SpamProbe - фильтруем спам

Опубликовано muff в Сб, 2010-09-11 21:38

В основу почтовых систем устанавливаю, в основном, Exim. С помощью системы фильтров и Greylisting [1] удается отсечь значительную часть СПАМа, однако с каждым днем спамеры все более находчивы, и поток нежелательной корреспонденции постоянно увеличивается. Поэтому решил "прикрутить" антиспам-систему.

Перелопатив информацию, о существующих решениях, нашел такие варианты:

- SpamAssassin
- DSPAM
- SpamProbe

SpamAssassin - самый известный, на мой взгляд. Однако, исходя из того, что SpamAssassin написан на Perl, при большом потоке входящей корреспонденции довольно ощутимо нагружает сервер. К тому же, если письмо с вложенным файлом еще и антивирусом проверяется (например Clamav [2]), то на слабых машинах этот вариант не допустим.

DSPAM - скорость работы DSPAM значительно превосходит скорость работы SpamAssassin. Программа умеет использовать все популярные СУБД для хранения token'ов и работы с ними (включая даже СУБД Oracle), а также может хранить данные в файлах в собственном формате. Следует отметить, что каждому пользователю будет создаваться отдельное хранилище, т.е. обработка спама для каждого пользователя происходит по индивидуальным правилам. Это гораздо лучше с точки зрения точности определения спама, однако более ресурсоемко. Однако, с 2007 года развитие проэкта затормозилось (после приобретения проэкта компанией Sensory Networks), также была информация о вероятности полной остановки развития.

Остановимся все-таки на **SpamProbe** (хотя DSPAM, наверное, тоже скоро протестирую). SpamProbe написан на C++, небольшой и довольно шустрый в работе. К тому же алгоритм работы программы построен на основе математической теоремы Байеса. А ведь именно этот метод статистической фильтрации является наиболее удачным и используется практически всеми спам-фильтрами (в том числе и DSPAM).

Метод Байеса подразумевает использование статистической, оценочной базы, разделенной на две части, одна из которых содержит черный список слов, а другая – белый. При анализе письма подсчитывается количество совпадений каждого отдельного слова (токена) со списками в базе, и на основании этого вычисляется оценка. Оценка эта колеблется в диапазоне от 0 до 1, где значение 0 означает отсутствие признаков спама, 1 – полную уверенность в том, что это спам.

Начинаем установку. Установка будет выполнена из портов:

```
# cd /usr/ports/mail/spamprobe/ && make install clean && rehash
```

После установки необходимо виполнить инициализацию базы данных. Создадим каталог, где будет размещена база данных **SpamProbe**, дадим команду создания базы данных, сделаем владельцем каталога и файлов пользователя exim групы mail (от имени этого пользователя работает Exim):

```
# mkdir /var/db/spamprobe

# spamprobe -d /var/db/spamprobe create-db

# chown -R exim:mail /var/db/spamprobe/
```

Создадим каталог, где будут лежать конфигурационные файлы и скрипты, необходимые для работы **SpamProbe**:

```
# mkdir /usr/local/etc/spamprobe
```

Теперь создадим файл системного фильтра, следующего содержания:

```
# cat /usr/local/etc/spamprobe/exim.filter
 # Exim filter
# Ни в коем случае не удаляйте верхнюю строку
# Путь к лог-файлу
logfile /var/log/spam-filter.log
# Указываем максимальный размер письма, которое будет
# подвержено анализу (512 Кбайт).
add 524288 to n0
# Проверяем письмо только один раз, даже если Exim
# с первого раза его не сможет доставить.
# Нечего плодить заголовки X-SpamProbe:
if first_delivery
        then
                if $message_size is above $n0
                then
                # Если не нужно вести лог-файл, закомментируйте строку
                # ниже
                logwrite "Размер письма $message_id от $return_path превышает $n0 байт; Письмо не
будет проходить проверку"
        else
                headers add "X-SpamProbe: ${run {/usr/local/etc/spamprobe/msgscore.sh \
                        ${message_id} ${quote:${message_headers}}} {$value} {expansion failed}
}"
                # Если не нужно вести лог-файл, закомментируйте строку
                # ниже
                logwrite "$tod_log $message_id FROM $return_path FOR $recipients $value"
                endif
endif
```

