Владимир Ануфриев

Прозрачный SMTP-мост для борьбы со спамом и вирусами

Transparent SMTP bridge for antispam and AV solutions

Введение

Задача интеграции программ для борьбы со спамом и вредоносными программами с почтовыми серверами, не имеющими для этого открытых интерфейсов, при современном состоянии сети Интернет возникает для организаций любого масштаба в обязательном

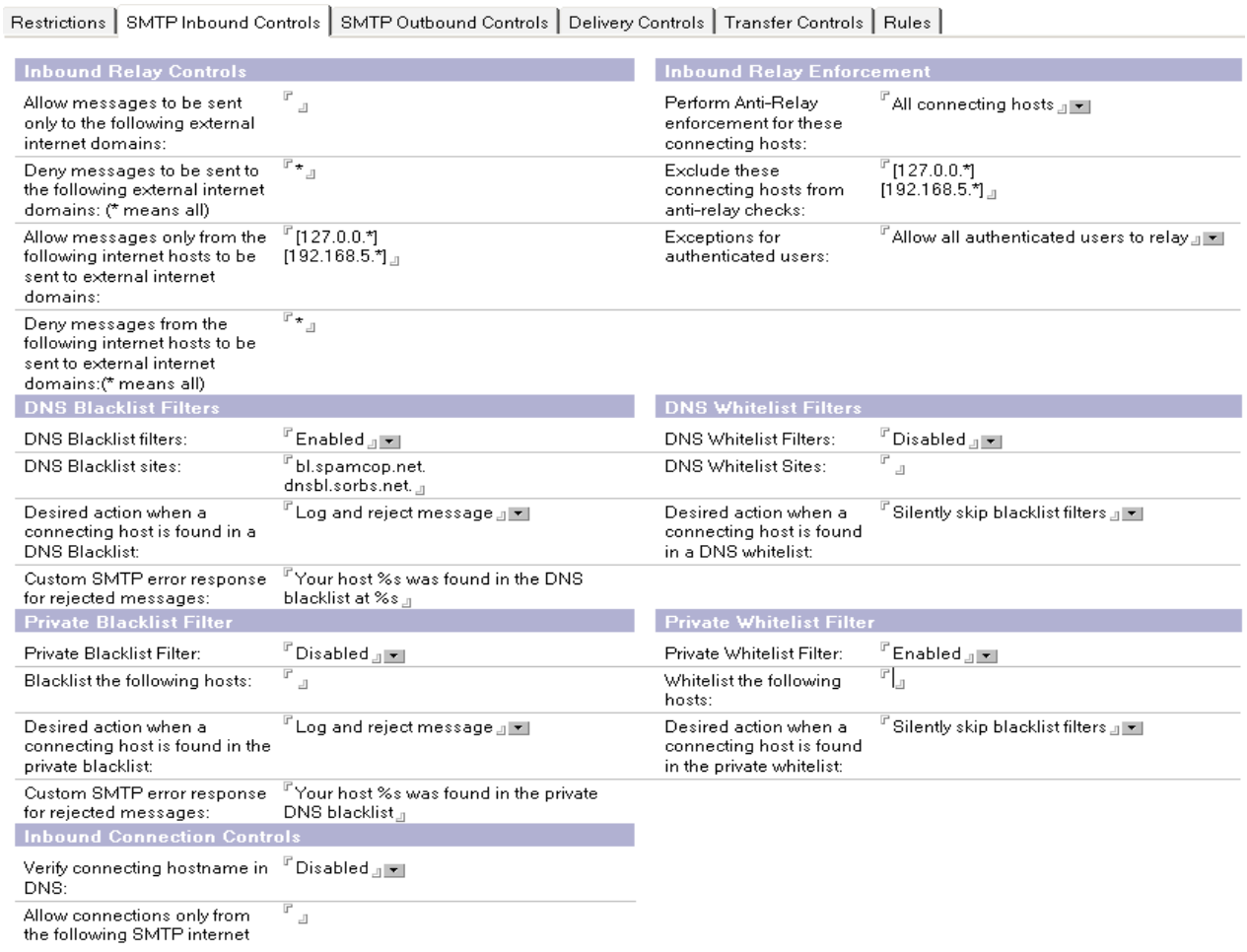

Рис. 1. Часть настроек контроля входящей SMTP почты сервера Lotus Domino 8.5

порядке, если организация использует в качестве почтового сервера коммерческое программное обеспечение. В частности, SMTP-сервер, встроенный в IBM Lotus Domino[1], снабжен некоторыми возможностями для отбраковки нежелательной корреспонденции,

1 http://www-142.ibm.com/software/sw-lotus/products/product4.nsf/wdocs/dominohomepage

однако эти возможности при ближайшем рассмотрении весьма ограничены.

Если посмотреть на соответствующие настройки сервера последней версии 8.5, частично представленные на рисунке 1, то видно, что можно ограничивать адреса и имена отправителей и получателей, использовать статические и динамические черные и белые списки и задавать несложные правила обработки почты, в том числе запрещать транзит.

Теоретически, с помощью Notes API[2] можно написать сколь угодно сложные расширения для изощренной обработки входящих и исходящих писем. Такие коммерческие решения существуют. Например, на этом приципе построена программа компании Trend Micro[3]. Есть и попытки написать бесплатный софт на том же принципе. Например, можно посмотреть проект OpenMailScan[4]. Однако, этот подход помимо сложностей в программировании и отладке программы промежуточного слоя на языке Си требует хороших знаний специализированного API и внутренней архитектуры Domino, которая отнюдь не отличается простотой. Приблизительно тоже самое можно сказать и о других почтовых серверах и соответствующих решениях. Кроме того, коммерческие решения в этой сфере далеко недешевы и обычно привязаны к конкретному почтовому продукту.

Что же делать, если хочется получить как можно более дешевое и в то же время относительно качественное решение?

В данной статье рассматривается применение доработанной программы ProxSMTP[5] для предварительной прозрачной для используемого почтового сервера фильтрации входящей почты. Работоспособность доработанной программы проверена в комплексе с сервером Lotus Domino 7 в среде OpenSuSE Linux 10.3 и 11.1 на платформе x86.

Выбор базового варианта программы

Из простых вариантов решения проблемы борьбы со спамом для Domino можно предложить попробовать поупражняться с правилами обработки почты и написанием обрабатывающих агентов на Lotus Script. Эта техника рассмотрена в статье в журнале Domino Power Magazine[6].

Реализовать с помощью таких средств хотя бы некое подобие возможностей стандартного решения Spam Assassin[7] (далее SA) будет весьма непросто. Чтобы получить представление о возможных результатах можно посмотреть на kSpam[8]. В нем, правда, задействовано и программирование на языке Си.

