IntegratePostfixViaSpampd

Integrating SpamAssassin into Postfix using spampd

Let's try the hard way to secure our mailsystems. Not as convenient as using spamd and spamc is the approach mentioned in the postfix FilterReadme. Some guys at http://www.WorldDesign.com/index.cfm/rd/mta/spampd.htm have published a spamd replacement that works as a SMTP-proxy. The advantage over spamd/spamc is efficiency (no fork/exec()ed processes or temporary files required for failsafe operation) and the ability to use before-queue content filtering.

It can easily be integrated as a "content_filter" in postfix. The knack is, that mail classified as spam is forwarded to users, where the filter of their local eMail client should detect the spam-status. Goal should be to forward spam to a special user named "spamking". This could be done by using an alias-map for all users that like their spam removed. The solution described here is for a Mail-server with a limited number of users with varying knowledge. All users are "local", meaning they get their mail via POP/IMAP from the mailserver.

First of all make sure your postfix-server is healthy. The things we'll do are somewhat confusing in the simple world of Postfix.

Begin by editing /etc/postfix/master.cf:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (50)
# ==========================================================================
...
smtp      inet  n       -       n       -       -       smtpd
...
scan               unix    -        -        n        -        10        smtp
localhost:10026    inet    n        -        n        -        10        smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o myhostname=filter.mynetwork.local
        -o smtpd_helo_restrictions=
        -o smtpd_client_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
spamtnsp          unix        -        n        n        -        -        local
        -o alias_maps=hash:/etc/postfix/spamalias
```

Next edit /etc/postfix/main.cf:

```
content_filter =scan:[127.0.0.1]:10025
header_checks = regexp:/etc/postfix/spamheadercheck
```

The service name "scan" is free and refers to the entry content_filter in main.cf. Scan defines that SMTP should be used. The scanpd-daemon is listening to 10025 and will deliver to localhost:10026. The transport "localhost:10026" defines an smtpd-server, with options slightly different to the main SMTP server. Especially the "content_filter=" is needed.

The file "spamheadercheck" mentioned in main.cf consist of one line:

```
/^X-Spam-Status: Yes/ FILTER spamtnsp:local
```

The regular expression searches every mail (including those coming from the local net!) for the header "X-Spam-Status: Yes", which is added by spamassassin in case of spam. The spam will be passed to the local transport spamtnsp defined in master.cf. The spamtnsp has the option alias_maps pointing to "spamalias". In spamalias every user that doesn't want his spam delivered to his mailbox has an alias:

```
user1:  spamking
user2:  spamking
...
```

The "spamking" user must have a home directory. Spamking can be used as a user for the site-wide bayes-filtering and as daemon user for spampd.

After thinking about we have done, we can start the spampd and postfix by calling

```
spampd --port=10025 --relayhost=127.0.0.1:10026 --user=spamking --tagall
rcpostfix reload
```

A test should be made with "telnet 127.0.0.1 10025". The Postfix-SMTP should be accessible via the Proxy.

Next step is ripping the spamassassin rc-script in order to start spampd. Perhaps someone is able to change spampd in the way that it can be started by postfix itself via master.cf.

Comments

I found that with this setup on my sever, SpamAssassin couldn't determine the envelope sender as needed for certain rules (e.g. DNS_FROM_*, NO_DNS_FOR_FROM, SPF_*). I fixed this by passing the -~~sef (~~-seh could work as well; but see documentation first) switch to spampd and then adding

```
 envelope_sender_header X-Envelope-From
```

to my SpamAssassin config. - JoshuaPettett

Spampd as a Before-Queue Content Filter

Following the guidelines at http://www.postfix.org/SMTPD_PROXY_README.html I was able to install spampd as a before-queue filter on an Ubuntu server. This allows rejection of spam before the connection with the offending SMTP client has been lost, thus eliminating the impossible choice of what to do about false positives in the case where you don't want users to be bothered with having to review their spam.

In a nutshell, it goes like this:

(1) Use Synaptic to install spampd, spamassassin, libnet-dns-perl and libmail-spf-query-perl.

(2) Make /root/.spamassassin writable for the spampd user.

(3) Edit /etc/default/spampd to set "STARTSPAMPD=1" and to make any other desired changes.

(4) Edit /etc/spamassassin/local.cf and make any desired changes. I set use_bayes to 0 to avoid the overhead and complexity of Bayesian filtering.

(5) Run sa-update to bring the filters up to date, and create a cron script /etc/cron.daily/spampd looking like this to perform these updates daily:

```
#!/bin/sh
sa-update && /etc/init.d/spampd restart
true
```

- Scott Lamb writes: "The cron script must [...] restart spampd for the changes to take effect. [...] on [RedHat]-based distributions, `sa-update && /sbin/service spampd restart` will do. (sa-update returns false if no updates were available.) This might cause mails in progress at the time to get a 4xx failure with 'queue file write error' because spampd does not support graceful restarts. That's harmless; the remote system will retry."

(6) Edit /etc/postfix/master.cf. The following replaces the initial "smtp" entry:

```
#
# Before-filter SMTP server. Receive mail from the network and
# pass it to the content filter on localhost port 10025.
#
smtp      inet  n       -       -       -       20      smtpd
        -o smtpd_proxy_filter=127.0.0.1:10025
        -o smtpd_client_connection_count_limit=10
#
# After-filter SMTP server. Receive mail from the content filter
# on localhost port 10026.
#
127.0.0.1:10026 inet n  -       n       -        -      smtpd
        -o smtpd_authorized_xforward_hosts=127.0.0.0/8
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o smtpd_data_restrictions=
        -o mynetworks=127.0.0.0/8
        -o receive_override_options=no_unknown_recipient_checks
```

(7) Create /etc/postfix/header_checks like this:

```
/X-Spam-Level: \*{7,}/ REJECT Looks like spam to me.
```

(8) Edit /etc/postfix/main.cf to include this:

```
header_checks = pcre:/etc/postfix/header_checks
```

(9) Run "/etc/init.d/spampd start" and "/etc/init.d/postfix restart".

Naturally, other distributions will require variations from this procedure.

- RodRoark

Handling Large Emails

By default, spampd passes through all emails over 65536 bytes. This is set by --maxsize. Increasing this limit can cause problems on systems with very little RAM. Larger email can be blocked by postfix with the main.cf option:

```
message_size_limit = 65536
```

Or you can scan all larger emails via procmail after postfix with the procmail rule:

```
:0fw: spamassassin.lock
* > 65536
| spamassassin

:0:
* ^X-Spam-Status: Yes
spam
```