Приступаем "рихтовать" файл **/usr/local/etc/spamprobe/msgscore.sh**, на который ссылается системный фильтр. В результате должно получиться следующее:

```
# cat /usr/local/etc/spamprobe/msgscore.sh
#!/bin/sh
# Путь к месту, где почта хранится до момента доставки
spooldir=/var/spool/exim/input
# Путь к базе, где SpamProb хранит токены
```

```
dbdir=/var/db/spamprobe
# Полный путь к программе SpamProbe
path_spamprobe=/usr/local/bin/spamprobe
# Находим полный путь к нашему письму.
# Если бы не опция " split_spool_directory = true "
# в конфигурационном файле Exim он бы соответствовал
# переменной .$spooldir/$1-D.
path_file=`/usr/bin/find $spooldir -name $1-D -print`
echo "$2" > $spooldir/$1-M
sed '1 s/.*//' $path_file >> $spooldir/$1-M
$path_spamprobe -8 -d $dbdir score $spooldir/$1-M
rm $spooldir/$1-M
exit 0
```

Поскольку у меня Exim работает от имени пользователя **exim** группы **mail**, то необходимо изменить права доступа (или сделать этого пользователя владельцем файлов):

```
# chown -R exim:mail /usr/local/etc/spamprobe

# chmod -R 700 /usr/local/etc/spamprobe
```

Теперь добавим в конфигурационный файл Exim упоминание о нашем фильтре (разместить необходимо перед секцией ACL-ей:

```
system_filter = /usr/local/etc/spamprobe/exim.filter

#############################################################################
          ACL CONFIGURATION                                ##         Specifies access c
ontrol lists for incoming SMTP mail      ##############################################
##############################
```

После внесения изменений, необходимо перезапустить Exim:

```
# sh /usr/local/etc/rc.d/exim restart
Stopping exim.
Starting exim.
```

Пора посмотреть, что "пишут" логи спам-фильтра. Отправил два письма: первое больше 512 килобайт (благодаря вложению), а второе - меньше. В результате можно наблюдать следующее:

```
# tail -f /var/log/spam-filter.log
SPAM FILTER: Размер письма от admin 'yxo' muff.kiev.ua превышает 524288 байт; Письмо не будет проходить проверку
SPAM FILTER: Письмо от admin 'yxo' muff.kiev.ua успешно прошло проверку; GOOD 0.3000000 c27434269f5ae537df6a44a2459eef8a
```

А в заголовках письма теперь можно обнаружить такую запись:

```
X-SpamProbe: GOOD 0.3000000 c27434269f5ae537df6a44a2459eef8a
```

Кажется все продвигается неплохо. Стоит отметить, что на данном этапе все письма будут промаркированы, как GOOD, поскольку еще не началось "обучение" **SpamProbe**.

Следующий этап - настройка в Exim обработки промаркированных писем. Для этого в секцию **ROUTERS CONFIGURATION** необходимо добавить такие блоки:

```
#############################################################################
         ROUTERS CONFIGURATION                            ##               Specifies ho
w addresses are handled                  ##############################################
##############################     THE ORDER IN WHICH THE ROUTERS ARE DEFINED IS IMPO
```

```
RTANT!       ## An address is passed to each router in turn until it is accepted.  #
#############################################################################

begin routers

### SpamProbe start ###
SP_spam_router:driver = acceptdomains = +local_domainslocal_part_prefix = spamtransport = SP_
spam_transport

SP_no_spam_router:driver = acceptdomains = +local_domainslocal_part_prefix = no-spam
transport = SP_no_spam_transport### SpamProbe end ###

dnslookup:  driver = dnslookup  domains = ! +local_domains  transport = remote_smtp
 ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8  no_more

### SpamProbe start ###
SP_check_router:driver = acceptdomains = +local_domains# Расскоментировать строку, если не
обходима проверка на СПАМ только # для определенных получателей#recipients = lsearch;/us
r/local/etc/spamprobe/userscondition = ${if and {{match{$h_X-SpamProbe:}{SPAM}} \
 {!match_address{$sender_address} \ {lsearch;/usr/local/etc/spamprobe/whitelist}}}transport = S
P_check_transportno_more### SpamProbe end ###
```

Пожалуйста, ничего не перепутайте, поскольку важна очередность роутеров. Теперь распишем, что за что отвечает...

