BetterDocumentation UsageTxt

USAGE

The following is the text of USAGE, based off of revision 264692.

Please feel free to edit it as much as you like to make it more useful. Periodically the version in Subversion will be updated to incorporate some of the changes.

To see the latest version in Subversion, click here

Feel free to write comments about your changes in the Comments section (at the bottom).

```
Important Note For Users Upgrading From Earlier Versions
--------------------------------------------------------

SpamAssassin no longer includes code to handle local mail delivery, as it
was not reliable enough, compared to procmail.  So now, if you relied on
spamassassin to write the mail into your mail folder, you'll have to
change your setup to use procmail as detailed below.

If you used spamassassin to filter your mail and then something else wrote
it into a folder for you, then you should be fine.

Steps to take for every installation:

  - Install Mail::SpamAssassin on your mail server, per the INSTALL
    document.

  - Test it:

      spamassassin -t < sample-nonspam.txt > nonspam.out
      spamassassin -t < sample-spam.txt > spam.out

    Verify (using a text viewer, ie. "less" or "notepad") that nonspam.out
    has not been tagged as spam, and that spam.out has.  The files should
    contain the full text and headers of the messages, the "spam.out"
    message should contain the header "X-Spam-Flag: YES" and be annotated
    with a report from SpamAssassin, and there should be no errors when you
    run the commands.

    Even though sample-nonspam.txt is not spam, nonspam.out will
    contain a SpamAssassin report anyway.  This is a side-effect of
    the "-t" (test) switch.  However, there should be less than 5
    points accumulated; when the "-t" switch is not in use, the report
    text would not be added. For more verbose (debugging) output, add
    the "-D" switch.

    If the commands do not work, DO NOT PROCEED TO THE NEXT STEP, as you
    will lose mail!



If you use KMail:

  - http://kmail.kde.org/tools.html mentions:

    The filter setup is the work of five minutes (if that!) if you have a
    working spamassassin set up.

    The filter in question is "<any header><matches regexp> ."

    The action is "<pipe through> spamassassin"

    Then, in the advanced options, uncheck the "If this filter matches,
    stop processing here" box. If you keep this filter at the top, it will
    analyze any incoming mail, decide whether it's spam or not, and flag
    it accordingly.
```

```
    [Then add] a second filter behind it, which searches for the added
    spam-flags and diverts them into a specific spam folder. [...]

    In the Kmail menu, click on "Message, Create filter" to create a new filter, or "Settings, Configure
filter..." to edit an existing filter.

By creating sub-folders of the Inbox, then creating filters, it is possible to have incoming mail first routed
through SpamAssassin, then sorted by Kmail into incoming folders.

Example: create folders for family, friends, business associates, and spam. Using messages already received, go
to Messages, Create filter..., then select an option for filtering: Filter on subject, filter on from, filter
on to. That will bring up a dialog screen in which you may fine-tune the filter.

If you want to change the behaviour of an existing filter, click on "Settings, Configure filter", then
highlight the filter to edit.

If you use procmail, or haven't decided on any of the above examples:

  - Make a backup of your .procmailrc (if you already have one).

      cp ~/.procmailrc ~/.procmailrc.bak

  - add the line from procmailrc.example to ~/.procmailrc, at the top of
    the file before any existing recipes.

    That'll process all mail through SA, and refile spam messages to
    a folder called "caughtspam" in your home directory.

  - Send yourself a mail message, and ensure it gets to you.  If it does
    not, copy your old backed-up .procmailrc file back into place and ask
    your sysadmin for help!  Here's commands to do that:

      cp ~/.procmailrc.bak ~/.procmailrc
      echo "Help!" | mail root



If you want to use SpamAssassin site-wide:

  - take a look at the notes on the Wiki website, currently at
    <http://wiki.apache.org/spamassassin/UsingSiteWide>.  You will probably
    want to use 'spamd' (see below).

  - *PLEASE* let your users know you've installed it, and how to turn it
    off!   This is our #1 tech support query, and the users are usually
    pretty frustrated once it reaches that stage.

  - *PLEASE* consider setting it up as "off by default" for most accounts,
    and let users opt-in to using it.  Quite a few folks prefer not to
    have their mail filtered, presumably because they don't use their
    email address publically and do not get much spam.

  - Note that procmail users adding spamc to /etc/procmailrc should
    add the line 'DROPPRIVS=yes' at the top of the file.


The Auto-Whitelist
------------------

The auto-whitelist is enabled using the 'use_auto_whitelist' option.
(See http://wiki.apache.org/spamassassin/AutoWhitelist for details on
how it works, if you're curious.)


Other Installation Notes
------------------------


  - Hashcash is a useful system; it requires that senders exercise a
    CPU-intensive task before they can send mail to you, so we give that
    some bonus points.  However, it requires that you list what addresses
```

you expect to receive mail for, by adding 'hashcash_accept' lines to
your ~/.spamassassin/user_prefs or /etc/mail/spamassassin/local.cf
files. See the Mail::SpamAssassin::Plugin::Hashcash manual page for
details on how to specify these.