Стандарных возможностей для прикручивания SA или ClamAV[9] к Domino, подобных

2 http://www-128.ibm.com/developerworks/lotus/downloads/toolkits.html – Notes C Api
3 Пакет ScanMail Suite для IBM Lotus Domino http://ru.trendmicro.com/ru/products/enterprise/scanmail-for-lotus-domino/index.html
4 Free domino server addin for scanning email attachments for viruses. Integrates your well known command line virusscanner in domino, vendor independent - http://wikifaq.openntf.org/Projects/pmt.nsf/ProjectLookup/MailScan
5 http://memberwebs.com/stef/software/proxsmtp/
6 **Daniel Koffler. FIGHT BACK AGAINST SPAM.** Hacking Domino mail rules to fight spam http://www.dominopower.com/issues/issue200312/00001158001
7 The Apache SpamAssassin Project - http://spamassassin.apache.org/
8 A spam filter for Lotus Domino, including rule based and Bayesian filtering. http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/kSpam
9 http://www.clamav.net/

интерфейсу Milter[10] для почтового сервера Postfix[11], не существует. Можно, конечно, использовать SA в комплексе с отдельным транзитным сервером (relay), поддерживающим Milter, но тогда пропадает возможность применения соответствующей функциональности Domino, которую мы рассмотрели выше, а кроме того, становятся чрезвычайно малоинформативными журналы почтового сервера, т.к. все письма начинают поступать к нему с одного и того же адреса – адреса транзитного сервера-фильтра. В условиях большого количества входящей корреспонденции пропускать каждое письмо через SA без предварительной отбраковки очевидно некорректных обращений нам представляется малоэффективным.

К счастью, еще в 2006 году была написана программа ProxSMTP[12], которая делает почти все, что нам нужно, то есть, может почти прозрачно обрабатывать входящий SMTP трафик, передавая необходимые запросы и ответы SMTP между обратившимся клиентом и принимающим сервером, а также вызывать внешнюю программу после получения тела письма, но до передачи его серверу.

Единственная проблема заключается в том, что при предполагаемом автором этой программы способе её применения мы опять же теряем информативность журнала почтового сервера и не можем использовать большую часть его встроенных фильтров, так как обращения к нему идут с одного и того же адреса. Последнее является следствием того, что Domino не понимает команды XCLIENT из расширений ESMTP[13].

Доработка прокси-сервера

К счастью, в Linux есть мощное средство перенаправления пакетов, встроенное в модуль ядра netfilter[14], а исходный текст ProxSMTP очень небольшой и вполне понятный, он не требует изучения сложнеших API и использует небольшой набор стандартных функций.

Поэтому вспомним немножко язык Си и сначала подрихтуем ProxSMTP. Мы добавим в него возможность, которую назовем SuperProxy. Дополнительная функциональность, включающаяся соответствующей новой опцией конфигурации, заключается в синхронных вызовах программы iptables – пользовательского интерфейса модуля ядра netfilter, которая задаст правила переброски пакетов между нашим сервером-фильтром и почтовым сервером.

Желаемая схема прохождения пакетов представлена на рисунке 2.

Клиент обращается к нашему SMTP-серверу по порту 25 (SMTP). Netfilter перебрасывает входящие пакеты на порт 10025 (можно использовать любой свободный порт), который слушается сервером-фильтром proxsmtpd. Фильтр прозрачно передает кадры протокола SMTP на почтовый сервер, получает от последнего ответы и переправляет их клиенту до тех пор, пока не начинается передача тела письма. При передаче пакетов от фильтра к почтовому серверу netfilter подставляет в качестве источника реальный адрес отправителя, а не адрес, с которого ведет отправку пакета фильтр. Таким образом, почтовый сервер не видит, что с ним общается не непосредственно клиент, а некоторый промежуточный сервер. Все возможности сервера в этом случае задействованы. В частности, почтовый сервер может проверить адрес отправителя в черных списках и прекратить сеанс

10 http://ru.wikipedia.org/wiki/Milter
11 http://www.postfix.org/
12 http://memberwebs.com/stef/software/proxsmtp/
13 http://ru.wikipedia.org/wiki/SMTP
14 http://www.netfilter.org/

на ранней стадии, что позволяет не занимать канал передачи данных бесполезным трафиком. В случае тарификации входящего трафика одно это уже может дать существенную экономию пользователю.

Как только начинается передача тела письма, фильтр сохраняет его в файле и запускает его проверку с помощью SpamAssassin, ClamAV или любой другой внешней программы. Если проверка дает положительный результат (обнаружен спам, вирус или другая нежелательная информация), то сеанс с клиентом прекращается, а сервер информируется о том, что клиент как бы прекратил сеанс по своей инициативе.

Посмотрим, что же нужно изменить в исходных текстах ProxSMTP, чтобы добиться нужного результата.

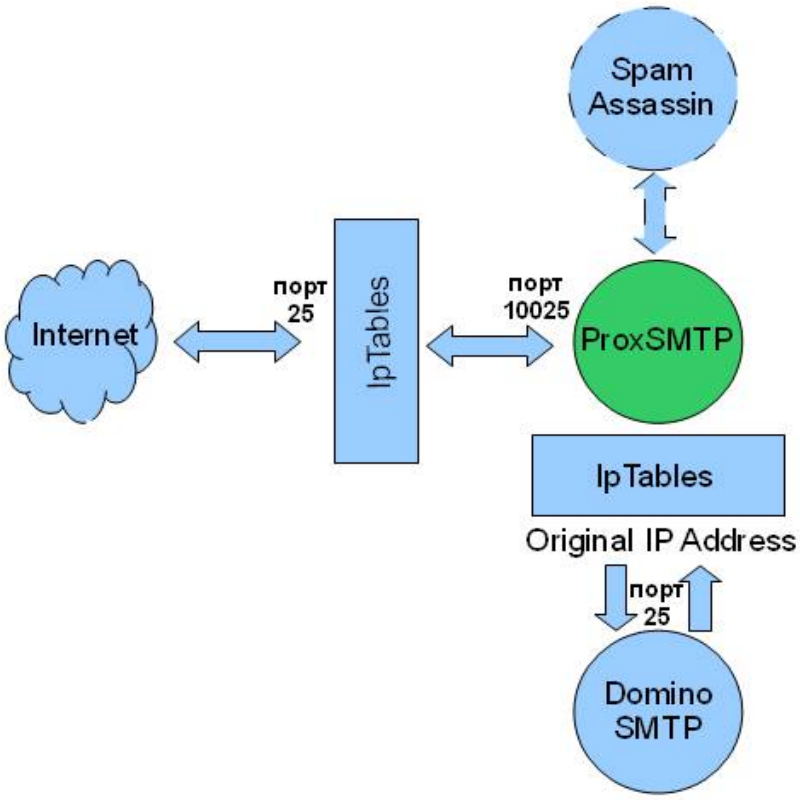

Рис. 2. Схема прохождения входящего SMTP потока в режиме SuperProxy

Развернем архив proxsmtp-1.8.tar.gz и перейдем в подкаталог common. Изменения будут касаться пяти файлов в этом каталоге. Начнем с заголовочных.

