+ attempts)( in \d+ secs)?|tried to use (disabled|disallowed) \S+ auth)\):( user=<\S*>,)?( method=\S+,)? rip=<HOST>(, lip=(\d{1,3}\.){3}\d{1,3})?(, TLS( handshaking(: SSL_accept\(\) failed: error:[\dA-F]+:SSL routines:[TLS\d]+_GET_CLIENT_HELLO:unknown protocol)?)?(: Disconnected)?)?(, session=<\S+>)?\s*$
            ^%(__prefix_line)s(Info|dovecot: auth\(default\)): pam\(\S+,<HOST>\): pam_authenticate\(\) failed: (User not known to the underlying authentication module: \d+ Time\(s\)|Authentication failure \(password mismatch\?\))\s*$
```

- Create the dovecot service definition:

```
vi /usr/local/etc/fail2ban/jail.d/dovecot-auth.conf
```

  - And add the following:

```
[dovecot]
enabled = true
filter  = dovecot
port    = pop3,pop3s,imap,imaps
logpath = /var/log/dovecot.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Postfix

- The fail2ban package already has a postfix filter:

```
[Definition]

_daemon = postfix/(submission/)?smtp(d|s)

failregex = ^%(__prefix_line)sNOQUEUE: reject: RCPT from \S+\[<HOST>\]: 554 5\.7\.1 .*$
            ^%(__prefix_line)sNOQUEUE: reject: RCPT from \S+\[<HOST>\]: 450 4\.7\.1 : Helo command rejected: Host not found; from=<> to=<> proto=ESMTP helo= *$
            ^%(__prefix_line)sNOQUEUE: reject: VRFY from \S+\[<HOST>\]: 550 5\.1\.1 .*$
            ^%(__prefix_line)simproper command pipelining after \S+ from [^[]*\[<HOST>\]:?$
```

- Create the postfix service definition:

```
vi /usr/local/etc/fail2ban/jail.d/postfix-auth.conf
```

- And add the following:

```
[postfix]
enabled = true
filter  = postfix
port    = smtp,ssmtp
logpath = /var/log/maillog
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Prosody

NOTE: This requires the **mod_log_auth** community module, installing 3rd party modules is covered here in Issue #535

- Create the prosody filter definition for fail2ban

```
/usr/local/etc/fail2ban/filter.d/prosody.conf
```

- And add the following

```
[Definition]
failregex = Failed authentication attempt \(not-authorized\) from IP: <HOST>
```

- Create the prosody service definition:

```
vi /usr/local/etc/fail2ban/jail.d/prosody-auth.conf
```

- And add the following:

```
[prosody]
enabled = true
filter  = prosody
port    = 5222
logpath = /var/log/prosody*.log
```

Now restart fail2ban and try to log in 4 times with a wrong password. The 4th attempt should give you a timeout for 15min.

Resources

- http://blog.alteroot.org/articles/2014-06-14/fail2ban-on-freebsd.html
- https://nileshgr.com/2013/04/18/securing-freebsd-server-with-fail2ban-and-ipfw
- https://anonymous-proxy-servers.net/wiki/index.php/FreeBSD_SSH_port_security_3#Setting_up_fail2ban
- http://www.rojtberg.net/711/secure-owncloud-server/
- https://github.com/gitlabhq/gitlabhq/issues/1001
- http://www.fail2ban.org/wiki/index.php/Postfix
- http://wiki.dovecot.org/HowTo/Fail2Ban
- http://wiki.prosody-modules.googlecode.com/hg/mod_log_auth.wiki
- https://23x.net/908/securing-wordpress-using-fail2ban.html

History

#1 - 03/11/2015 02:20 PM - Daniel Curtis

- Subject changed from Installing Fail2ban on FreeBSD to Install Fail2ban on FreeBSD

- Description updated

- Status changed from New to In Progress

#2 - 03/11/2015 02:28 PM - Daniel Curtis

- % Done changed from 0 to 90

#3 - 03/11/2015 04:26 PM - Daniel Curtis

- Description updated

#4 - 03/11/2015 04:39 PM - Daniel Curtis

- Description updated

#5 - 03/13/2015 09:30 PM - Daniel Curtis

- Description updated

#6 - 03/15/2015 04:50 PM - Daniel Curtis

- Description updated

#7 - 03/15/2015 05:01 PM - Daniel Curtis

- Description updated

#8 - 03/15/2015 07:29 PM - Daniel Curtis

- Description updated

- % Done changed from 90 to 100

#9 - 03/15/2015 08:08 PM - Daniel Curtis

- Description updated

#10 - 03/15/2015 08:10 PM - Daniel Curtis

- Description updated

#11 - 03/17/2015 02:09 PM - Daniel Curtis

- Description updated

#12 - 04/14/2015 12:59 PM - Daniel Curtis

- Status changed from In Progress to Resolved

#13 - 06/11/2015 10:56 AM - Daniel Curtis

- Description updated

#14 - 06/11/2015 12:33 PM - Daniel Curtis

- Description updated

#15 - 11/27/2015 04:48 PM - Daniel Curtis

- Status changed from Resolved to Closed