Настройка Gnus для обработки спама

Alex Ott

8 января 2005 г.

Данная статья является небольшим введением в технологию борьбы со спамом средствами пакета Gnus. Данная статья основывается на возможностях пакета, которые появились начиная с ветки Oort Gnus, существуют в текущей стабильной версии 5.10.x и сейчас развивается в версии NoGnus.

1 Базовые понятия

Для правильного понимания того, как производить настройку и работать с Gnus, необходимо уточнить некоторые понятия:

пометка (mark) символьный атрибут, который устанавливается на статью;

функция разбиения (splitter) функция, которая для каждого письма должна вернуть имя группы в которую его необходимо перенести, или nil, если продолжать обработку дальше;

хук (hook) функция, выполняемая при возникновении какого-то условия и включении/выключении режима. Используется для предоставления пользователю возможности выполнять дополнительную настройку или другие действия.

2 Обработка спама

2.1 Основы настройки

Настройка анти-спама производится в несколько этапов, для выполнения которых необходимо в файл `~/.gnus` добавить несколько команд:

1. Сначала надо задать метод обработки. Gnus предоставляет несколько методов, которые могут быть использованы для определения спама — статистический метод, bogofilter, ifile, черные и белые списки адресов. Для статистического метода команда задания будет выглядеть так:

```
(setq spam-install-hooks t) ;; заставим установить все нужные хуки
(setq spam-use-stat t) ;; мы будем использовать статистический метод
```

2. Затем необходимо загрузить нужные модули Gnus:

```
(require 'spam-stat) ;; для использования spam-stat
(require 'spam)
```

3. Теперь надо выполнять проверку входящей почты на содержание спама. Обычно при обработке спама используется fancy-разбиение, которое позволяет использовать произвольный код ELisp внутри обработчика. Так, например, для nnmail разбиение будет выглядеть так (минимальная версия):

```
(setq nnmail-split-methods 'nnmail-split-fancy)
(setq nnmail-split-fancy
      '(|
          ;; перехватываем спам
          (: spam-split)
          ;; все остальное кладем в папку inbox
          "inbox"))
```

   Аналогичным образом может быть настроена и функция разбиения для метода доступа nnimap.

4. После этого, при обработке почты письма, определенные как спам, будут идти в специальную группу "spam", значение которой пользователь может переопределить.

5. Однако, особено в начале эксплуатации, спам все равно может проходить сквозь фильтры. Поэтому, пользователь может вручную помечать письма содержащие спам, и при выходе из группы для этих писем будет вызываться обработчик, специфический для конкретного метода обработки спама — spam-stat, bogofilter, ifile и другие. После этого, указанные спамерские письма будут учитываться при обработке следующих писем.

2.2 Схема обработки спама

Схема обработки почты Gnus'ом показана на рисунке XrefId[??].

Здесь можно выделить несколько этапов:

1. Новая почта получается из настроенных источников и передается на обработку в функцию разбиения. Функция разбиения, кроме прочих задач, настроена еще и на обнаружение спама (эта часть показана в виде выбора "Спам?");

2. В функции разбиения принимается решение о том к какой группе отнести обрабатываемое письмо. В зависимости от решения, письмо перемещается либо в группу для спама (красная, пунктирная черта), либо в группу для обычной не классифицированной почты. Эти операции производятся автоматически;

3. Затем, пользователь читает почту из группы с обычной почтой, и помечает спам (обычно это выполняется с помощью клавиши **M**–**d**);

4. После выхода из группы с обычными письмами происходит обработка оставшихся писем с помощью соответствующих процессоров. Письма помеченные как спам передаются на обработку процессору спама, и затем помечаются как expirable, или