Листинг 1. Файл common/smtppass.h

```
--- common/smtppass.h 2008-05-27 16:44:53.000000000 +0400
+++ common/smtppass.h 2009-08-13 13:14:51.031250000 +0400
@@ -134,6 +134,13 @@
     char* xforwardaddr;             /* The IP address proxied for */
     char* xforwardhelo;             /* The HELO/EHLO proxied for */

+#ifdef SUPERPROXY
+    struct sockaddr_any peer;       /* Client address */
+    struct sockaddr_any proxy;      /* Proxy local bound address */
+    struct sockaddr_any smtpd;      /* Server address */
+    int                 masq;       /* Flag for successful masquerading */
+#endif
+
     int _crlf;                      /* Private data */
 }
 spctx_t;
```

Из листинга файла различий между оригиналом и нашим вариантом кода (см. листинг 1) видно, что мы добавили несколько полей в структуру spctx_t.

Листинг 2. Файл common/sock_any.h

```
--- common/sock_any.h 2007-05-27 04:29:08.000000000 +0400
+++ common/sock_any.h 2009-08-13 13:14:51.062500000 +0400
@@ -63,6 +63,14 @@
 #define SANY_ADDR(any)  ((any).s.a)
 #define SANY_LEN(any)   ((any).namelen)
 #define SANY_TYPE(any)  ((any).s.a.sa_family)
+
+#ifdef SUPERPROXY
+#define SANY_PORT(any)  ((any).s.in.sin_port)
+#define SANYP_ADDR(anyp)  ((anyp)->s.a)
+#define SANYP_LEN(anyp)   ((anyp)->namelen)
+#define SANYP_TYPE(anyp)  ((anyp)->s.a.sa_family)
+#define SANYP_PORT(anyp)  ((anyp)->s.in.sin_port)
+#endif

 int sock_any_pton(const char* addr, struct sockaddr_any* any, int opts);
```

В файл sock_any.h (см. листинг 2) мы добавили для собственного удобства несколько макроопределений.

Листинг 3. Файл common/sppriv.h

```
--- common/sppriv.h   2007-05-27 04:28:21.000000000 +0400
+++ common/sppriv.h   2009-08-13 13:14:51.109375000 +0400
@@ -49,6 +49,10 @@
     struct timeval timeout;         /* Timeout for communication */
     int keepalives;                 /* Send server keep alives at this interval */
     int transparent;                /* Transparent proxying */
+#ifdef SUPERPROXY
+    int spoofing;                   /* Address spoofing */
+    const char* iptables;           /* iptables binary */
+#endif
     int xclient;                    /* Send XFORWARD info */
     const char* directory;          /* The temp directory */
     const char* user;               /* User to run as */
```

В файле sppriv.h (см. листинг 3) мы добавили пару полей в структуру spstate_t.

Некоторые изменения, связанные с дополнительными опциями конфигурации, необходимо внести в файл common/smtppass.c. Так же необходимо отметить, что несколько экземпляров программы iptables не могут работать одновременно, поэтому их нужно вызывать последовательно, что в нашем многонитевом (многотридовом) приложении требует применения семафоров. Изменения в файле common/smtppass.c уже настолько большие, что мы вынесли соответствующий файл с расхождениями в приложение 1, чтобы не перегружать текст самой статьи.

И наконец, наибольшие изменения требуется внести в файл spio.c (см. приложение 2). В этом файле, собственно, и запрограммирован вызов iptables с необходимыми параметрами.

Теперь, чтобы собрать программу, нужно не забыть добавить в Makefile для компилятора Си ключ -DSUPERPROXY. В конечном итоге CFLAGS будет выглядеть примерно следующим образом (с учетом нужд отладки):

```
CFLAGS =  -I/usr/local/include -pthread -D_POSIX_PTHREAD_SEMANTICS -DNO_CPP_DEMANGLE -g -DSUPERPROXY
-ggdb -rdynamic
```

Формат статьи не позволяет привести здесь все изменения с точностью до знака, но вышепреведенного полностью достаточно для того, чтобы самостоятельно собрать программу.

Для того чтобы получить из исходных файлов необходимые варианты необходимо перейти в подкаталог proxsmtp-1.8, сохранить приведенные в статье файлы расхождения в файлы с расширением .diff и запустить пять раз программу patch -p0 < имя_файла для каждого файла. После чего запустить configure и выполнить команду make:

```
# patch -p0 common/smtppass.h < smtppass.h.diff
# patch -p0 common/sock_any.h < sock_any.h.diff
# patch -p0 common/sppriv.h < sppriv.h.diff
# patch -p0 common/smtppass.c < smtppass.c.diff
# patch -p0 common/spio.c < spio.c.diff
# export CFLAGS=-DSUPERPROXY; ./configure
# make
```

В следующем параграфе рассмотрим как установить, настроить и интегрировать программу с нашим окружением.

Настройка и интеграция

Получившуюся в результате некоторых усилий программу proxsmtpd установим в каталог /usr/local/sbin (можно установить в любой другой, согласно вашим предпочтениям). В каталоге /usr/local/etc создадим файл proxsmtpd.conf, примерный вид которого приведен в листинге 4. Описание опций можно прочитать с помощью man 5 proxsmtpd.conf, если вы установили прикладывающийся к поставке ProxSMTP файл proxsmtpd.conf.5 в нужный каталог.

