Fail2Ban (with iptables) And Asterisk

Fail2Ban

Fail2Ban is a limited intrusion detection/prevention system. It works by scanning log files and then taking action based on the entries in those logs.
We are implementing Fail2Ban with a configuration to be able to prevent SIP brute force attacks against our Asterisk PBXs.

You can get Fail2Ban, as well as more documentation, at www.fail2ban.org. At the time this is being written, the current release is **0.8.4.**

Page Contents

Fail2Ban With Asterisk

The following describes how to setup Fail2Ban to protect an Asterisk PBX from SIP brute force attempts and scans utilizing the iptables firewall.

SECURITY NOTE: fail2ban is rather limited in its ability to detect attacks against asterisk.
More info http://forums.asterisk.org/viewtopic.php?p=159984

Easy Install Script for Fail2ban version 0.8.4 / Red Hat

This script was written by Cédric Brohée in order to simplify and accelerate the integration of the solution in a basic Asterisk configuration on Red Hat. Do not hesitate to read the bash script and make changes to match your own configuration.

Before running it, you will have to do chmod 755.

Download script :

Dan at VoicePlex fixed a small error in the script to download and unpack fail2ban ...

Installing

Log into the system and **su - root**, or **sudo -i** to get a root shell on Ubuntu.

CentOS/Red Hat (this method may install an older version of fail2ban):

Install rpmforge or optionally fetch the fail2ban rpm directly from rpmforge.
Install fail2ban using yum:

```
yum install fail2ban
```

Debian/Ubuntu:

```
apt-get install fail2ban
```

Source installation:

Source installation:
Change directories to **/usr/src**:

```
cd /usr/src
```

Download and extract Fail2Ban (check for newer releases):

```
wget http://sourceforge.net/projects/fail2ban/files/fail2ban-stable/fail2ban-0.8.4/fail2ban-0.8.4.tar.bz2/download
tar jxf fail2ban-0.8.4.tar.bz2
```

Enter the Fail2Ban directory you just extracted:

```
cd fail2ban-0.8.4
```

Make sure python and iptables are installed:

CentOS/Red Hat:

```
yum install python iptables
```

Debian/Ubuntu:

```
apt-get install python iptables
```

Install Fail2Ban:

```
python setup.py install
```

Install the Fail2Ban init script (for source installations):

Centos/Red Hat (if you installed via yum/rpm, the init script has already been installed):

```
cp /usr/src/fail2ban-0.8.4/files/redhat-initd /etc/init.d/fail2ban
chmod 755 /etc/init.d/fail2ban
```

For other distributions' init scripts, please refer to documentation specific to them.

Configure Fail2Ban

We need to create a configuration for Fail2Ban so that it can understand attacks against Asterisk.

Create a new filter configuration for Asterisk:

```
touch /etc/fail2ban/filter.d/asterisk.conf
```

The contents of **/etc/fail2ban/filter.d/asterisk.conf** should be the following:

```
# Fail2Ban configuration file
#
#
# $Revision: 250 $
#

[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
#before = common.conf


[Definition]

#_daemon = asterisk

# Option:  failregex
# Notes.:  regex to match the password failures messages in the logfile. The
#          host must be matched by a group named "host". The tag "<HOST>" can
#          be used for standard IP/hostname matching and is only an alias for
#          (?:::f{4,6}:)?(?P<host>\S+)
# Values:  TEXT
#

failregex = NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Wrong password
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - No matching peer found
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Username/auth name mismatch
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Device does not match ACL
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - Peer is not supposed to register
            NOTICE.* .*: Registration from '.*' failed for '<HOST>' - ACL error (permit/deny)
            NOTICE.* <HOST> failed to authenticate as '.*'$
            NOTICE.* .*: No registration for peer '.*' \(from <HOST>\)
            NOTICE.* .*: Host <HOST> failed MD5 authentication for '.*' (.*)
            NOTICE.* .*: Failed to authenticate user .*@<HOST>.*

# Option:  ignoreregex
# Notes.:  regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

If you're having issues with your system not banning properly when the "Registration from" section in your log file contains a quotation mark (") as in this example:

```
[2011-04-07 17:53:11] NOTICE[7557] chan_sip.c: Registration from '"69106698"<sip:69106698@
123.123.123.123>' failed for '123.123.123.123' - No matching peer found
```

Add the following line, with the others above, in asterisk.conf:

NOTICE.* .* : Registration from '\".*\".*' failed for '<HOST >' - No matching peer found

Recently noticed attacks :

```
[2011-06-21 17:53:11] NOTICE[7557] chan_sip.c: Registration from '"XXXXXXXXXX"<sip:XXXXXXXXXX@
123.123.123.123>' failed for '123.123.123.123' - Wrong Password
```

Adding the following line will block these attempts:

NOTICE.* .* : Registration from '\".*\".*' failed for '<HOST >' - Wrong password

Next edit **/etc/fail2ban/jail.conf** to include the following section so that it uses the new filter. This does a 3-day ban on the IP that performed the attack. It is recommend to set the **bantime** in the [DEFAULT] section so if affects all attacks. It is also recommend to turn on an iptables ban for ssh, httpd/apache, and ftp if they are running on the system. Be sure to edit the **sendmail-whois** action to send notifications to an appropriate address:

```
[asterisk-iptables]

enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, protocol=all]
           sendmail-whois[name=ASTERISK, dest=root, sender=fail2ban@example.org]
logpath  = /var/log/asterisk/messages
maxretry = 5
bantime = 259200
```

note: logpath = /var/log/asterisk/messages is for vanilla asterisk, use logpath = /var/log/asterisk/full for freepbx. You can check the name of the log file in logger.conf.

note: if fail2ban still failed to identify login attempts, try the syslog logging way.

note: if fail2ban still failed to identify login attempts, try the syslog logging way.

Don't Ban Yourself

We don't want to ban ourselves by accident. Edit **/etc/fail2ban/jail.conf** and edit the **ignoreip** option under the [DEFAULT] section to include your IP addresses or network, as well as any other hosts or networks you do not wish to ban. Note that the addresses must be separated by a SPACE character!

Asterisk Logging

We must change how Asterisk does its time stamp for logging. The default format does not work with Fail2Ban because the pattern Fail2Ban uses that would match this format has a beginning of line character (^), and Asterisk puts its date/time inside of []. The other formats that Fail2Ban supports, however, do not have this character and can be used with Asterisk.

To change this format, open **/etc/asterisk/logger.conf** and add the following line under [general] section (You may have to create this before the [logfiles] section). This causes the date and time to be formatted as Year-Month-Day Hour:Minute:Second, [2008-10-01 13:40:04] is an example.

```
[general]
dateformat=%F %T
```

Then reload the logger module for Asterisk. At the command line, run the following command:

```
asterisk -rx "logger reload"
```

If for some reason you do not want to change the date/time format for your normal asterisk logs (maybe you already have scripts that use it or something and do not want to edit them), you can do the following instead:

In **/etc/asterisk/logger.conf**, add the following line under the [logfiles] section for Asterisk to log NOTICE level events to the syslog (/var/log/messages) as well as its normal log file. These entries in syslog will have a Date/Time stamp that is usable by Fail2Ban.

```
syslog.local0 => notice
```

Be sure to reload the logger module for Asterisk — check above for the command to do so. If you chose this option, you will also have to change the **/etc/fail2ban/jail.conf** setting under the [asterisk-iptables] section for the **logpath** option to the following:

```
logpath  = /var/log/messages
```

Turning it On

Now it is time to put fail2ban to work. There are a couple steps we need to do first.

Turn IPTABLES on

By default, iptables allows all traffic. So if we turn it on, it will not block any traffic until Fail2Ban creates deny rules for attackers. You should create your own firewall rules and setup for iptables, but that is beyond the scope of this guide. Just know that Fail2Ban, by default, inserts rules at the top of the chain, so they will override any rules you have configured in iptables. This is good because you may allow all sip traffic in and then the Fail2Ban will block individual hosts, after they have done an attack, before they are allowed by this rule again.

To start iptables, run the following as root:

```
/etc/init.d/iptables start
```

Depending on your install, you may or may not have the iptables init script installed. Please refer to an iptables install/setup guide for your distribution for more information.

Turn on Fail2Ban

To start Fail2Ban, run the following as root:

```
/etc/init.d/fail2ban start
```

Check It

If both started properly, issue the following command to view your iptables rules:

```
iptables -L -v
```

You should see something like the following for the INPUT chain (you will see more if you have other Fail2Ban filters enabled):

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out      source                  destination
2104K  414M fail2ban-ASTERISK  all  –  any     any      anywhere                anywhere
```

If you do not see something similar to that, then you have some troubleshooting to do; check out **/var/log/fail2ban.log**.

If you do not see all your rules, or if you see a different subset of rules after stopping and restarting fail2ban, you may be experiencing the issue described on this page on the Fail2ban talk:Community Portal and may wish to use the suggested fix:

fail2ban.action.action ERROR on startup/restart

I had multiple fail2ban.action.action ERROR on startup/restart. It seems there was a "race" condition with iptables. I solved the problem completely on my system by editing /usr/bin/fail2ban-client and adding a **time.sleep(0.1)**

```
def __processCmd(self, cmd, showRet = True):
        beautifier = Beautifier()
        for c in cmd:
                time.sleep(0.1)
                beautifier.setInputCmd(c)
```

Turn it on for good

If all is well up to this point, let's make sure that fail2ban and iptables restart with the server by issuing the following commands.

Centos/Red Hat:

```
chkconfig iptables on
chkconfig fail2ban on
```

Debian/Ubuntu:

```
update-rc.d iptables defaults
update-rc.d fail2ban defaults
```

You should now be somewhat protected against SIP scans and brute force attacks!

Try a reboot

Once you have fail2ban working ok, make sure that it continues that way after rebooting the server. On some distributions (including Ubuntu daper) fail2ban won't start after the system reboots because the /var/run/fail2ban directory gets deleted and needs to be re-created. This can be frustrating as there is also nothing that shows up in the logs to indicate what the problem is. If this happens, please see the link below for instructions on modifying the startup script so that it checks for and creates the /var/run/fail2ban directory if needed:

http://informationideas.com/news/2010/04/21/fail2ban-does-not-start-after-reboot/

Additional Information

- For those who may want a bit of additional security, this thread on iptables rate limiting at the PBX in a Flash Forum discusses a possible way to limit the number of attempts a bot can make at registering before fail2ban kicks in (e.g., if the bot is so fast it can make many attempts before fail2ban detects that many > 3).
- You may also want to consider adding Asterisk security through geographic IP address restriction
- See http://www.opensolutions.ie/blog/2010/09/sip-brute-force-attacks/ for a quick howto on using Fail2ban with Ubuntu/Debian as well as some discussion on SIP brute force attacks.
- An alternative to fail2ban which is more simple (but of course less configurable) http://www.dumaisnet.ca/index.php?p=asteriskapp#astban

- Alternate using **Perl** and **iptables**: Team Forrest - Automatically Block Failed SIP Peer Registrations

Created by: bulak ,Last modification on Sat 09 of Jul, 2011 [21:06 UTC] by obeliks

Search: Search