Fail2Ban (thanks to bulak)

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 Trixbox PBX's.

You can get Fail2Ban at **www.fail2ban.org**, as well as more documentation. At the time this is being written the current release is **0.8.3.**

Fail2Ban With Trixbox

The following describes how to setup Fail2Ban to protect Trixbox from SIP and IAX brute force attempts and scans, utilizing the iptables firewall.

Installing

Log into the system and sudo su - to root.

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

```
cd /usr/src
```

Download and Extract Fail2Ban (check for newer releases):

```
wget http://superb-east.dl.sourceforge.net/sourceforge/fail2ban/fail2ban-0.8.3.tar.bz2
tar -jxf fail2ban-0.8.3.tar.bz2
```

Enter the Fail2Ban directory you just extracted:

```
cd fail2ban-0.8.3
```

Install Fail2Ban:

```
python setup.py install
```

Install the Fail2Ban init script:

```
cp /usr/src/fail2ban-0.8.3/files/redhat-initd /etc/init.d/fail2ban
chmod 755 /etc/init.d/fail2ban
```

Configure Fail2Ban

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

Go to the filter folder for Fail2Ban:

```
cd /etc/fail2ban/filter.d
```

Create a new filter configuration for Asterisk:

```
touch 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.* <HOST> failed to authenticate as '.*'$
            NOTICE.* .*: No registration for peer '.*' (from )
            NOTICE.* .*: Host  failed MD5 authentication for '.*' (.*)

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

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. I would recommend setting the **bantime** in the [DEFAULT] section so if affects all attacks. I would also recommend turning 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/full
maxretry = 5
bantime = 259200
```

Don't Ban Yourself

We don't want to ban ourselves on 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.

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 []. However the other formats that Fail2Ban supports 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 enter 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 of 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 type the following as root:

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

Turn on Fail2Ban

To start Fail2Ban type 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 seem something similar to that then you have some troubleshooting to, check out **/var/log/fail2ban.log**.

Turn it on for good

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

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

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