Листинг 4. Файл /usr/local/etc/proxsmtpd.conf

```
# ------------------------------------------------------------------------------
#                          PROXSMTPD CONFIG FILE
# ------------------------------------------------------------------------------
#
# - Comments are a line that starts with a #
# - All the options are found below with sample settings
# The Filter Command run for each email. See 'man proxsmtpd' for details
# The following command is a simple which just creates temp files.
FilterCommand: /usr/local/bin/spamassassin.sh
# The amount of time to wait for data from FilterCommand
FilterTimeout: 300
# The type of filter ('pipe' to pipe data through filter,
# or 'file' to pass a file to the filter)
FilterType: file
# The maximum number of connection allowed at once.
# Be sure that clamd can also handle this many connections
MaxConnections: 256
# Amount of time (in seconds) to wait on network IO
TimeOut: 60
# A header to add to all scanned email
Header: X-Filtered: Scanned from %i by ProxSMTP
# Keep Alives (ie: NOOP's to server)
```

```
KeepAlives: 3
# Send XCLIENT commands to receiving server (Domino not recognize it)
XClient: off
# Address to listen on (defaults to all local addresses on port 10025)
Listen: yourhost.com:10025
# Directory for temporary files
TempDirectory: /tmp/proxsmtp
# Enable transparent proxy support
TransparentProxy: on
# User to switch to
#User: mail
SuperProxy: on
IpTables: /usr/sbin/iptables
```

Наши добавления к стандартной конфигурации попали в последние две строки файла. Они не требуют дополнительных комментариев.

Если сейчас запустить proxsmtpd,

```
#  proxsmtpd -d 4 -p /var/run/proxsmtp/proxsmtpd.pid
```

то в предположении, что почтовый сервер у вас уже работает, ничего не изменится. Для того чтобы пакеты начали приходить на вход нашему фильтру, нужно дать решающую команду iptables. Но прежде чем это сделать, опишем всю схему обработки входящей почты. Как ни странно, она получается достаточно сложной и требует подготовки еще как минимум одного файла, обозначенного на рисунке 3 как Script 1.

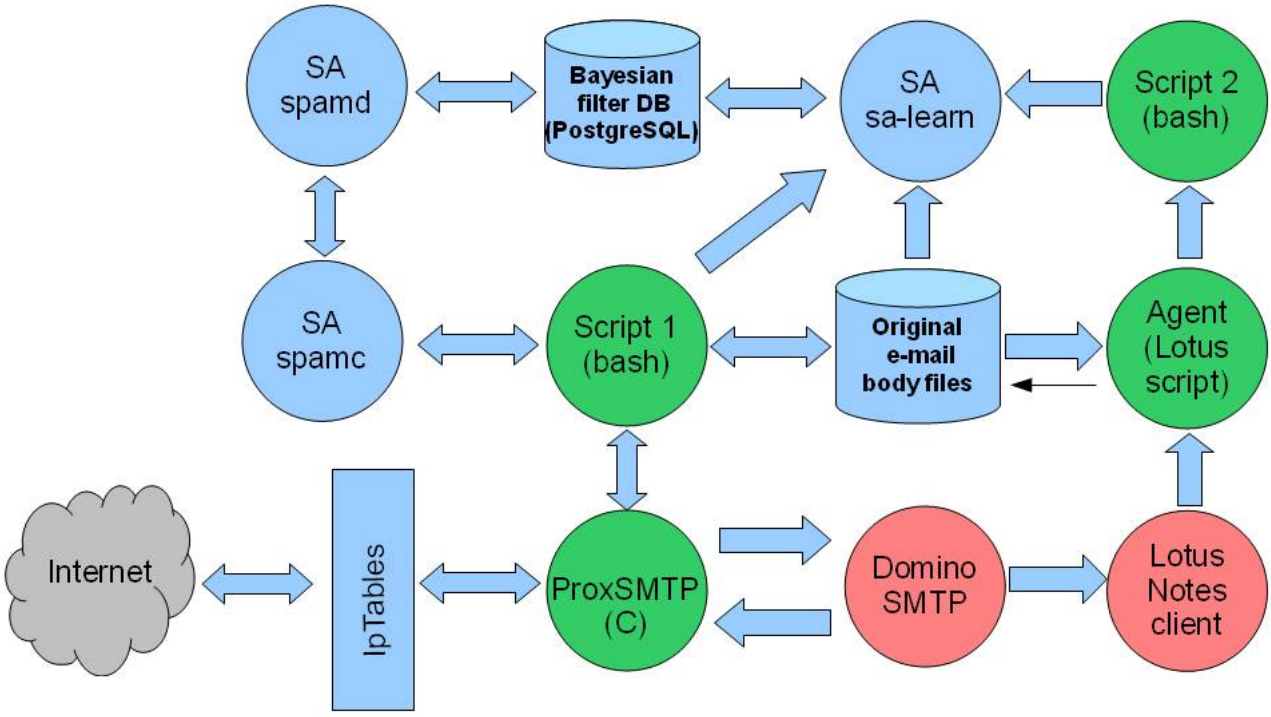

Рис. 3. Общая схема обработки входящей почты в антиспамерном решении на базе модифицированного ProxSMTP и SpamAssassin с Lotus Domino в качестве почтового сервера.

На схеме на рисунке 3 наши разработки показаны зеленым цветом, свободное программное обеспечение — голубым, а коммерческое — розовым.

Мы не описываем здесь установку и настройку почтового сервера, сервера базы данных и антиспамерного комплекса SpamAssassin. Предполагается, что все эти компоненты установлены в соответствии с документацией и работают в штатном режиме.

Итак, пакеты перенаправляются модулем ядра netfilter нашему фильтру ProxSMTP (этот процесс стартует как только мы дадим специальную команду, которая приведена ниже). ProxSMTP перенаправляет команды протокола SMTP почтовому серверу, а получаемые от него ответы — клиенту, до тех пор, пока не начнется передача тела письма командой DATA. Единственное отличие нашего доработанного режима SuperProxy от штатного заключается в том, что прежде чем начать обработку, ProxSMTP выполняет две настройки netfilter:

```
iptables -t nat -I -p tcp -s «фильтр» --sport «порт» -d «сервер» --dport 25 -j SNAT \
--to-source «клиент»
iptables -t nat -I -p tcp -s «сервер» --sport 25 -d «клиент» --dport «порт» -j DNAT \
--to-destination «фильтр»
```

Здесь «фильтр» - это порт, с которого фильтр пересылает пакеты серверу, «сервер» - это ip-адрес почтового сервера, «клиент» - оригинальный ip-адрес подключившегося клиента.

С помощью netfilter почтовый сервер получает пакеты с адресом оригинального клиента, таким образом, он может использовать все свои защитные механизмы и правильно формировать журналы. При пересылке ответных пакетов, они перехватываются и направляются фильтру ProxSMTP, и только он уже, в свою очередь, пересылает их клиенту, если нужно меняя или корректируя их содержимое в соответствие с собственным алгоритмом.

