Kolab SPAM/HAM

Teil 1:

Kolab Groupware - Administration Guide

Teil 2:

Hi Jelle,

unfortunately filtering spam does not work out of the box in Kolab 3.2
but the documentation you are referring is a good starting point.
First you must decide what kind of SPAM/HAM/UNKNOWN_SPAM folders you
like to use; either shared or per user folders.
For my users I use individual folder (each user has its own folders) so
I try to describe it this way.
This process begins with the kolab.conf file where you can set the auto
create folders with the option * autocreate_folders =*
Here's a snippet of my config:
---------snip----------

'Spam': {
'annotations': {
'/private/vendor/kolab/folder-type': "mail.junkemail",
},
},
'Spam/SpamUnknown': {
'annotations': {
'/private/vendor/kolab/folder-type': "mail.junkemail",
},
},
'Spam/NoSpam': {
'annotations': {
'/private/vendor/kolab/folder-type': "mail.junkemail",
},
},
---------snip----------

So, every time I create a mailbox this folders are also created. I
instruct the users to move unrecognized spam mails to folder SpamUnknown
and false spam mails to NoSpam. Recognized spam is moved via sieve rules
to the *Spam* folder. These folders are used to train spamassassin. I
get later to this.

So far so good. To begin using spamassassin it is important to inject
some spam/ham mails to spammassassin's Bayes database. This is described
here [1].
After you train spamassassin you can check the bayes database with

sa-learn --dump magic

The output should be something like this:
0.000 0 3 0 non-token data: bayes db version
0.000 0 2246 0 non-token data: nspam
0.000 0 4336 0 non-token data: nham
0.000 0 401053 0 non-token data: ntokens
0.000 0 993467899 0 non-token data: oldest atime
0.000 0 1404428730 0 non-token data: newest atime
0.000 0 1392945305 0 non-token data: last journal
sync atime
0.000 0 0 0 non-token data: last expiry atime
0.000 0 0 0 non-token data: last expire
atime delta
0.000 0 0 0 non-token data: last expire
reduction count

I'll post my /etc/mail/spamassassin/local.conf to have a reference

###########begin config ####################
These values can be overridden by editing ~/.spamassassin/user_prefs.cf
(see spamassassin(1) for details)

These should be safe assumptions and allow for simple visual sifting
without risking lost emails.

required_hits 5

report_safe 0
rewrite_header Subject [SPAM]
score BAYES_99 7.308
add_header all Status _YESNO_, score=_SCORE_ required=_REQD_
tests=_TESTS_ autolearn=_AUTOLEARN_ version=_VERSION_
add_header all Level _STARS(*)_
add_header all Checker-Version SpamAssassin _VERSION_ (_SUBVERSION_) on
HOSTNAME
#bayes
use_bayes 1
use_bayes_rules 1
bayes_auto_learn 1
bayes_file_mode 0600
bayes_path /etc/spamassassin/bayes
bayes_auto_expire 0
bayes_journal_max_size 15000000
bayes_expiry_max_db_size 20000000

dcc
use_dcc 1
dcc_path /usr/local/bin/dccproc

#pyzor
use_pyzor 1
pyzor_path /usr/bin/pyzor

#razor
use_razor2 1
razor_config /var/spool/amavisd/razor-agent.conf

Blacklist

Whitelist

##############end config#########################
Pyzor and razor must be installed separately. There are good
documentation on the net how to do this.

Now it's time to alter the configuration of /etc/amavisd/amavisd.conf
Here are the relevant parts of the config:

##############begin config#####################

#@bypass_virus_checks_maps = (1); # controls running of anti-virus code
#@bypass_spam_checks_maps = (1); # controls running of anti-spam code
$bypass_decode_parts = 1;   # controls running of
decoders&dearchivers
$mydomain = 'your_primary_domain';
$QUARANTINEDIR = '/var/spool/amavisd/quarantine'; #virus mails are moved
here
$sa_tag_level_deflt = -10; # add spam info headers if at, or above
that level
$sa_tag2_level_deflt = 6.2; # add 'spam detected' headers at that level
$sa_kill_level_deflt = 6.9; # triggers spam evasive actions (e.g.
blocks mail)
$sa_dsn_cutoff_level = 10;   # spam level beyond which a DSN is not sent
$sa_spam_subject_tag = '[SPAM]';
$undecipherable_subject_tag = undef; # this prevents amavisd to tag
encrypted mail with *UNCHECKED* subject
$myhostname = 'FQDN of your server'
$final_virus_destiny   = D_DISCARD;
$final_banned_destiny   = D_BOUNCE;
$final_spam_destiny   = D_PASS; #!!! D_DISCARD / D_REJECT
@local_domains_maps = (["."]); # list of all local domains. this is
important if you have a multi-domain setup
##################end config#########################

I don't have to mention that you have to restart each service after
changing the config files

Now its time to filter recognized Spam mails to the users folder 'Spam'
Here comes sieve in the game. At the moment each user has to create his
own filter but I wrote a script to automate this step. It needs a lil
bit more tweaking but I'll post it on the list when its done.
You can create the filter based on the subject line [SPAM] or based on
the XPAM flag. Here's my sieve roundcube.script

################begin config#####################
rule:[SPAM]
if header :contains "x-spam-flag" "YES"
{
      fileinto "Spam";
}
##################end config#####################

To wrap it up you can test the spam filter by sending (from an external
account) a spam mail [2]

If everything works as expected you can proceed to the next step:
continuously train spamassassin from each users Spam_Unknown and NoSpam folder. Therefore I use a very handy script [3]. This script is run via cron and scans all mailboxes:
##########begin crontab###############

15 2 * * * root /usr/sbin/sa-learn-cyrus

#############end crontab##############

And here are the relevant parts of the config:

###########begin config####################

spam_folder = 'Spam.SpamUnknown'
ham_folder = 'Spam.NoSpam'
remove_spam = yes
remove_ham = no
site_config_path = /etc/mail/spamassassin
bayes_storage = berkely
prefs_file = /etc/mail/spamassassin/local.cf
base_dir = /var/spool/imap
initial_letter = yes # please see config to match your setup
domains = 'example1.com exampl2.com' list all your domains you want to scan
unixhierarchysep = yes # please see config to match your setup
purge_cmd = /usr/lib/cyrus-imapd/ipurge
# Cyrus-IMAPd user
user = cyrus
################end config #################

[1] http://docs.kolab.org/administrator-guide/combating-spam.html#preseeding-the-bayes-database-using-spamassassin-public-corpus

[2] https://spamassassin.apache.org/gtube/

[3] http://www.pollux.franken.de/en/mail-server-tools/

Thats it.

Hope this helps.

Eindeutige ID: #1013
Verfasser: n/a
Letzte Änderung: 2014-07-05 14:21