- SpamAssassin now uses a temporary file in /tmp (or $TMPDIR, if that's
set in the environment) for Pyzor and DCC checks. Make sure that this
directory is either (a) not writable by other users, or (b) not shared
over NFS, for security.

- You can create your own system-wide rules files in
/etc/mail/spamassassin; their filenames should end in ".cf". Multiple
files will be read, and SpamAssassin will not overwrite these files
when installing a new version.

- You should not modify the files in /usr/share/spamassassin; these
will be overwritten when you upgrade. Any changes you make in
files in the /etc/mail/spamassassin directory, however, will
override these files.

- Rules can be turned off by setting their scores to 0 in a
configuration or user-preference file.

- Speakers of Chinese, Japanese, Korean or Arabic may find it useful to
turn off the rules listed at the end of the "user_prefs.template"
file; we've found out that these rules are still triggering on
non-spam CJK mails.

- If you have an unusual network configuration, you should probably
set 'trusted_networks'. This allows SpamAssassin to determine where
your internal network ends and the internet begins, and allows DNS
checks to be more accurate. If your mail host is NATed, you will
almost certainly need to set 'trusted_networks' to get correct
results.

- A very handy new feature is SPF support, which allows you to check
that the message sender is permitted by their domain to send from the
IP address used. This has the potential to greatly cut down on mail
forgery. (see http://spf.pobox.com/ for more details.)   However,
sendmail will not expose the MAIL FROM: sender address by default. So
if you're using sendmail, please add this to /etc/sendmail.cf :

```
H?l?X-Envelope-From: $f
```

- MDaemon users should add this line to their "local.cf" file:

```
report_safe_copy_headers X-MDRcpt-To X-MDRemoteIP X-MDaemon-Deliver-To
```

Otherwise, MDaemon's internal delivery will fail when SpamAssassin
rewrites a message as spam.

- The distribution includes 'spamd', a daemonized version of
SpamAssassin which runs persistently. Using its counterpart,
'spamc', a lightweight client written in C, an MTA can process
large volumes of mail through SpamAssassin without having to
fork/exec a perl interpreter for each message. Take a look in the
'spamd' and 'spamc' directories for more details.

- spamc can now be built as a shared library for use with milters or
to link into other existing programs; simply run "make libspamc.so"
to build this.

```
  - If you get spammed, it is helpful to everyone else if you re-run
    spamassassin with the "-r" option to report the message in question as
    "verified spam".  This will add it to Vipul's Razor, DCC and Pyzor,
    assuming you've set these up appropriately.

      spamassassin -r < spam-message

    If you use mutt as your mail reader, this macro will bind the X key to
    report a spam message.

      macro index X "| spamassassin -r"

    This is, of course, optional -- but you'll get lots of good-netizen
    karma. ;)


  - Quite often, if you've been on the internet for a while, you'll have
    accumulated a few old email accounts that nowadays get nothing but
    spam.  You can set these up as spam traps using SpamAssassin; see the
    ''SPAM TRAPPING'' section of the spamassassin manual page for details.

    If you don't want to go to the bother of setting up a system yourself
    to do this, take a look here [1] for a simple forwarding-based
    alternative.

      [1]: http://wiki.apache.org/spamassassin/SpamTrapping


  - Scores and other user preferences can now be loaded from, and Bayes
    and auto-whitelist data can be stored in, an SQL database; see the
    'sql' subdirectory for more details.

    If you are setting up a large 'spamd' system-wide installation, with
    Bayes and/or auto-whitelists, we strongly recommend using SQL as
    storage.  It has proven more reliable than the default DB_File storage
    backend at several large sites.


  - If you are running SpamAssassin under a disk quota, or are setting up
    'spamd' with users with disk quotas, be warned that the DB_File
    database module used by SpamAssassin for Bayes and AWL storage seems
    to be unreliable in the face of quotas (bug 3796). In this situation,
    we recommend using SQL storage for those databases, instead of DB_File.


  - Lots more ways to integrate SpamAssassin can be read at
    http://wiki.SpamAssassin.org/ .


(end of USAGE)

// vim:tw=74:
```

Comments

Please enter comments here. You can type @"SIG@ to insert your signature. – DuncanFindlay <<DateTime(2005-08-22T02:42:21Z)>>