При разрыве соединения ProxSMTP восстанавливает таблицы netfilter, выдавая две симметричные вышеописанным команды:

```
iptables -t nat -D -p tcp -s «фильтр» --sport «порт» -d «сервер» --dport 25 -j SNAT \
--to-source «клиент»
iptables -t nat -D -p tcp -s «сервер» --sport 25 -d «клиент» --dport «порт» -j DNAT \
--to-destination «фильтр»
```

Когда начинается передача тела письма от клиента по команде DATA протокола SMTP, фильтр сохраняет полученное письмо в файле (на рис. 3 хранилище этих файлов обозначено как Original e-mail body files) и вызывает специальный скрипт, обозначенный на схеме как Script 1. В нашем случае Script 1 написан на языке Bash (GNU Bourne-Again SHell), но в целях повышения эффективности его можно написать на Perl или даже на Си.

Пример текста скрипта Script 1 приведен в приложении 4. В нашей реализации у него помимо описанных в документации на ProxSMTP есть еще два параметра: путь к хранилищу файлов Original e-mail body files (константа JUNKMAILDIR) и граничное значение (константа HARDSPAMLEVEL). Скрипт перенаправляет полученное письмо на проверку SA через вызов программы spamc. В случае, если возваженный уровень «спамовости» (score level) меньше -50, то со всей очевидностью, это совершенно нормальное письмо, и мы можем

обучить на его примере байесовский фильтр[15] с помощью выполнения sa-learn -ham. Если возвращенный spamc уровень меньше HARDSPAMLEVEL, то мы передаем это письмо почтовому серверу, который может, в свою очередь сделать дополнительные проверки, как показано ниже. В случае, если уровень больше HARDSPAMLEVEL, мы через фильтр сообщаем клиенту, что он передает явный спам, разрываем с ним соединение и обучаем байесовский фильтр с помощью выполнения sa-learn -spam.

В то же время SA может быть настроен на другой порог, меньший или равный HARDSPAMLEVEL. Это задается в конфигурационном файле /etc/mail/spamassassin/local.cf:

```
# Set the score required before a mail is considered spam.
required_score 8.00
required_hits  8.0
```

К этому моменту сервер SA должен уже работать (запуск производится командой /etc/init.d/spamd start, а автоматический запуск при старте операционной системы можно сконфигурировать командой chkconfig spamd on). Почтовый сервер также работает и ни о чем не подозревает.

Рис. 4. Пример окна почтового сервера с входящими письмами

Вот теперь, когда основные элементы нашей системы настроены, можно запустить фильтрацию, перенаправив входящие на порт 25 (SMTP) пакеты фильтру (в нашем случае внешний интерфейс компьютера называется eth1):

```
/usr/sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 25 -j REDIRECT --to-port 10025
```

15 **Байесовский фильтр** (Bayesian filter) — спам-фильтр, использующий статистический алгоритм, созданный английским математиком Т. Байесом (T. Bayes). Способен к автоматическому дообучению. Spam Assassin имеет встроенный байесовский фильтр, который может опционально быть включен или выключен.

Теперь в интерфейсе почтового сервера мы можем лицезреть только прорвавшиеся через наш фильтр письма. В случае, если их score level превосходит установленный в конфигурации SA, их темы будут изменены и помечены как спам. Пример окна почтового сервера с входящими письмами представлен на рисунке 4.

Дальнейшие настройки фильтра проводятся с помощью параметров конфигурации почтового сервера и SA. Над почтовым клиентом мы еще поработаем, а сейчас автоматизируем запуск и останов нашего фильтра.

В Linux запуск и остановка сервера ProxSMTP производится скриптом, который мы разместим в /etc/init.d. Полный пример такого скрипта приведен в приложении 3. Теперь можно запустить proxsmtpd с помощью команды /etc/init.d/proxsmtpd start, а остановить с помощью команды /etc/init.d/proxsmtpd stop. Необходимые команды iptables будут выполнены автоматически.

Чтобы хранилище файлов сообщений не переполнялось можно его периодически в автоматическом режиме.чистить от старых файлов. Для этого нужно в каталог /etc/cron.daily скопировать, к примеру, такой скрипт (см. листинг 5):

Листинг 5. Файл clear_junkmail.sh

```
#!/bin/bash
JUNKMAIL=/opt/lotus/junkmail
# a week
MAXTIME=604800
today=`date +%s`
for FILE in $JUNKMAIL/* ; do
  mtime=`stat -c %Y $FILE 2> /dev/null`
  if [ $? -eq 0 ] ; then
    differ=$(($today - $mtime))
#   echo $differ
    if [ $differ -ge $MAXTIME ] ; then
      rm -fv $FILE
    fi
  fi
done
```

В результате его ежедневного исполнения в каталоге $JUNKMAIL файлы будут храниться не дольше недели.

Настройка интерфейса почтового сервера

Вручную отбирать письма, помеченные как спам, в большинстве почтовых клиентов не требуется, так как в них предусмотрена настройка соответствующей системы правил разбора входящей почты.

В случае почтового клиента Lotus Notes правила разбора входящей почты можно задать через меню Tools->Mail Rules. Альтернативные способы перехода к редактированию

правил представлены на рисунке 5.

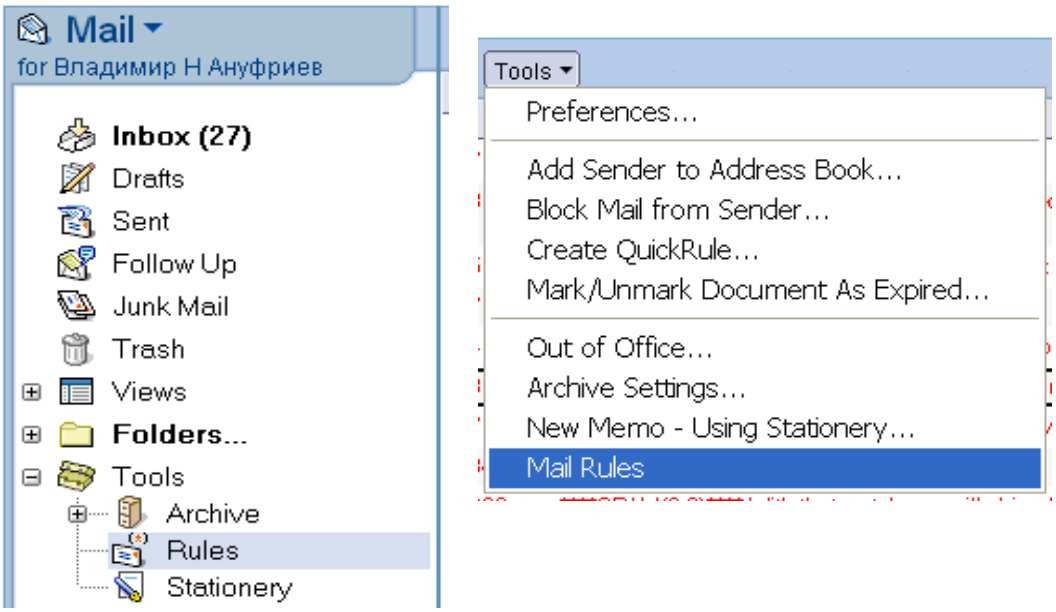

Рис. 5. Вызов настройки правил разбора почты из меню Lotus Notes

Создаем простое правило (см. рис. 6), по которому все письма с пометкой «SPAM» из Inbox перекладываются в папку Junk Mail. При этом письма, пришедшие от других пользователей вашего домена Lotus Notes, не затрагиваются. Этим достигается возможность пересылки друг другу любых писем, в том числе помеченных.