Роутеры **SP_spam_router** и **SP_no-spam_router** необходимы для обучения нашего спам-фильтра. «Не ошибается тот, кто ничего не делает» – гласит пословица, так и в нашем случае. Если фильтр допустил ошибку и вместо того, чтобы пометить письмо как SPAM, пометил его GOOD, перешлите это письмо как вложение на адрес **spam@(ваш домен)**, и вы увидите, что в следующий раз это письмо будет промаркировано правильно. Аналогично поступаем в обратной ситуации, но пересылать письмо теперь будем на адрес **no-spam@(ваш домен)**.
Роутер **SP_check_router** собственно выполняет проверку и дает оценку письму путем добавления в тело заголовка X-SpamProbe. Также стоит обратить внимание на файл **/usr/local/etc/spamprobe/whitelist** - в нем находится список "белых" адресов, почта с которых не будет спамом.

Синтаксис файла очень прост - по одному аккаунту на строке:

```
# cat /usr/local/etc/spamprobe/whitelist

# Список адресов, которым можно доверять,
# и не анализировать почту, приходящую от них
username [at] ukr [dot] net
meren [at] mail [dot] ru
office [at] firma [dot] com
```

Синтаксис файла **/usr/local/etc/spamprobe/users** такой же (не забудьте создать этот файл, если планируется проверка только для определенных пользователей).

Кстати, дополнительно несколько примеров проверки условий (параметр **condition** в **SP_check_router**). Итак, если необходимо выполнять проверку на СПАМ для всех пользователей без исключений, строка имеет следующее значение:

```
condition = ${if and {{match{$h_X-SpamProbe:}{SPAM}}
```

Если необходимо выполнять проверку для всех, кроме определенных пользователей

(перечисленных в файле **/usr/local/etc/spamprobe/whitelist**, строка приобретает значение:

```
condition = ${if and {{match{$h_X-SpamProbe:}{SPAM}} \
 {!match_address{$sender_address} \ {lsearch;/usr/local/etc/spamprobe/whitelist}}}}
```

Если же есть необходимость не выполнять проверку, если отправителем является локальный пользователь, то необходимо воспользоваться следующим правилом:

```
condition = ${if and {{match{$h_X-SpamProbe:}{SPAM}} \ {!match_domain{$sender_addres
s_domain}{+local_domains}}}}
```

Также есть возможность выполнять проверку всех отправителей, кроме отправителей из "белого" списка и отправителей, являющихся локальными пользователями:

```
condition = ${if and {{match{$h_X-SpamProbe:}{SPAM}}{!match_address{$sender_address}
 / {lsearch;/usr/local/etc/spamprobe/white-address}} / {!match_domain{$sender_addres
s_domain}{+local_domains}}}}
```

Вариантов много. На каком остановиться - решать Вам...

Итак, продолжим. Пора добавить непосредственно сам транспорт (в секцию **TRANSPORTS CONFIGURATION**). В моем частном случае, всю корреспонденцию, которая отсеивалась, необходимо было складывать в отдельный почтовый ящик (пользователь spamfilter [at] domain [dot] com):

```
#########################################################################
         TRANSPORTS CONFIGURATION                       ##############################
##########################################                      ORDER DOES NOT MAT
TER                          ##     Only one appropriate transport is called for each
delivery.    #########################################################################

begin transports

### SpamProbe start ###
SP_check_transport:driver = appendfilemaildir_format = truedirectory = /var/exim/domain.com
/spamfilter

SP_spam_transport:driver = pipecommand = "/usr/local/bin/spamprobe -d /var/db/spampr
obe spam"return_path_add = falsereturn_fail_output = truelog_output = trueuser = exi
mgroup = mail

SP_no_spam_transport:driver = pipecommand = "/usr/local/bin/spamprobe -d /var/db/spa
mprobe good"return_path_add = falsereturn_fail_output = truelog_output = trueuser =
eximgroup = mail### SpamProbe end ###
```

В разделе **TRANSPORTS CONFIGURATION** последовательность не настолько важна, как в **ROUTERS CONFIGURATION**.

В результате, письма, отмеченные как SPAM, отправляются пользователю spamfilter. В случае ошибки спам-фильтра всегда можно заглянуть в нее и поискать пропавшее письмо.

На этом базовую настройку можно считать оконченной.

При написании статьи использовались материалы статьи ***Павла Литвинова*** "Фильтруем спам в Exim с помощью SpamProbe", опубликованной в журнале **"*Системный администратор*" *№5, май 2008г***.

Источник (получено *2026-05-05 12:02*): http://muff.kiev.ua/content/spamprobe-filtruem-spam

Ссылки:
[1] http://muff.kiev.ua/content/greylisting-pora-poznakomitsya-s-tekhnologiei-serykh-spiskov
[2] http://muff.kiev.ua/content/clamav-antivirusnaya-zashchita-servera