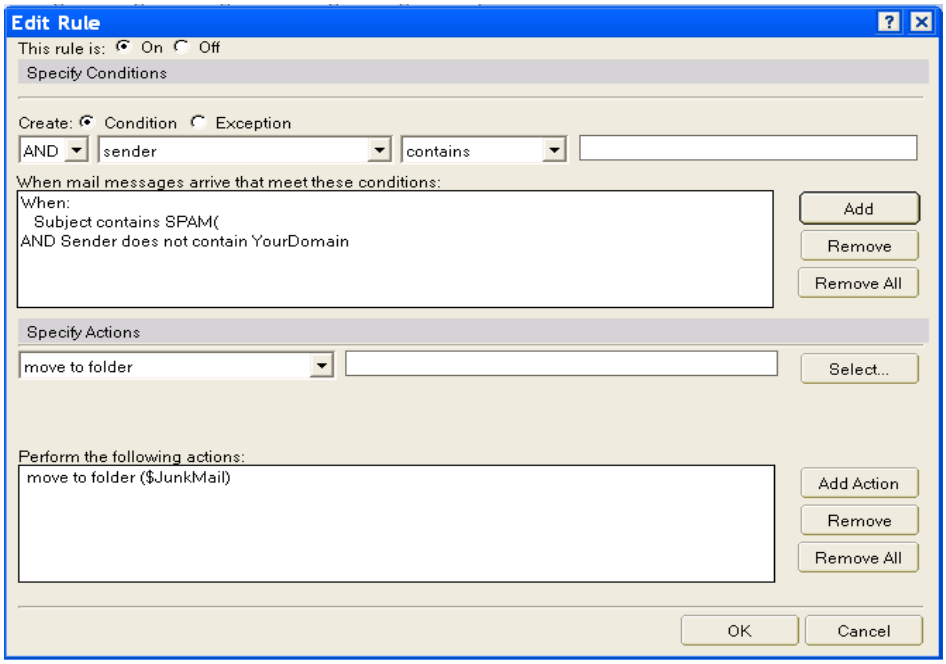

Рис. 6. Создание правила автоматического переноса подозрительных писем в папку Junk Mail

Теперь в папке Inbox пользователь видит только письма, прошедшие все проверки.

Бывают случаи, когда SA ошибается (с его настройками можно экспериментировать бесконечно долго, их сотни). Поэтому мы ввели зазор между письмами, которые он считает спамом и теми, что убиваются еще до передачи почтовому серверу. Письма со score level, попавшим в этот зазор, оказываются в папке Junk Mail. Прежде чем их удалять пользователь может их бегло просмотреть, чтобы убедиться, что ничего ценного не потерялось.

Убедившись, что в папке Junk Mail остались только ненужные письма, и добавив к ним единицы, прорвавшиеся через наши заслоны, желательно бы указать SA, что письма такого характера в дальнейшем нежелательны. К сожалению, Lotus Notes не имеет готовых интерфейсов с SA, поэтому мы вынуждены слегка поменять дизайн почтового ящика. Хотя IBM делать этого не рекомендует, так как при следующем обновлении шаблона почтового ящика изменения могут быть потеряны, но при соблюдении определенных правил все возможно.

Войдем в Domino Designer и создадим в шаблоне почтового ящика или непосредственно в нужной почтовой базе два агента. Один будет вызывать другой, и первый нам нужен исключительно как интерфейс ко второму. Назовем первого агента Report spam (см. рис. 7). Его исходный текст на языке Lotus Script простейший:

Листинг 6. Агент Report spam

```
Sub Initialize
        Dim s As New NotesSession
        Dim agent As NotesAgent
        Set db = s.CurrentDatabase
        Set agent = db.GetAgent( "doReportSpam" )
        If agent.RunOnServer = 0 Then
                Messagebox "Spam routed to SpamAssassin for learning",, "Success"
        Else
                Messagebox "Agent did not run",, "Failure"
        End If
End Sub
```

Этот агент нужен только для того, чтобы второй агент запустился на сервере, а не на клиенте.

Второй агент немного посложнее. Он называется doReportSpam и его исходный текст приведен в приложении 5. Этот агент для каждого файла из папки Junk Mail вызывает на сервере скрипт, обозначенный на рисунке 3 как Script 2. Полнофункциональный пример исходного текста этого скрипта на языке Bourne Shell представлен в приложении 6. Этот скрипт учитывает наличие двух серверов Domino, объединенных в кластер. В приложении приведен скрипт, предназначенный для сервера с DNS именем main. В копии этого скрипта на втором сервере кластера необходимо изменить настройки (константы member и mountpoint). Приведенный пример рассчитан на определенные внутренние соглашения об именовании каталогов и размещении каталога junkmail. В каждом конкретном случае необходима проверка и корректировка этого скрипта.

После сохранения проведенных изменений в дизайне баз для обучения байесовского фильтра SA можно использовать появившийся новый пункт меню, показанный на рисунке 8.

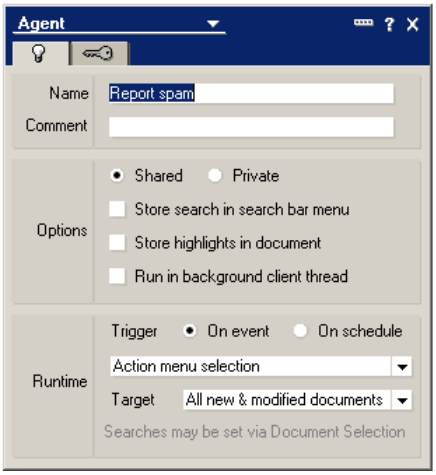

Рис. 8. Свойства агента Report spam

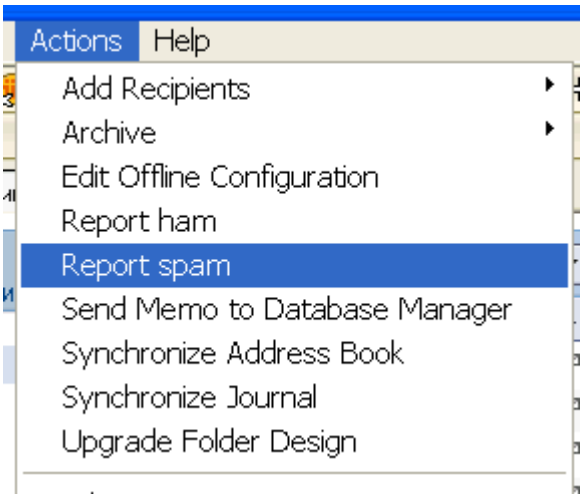

Рис. 7. Меню вызова агента для подстройки байесовского фильтра SA

Заключение

Предлагаемый подход реализован и проверен на практике. Он неплохо работает на низкопроизводительном оборудовании со средне-нагруженным сервером (несколько тысяч входящих писем в сутки).

Неоспоримые достоинства такого решения — нулевая стоимость дополнительного программного обеспечения (почтовый сервер мы рассматриваем как данность), экономия входящего трафика и возможность расширения.

Среди недостатков можно отметить относительную медлительность в обработке почты, которая обуславливается запуском целой связки приложений и shell-скрипта в сеансе SMTP. Решение реализовано только для протокола IPv4.

2009-08-23

AVN <avn@avnsite.com>

Приложение 1. Файл различий common/smtppass.c

```
Листинг доступен в полной версии статьи
```

Приложение 2. Файл различий common/spio.c

```
Листинг доступен в полной версии статьи.
```

Приложение 3. Файл /etc/init.d/proxsmtpd

```
#!/bin/sh
### BEGIN INIT INFO ######################################################
# Provides:          proxsmtpd
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Short-Description: smtp proxy filtering daemon
### END INIT INFO
# Note on runlevels:
# 0 - halt/poweroff                  6 - reboot
# 1 - single user                    2 - multiuser without network exported
# 3 - multiuser w/ network (text mode)  5 - multiuser w/ network and X11 (xdm)
#
#########################################################################
# CONFIGURATION
# Most configuration options are found in the proxsmtpd.conf file.
# For more info see:
#   man proxsmtpd.conf
# The prefix proxsmtpd was installed to
prefix=/usr/local/
PROXSMTPD_BIN=$prefix/sbin/proxsmtpd
INTERFACE=eth1
ADDFLAGS="-d 4"
# The location for pid file
piddir=/var/run/proxsmtp
#########################################################################
# SCRIPT
if [ ! -d $piddir ] ; then
  mkdir $piddir
fi
test -x $PROXSMTPD_BIN || { echo "$PROXSMTPD_BIN not installed";
       if [ "$1" = "stop" ]; then exit 0;
       else exit 5; fi; }
. /etc/rc.status
rc_reset

case $1 in
start)
        echo -n "Starting proxsmtpd... "
        if [ -f /var/log/proxsmtpd.log ]; then
          /bin/mv /var/log/proxsmtpd.log /var/log/proxsmtpd.log.`date +%F-%T`
```

```
        fi
        /bin/mkdir -p $piddir
        $PROXSMTPD_BIN $ADDFLAGS -p $piddir/proxsmtpd.pid 2> /var/log/proxsmtp.log &
        rc_status -v
        ;;
stop)
        echo -n "Stopping proxsmtpd... "
        [ -f $piddir/proxsmtpd.pid ] && kill `cat $piddir/proxsmtpd.pid`
        /bin/rm -f $piddir/proxsmtpd.pid
        rc_status -v
        ;;
restart)
        echo -n "Stopping proxsmtpd... "
        [ -f $piddir/proxsmtpd.pid ] && kill `cat $piddir/proxsmtpd.pid`
        /bin/rm -f $piddir/proxsmtpd.pid
        rc_status -v
        echo -n "Starting proxsmtpd... "
        if [ -f /var/log/proxsmtpd.log ]; then
          /bin/mv /var/log/proxsmtpd.log /var/log/proxsmtpd.log.`date +%F-%T`
        fi
        /bin/mkdir -p $piddir
        $PROXSMTPD_BIN $ADDFLAGS -p $piddir/proxsmtpd.pid 2> /var/log/proxsmtp.log &
        rc_status -v
        ;;
suspend)
        echo -n "Temporary suspending proxsmtpd... "
         /usr/sbin/iptables -t nat -D PREROUTING -i $INTERFACE -p tcp --dport 25 -j REDIRECT --to-
port 10025 2> /dev/null
        rc_status -v
        ;;
continue)
        echo -n "Rerouting packets to proxsmtpd... "
         /usr/sbin/iptables -t nat -A PREROUTING -i $INTERFACE -p tcp --dport 25 -j REDIRECT --to-
port 10025 2> /dev/null
        rc_status -v
        ;;
*)
        echo "usage: proxsmptd {start|stop|restart|suspend|continue}" >&2
        ;;
esac
```

Приложение 4. Файл spamassassin.sh

```
#!/bin/bash

##############################################################################
# PROXSMTP SCRIPT
#
# proxsmtp script which sends e-mail through SpamAssassin.
#
# See proxsmtpd.conf(5) for configuration details
#

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#   WARNING WARNING WARNING WARNING WARNING WARNING WARNING
#
#  By using variables passed in from smtp server in file
#  manipulation commands without escaping their contents
#  you are opening yourself up to REMOTE COMPROMISE. You
#  have been warned. Do NOT do the following unless you
#  want to be screwed big time:
#
#  mv $EMAIL "$SENDER.eml"
#
#  An attacker can use the above command to compromise your
#  computer. The only variable that is guaranteed safe in
#  this regard is $EMAIL.
#
#  The following script does not escape its variables
#  because it only uses them in safe ways.
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#
#  Enhanced variant
#
#  Author: Vladimir Anufriev <avn@avnsite.com>
#  Copyright (C) by AVN, 2009. All rights reserved.
#
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# Path for original e-mail body saving
# Must be open for r/w for smtp server owner
JUNKMAILDIR=/opt/lotus/junkmail
# Drop e-mail if SA returns spam level >= HARDSPAMLEVEL
HARDSPAMLEVEL=10
```

```
############################################################

if [ -z $EMAIL ] ; then
  echo "This script must be started by proxsmtpd daemon"
  exit 0
fi

shortname=`basename $EMAIL`
if [ $EMAIL != $JUNKMAILDIR/$shortname ] ; then
  /bin/cp $EMAIL $JUNKMAILDIR/$shortname
fi
/bin/chown notes.notes $JUNKMAILDIR/$shortname
#
tmpfile=`mktemp -p $TMPDIR`
echo "X-FileName: $JUNKMAILDIR/$shortname" > $tmpfile
/usr/bin/spamc -E < $EMAIL >> $tmpfile 2> /dev/null
result=$?
/bin/rm -f $EMAIL
/bin/mv $tmpfile $EMAIL

exitcode=1
logger -i -d -p mail.info "spamc: $EMAIL result=$result"

case "$result" in
0)
# Filter success
    while read line
    do
      if [[ $line =~ X-Spam-Level:\ (-[0-9]+) ]]
      then
        level=${BASH_REMATCH[1]}
        logger -i -d -p mail.info "$0: Message level=$level for $EMAIL from $SENDER"
        if [ $level -lt -50 ] ; then
          sa-learn --ham $JUNKMAILDIR/$shortname > /dev/null 2>&1
          rm -f $JUNKMAILDIR/$shortname
          logger -i -d -p mail.info "$0: Good message level=$level $EMAIL"
        fi
#        cp $EMAIL `mktemp -p $TMPDIR`
        break
      fi
    done < $EMAIL
    exitcode=0
    ;;
1)
    while read line
    do
```

```
      if [[ $line =~ X-Spam-Level:\ ([0-9]+) ]]
      then
        level=${BASH_REMATCH[1]}
        logger -i -d -p mail.info "$0: Message level=$level for $EMAIL from $SENDER"
        if [ $level -ge $HARDSPAMLEVEL ]
        then
           sa-learn --spam $JUNKMAILDIR/$shortname > /dev/null 2>&1
           rm -f $JUNKMAILDIR/$shortname
          logger -i -d -p mail.err "$0: Obvious spam message (level=$level) $EMAIL"
          # The last line of output to stderr will be used as an error message when the filter fails
           echo "550 Content Rejected: Suspected spam message" >&2
        else
          exitcode=0
       fi
        break
      fi
    done < $EMAIL
    ;;
64)
    logger -i -d -p mail.err "spamc: command line usage error"
    ;;
65)
    logger -i -d -p mail.err "spamc: data format error"
    ;;
66)
    logger -i -d -p mail.err "spamc: cannot open input"
    ;;
67)
    logger -i -d -p mail.err "spamc: addressee unknown"
    ;;
68)
    logger -i -d -p mail.err "spamc: host name unknown"
    ;;
69)
    logger -i -d -p mail.err "spamc: service unavailable"
    ;;
70)
    logger -i -d -p mail.err "spamc: internal software error"
    ;;
71)
    logger -i -d -p mail.err "spamc: system error (e.g., can't fork)"
    ;;
72)
    logger -i -d -p mail.err "spamc: critical OS file missing"
    ;;
73)
```

```
    logger -i -d -p mail.err "spamc: can't create (user) output file"
    ;;
74)
    logger -i -d -p mail.err "spamc: input/output error"
    ;;
75)
    logger -i -d -p mail.err "spamc: temp failure; user is invited to retry"
    ;;
76)
    logger -i -d -p mail.err "spamc: remote error in protocol"
    ;;
77)
    logger -i -d -p mail.err "spamc: permission denied"
    ;;
78)
    logger -i -d -p mail.err "spamc: configuration error"
    ;;
*)
    logger -i -d -p mail.err "spamc: unknown error: $result"
    ;;
esac

exit $exitcode
```

Приложение 5. Агент сообщения о спаме для почтового ящика Lotus Notes

```
Sub Initialize
'
' Learn bayes filter
'
' Author: Vladimir Anufriev <avn@avnsite.com>
' Copyright (C) by AVN, 2009. All rights reserved.
'
	Dim session As New NotesSession
	Dim db As NotesDatabase
	Dim view As NotesView
	Dim doc As NotesDocument
	Dim olddoc As NotesDocument
	Dim notesItem As NotesItem
	Dim s As String
	Dim sript As String
	Dim rc As Integer
	Dim flag As Boolean

	If( Not session.IsOnServer ) Then
		Print "This agent must run on server"
		Exit Sub
	End If

	Print "doReportSpam agent running..."
	Set db = session.CurrentDatabase
	If Not db.IsOpen Then
		Print db.FileName & " does not exist on " & db.Server
	End If
	script = "/usr/local/bin/reportspam.sh"
	Set view = db.GetView( "JunkMail" )
	Set doc = view.GetFirstDocument
	On Error Goto nextdoc
	While Not(doc Is Nothing)
		flag = False
		If( Not doc.IsDeleted ) Then
			Print "Processing document..."
			Set notesItem = doc.GetFirstItem( "X_FileName" )
			If Not( notesItem Is Nothing ) Then
				s = notesItem.Text
				If( s <> "" ) Then
					Print script & " " & s
					rc = Shell( script & " " & s )
					If( rc <> 33 ) Then
```

```
                                        Print "Shell start error: "& script & " " & s & " : " &
rc
                                Else
                                        flag = True
                                End If
                        End If
                End If
        End If
nextdoc:
        Set olddoc = doc
        Set doc = view.GetNextDocument(doc)
        If( flag ) Then
                olddoc.remove( False )
        End If
    Wend
End Sub
```

Приложение 6. Скрипт обучения фильтра

```
#!/bin/sh
#
# SpamAssassin learning script for Lotus Notes agent
# Author: Vladimir Anufriev <avn@avnsite.com>
# Copyright (C) by AVN, 2009. All rights reserved.
#
member=backup
mountpoint=/mnt/sata
memberdir=/lotus/junkmail

if [ -z $1 ] ; then
  echo "Usage: $0 message"
  exit 1
fi
if [ -f $1 ] ; then
  /usr/bin/sa-learn --spam $1
  /bin/rm -f $1
  exit 0
else # try to find file on backup
  mountflag=0
  shortname=`basename $1`
  filename=/mnt/$member$mountpoint$memberdir/$shortname
  if /bin/grep "/mnt/$member$mountpoint " < /proc/mounts > /dev/null 2>&1 ; then
    logger -i -d -p mail.info "$0: /mnt/$member$mountpoint already mounted"
  elif /bin/mount /mnt/$member$mountpoint ; then
    mountflag=1
  else
    logger -i -d -p mail.err "$0: /mnt/$member$mountpoint mount error"
    exit 1
  fi
  if [ -f $filename ] ; then
    /usr/bin/sa-learn --spam $filename
    /bin/rm -f $filename
    if [ $mountflag -eq 1 ] ; then
      /bin/umount /mnt/$member$mountpoint
    fi
    exit 0
  fi
  logger -i -d -p mail.err "$0: File $filename not found or of unknown mode"
  if [ $mountflag -eq 1 ] ; then
    /bin/umount /mnt/$member$mountpoint
  fi
fi
exit 0
```