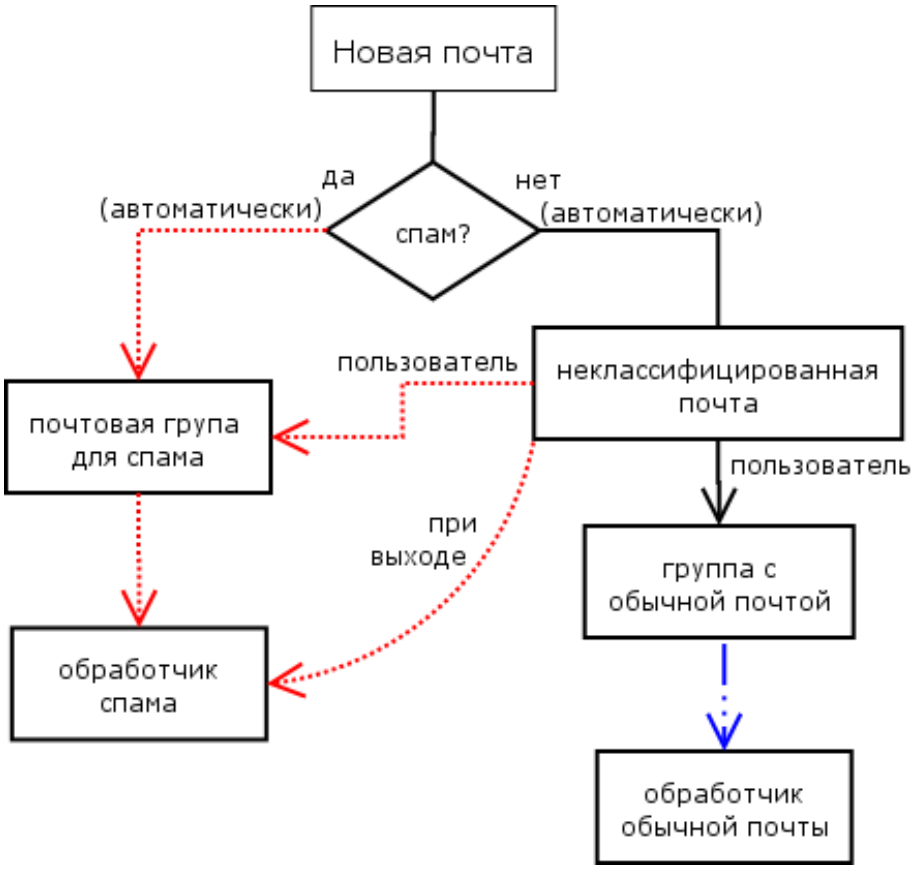

Рис. 1: Схема обработки почты

перемещаются в другую группу. А обычные письма передаются другому процессору, которые добавляет обычные письма в соответствующую базу.

2.3 Методы обработки спама

Gnus поддерживает разные методы и программы для определения писем со спамом. Вот список поддерживаемых методов:

- Черные и белые списки. Позволяют пользователю задавать списки адресов, которые будет считаться либо спамом (черный список), либо нормальным письмом (белый список). У данного метода есть несколько вариантов, как например белые списки из BBDB, когда все письма с адресов, которые имеются в BBDB рассматриваются как нормальная почта. Также есть модификация данного метода, когда в качестве признака нормальных писем используются токены Hashcash, а не почтовые адреса. Кроме этого, черные списки могут получаться из внешних источников;

- Соответствие заголовков регулярным выражениям. Данный метод позволяет пользователю задавать регулярные выражения, относительно которых будут проверяться заголовки писем, и по этим признакам письмо будет относится к нормальным письмам или письмам со спамом;

- Внешние программы — bogofilter, ifile и SpamOracle. В качестве фильтра спама используются соответствующие программы, которым для корректировки работы передаются письма вручную помеченные пользователем. Эти программы являются разными реализациями анти-спамовых систем на основе алгоритма Байеса;

- Spam-stat. Собственная реализация алгоритма Баейса на Emacs Lisp. Работает на любой платформе, на которой существует Emacs, но медленнее чем соответствующие аналоги, и требует работы больше оперативной памяти[1].

3 Настройка

Настройка обработчиков писем со спамом и нормальных писем (как и прочих параметров обработки спама) производится с помощью параметров из группы spam (используйте **M–x customize-group** `spam`). В этой группе настройки пользователь может вручную указать какой метод обработки использовать, куда складывать письма со спамом, а также другие параметры.

Настройку обработки спама можно разделить на две большие группы — использование возможностей пакета spam и использование комбинации стандартных возможностей Gnus.

3.1 Настройка конкретных методов пакета spam

Как уже упоминалось выше, настройка конкретных методов обработки спама производится установкой переменной, соответствующей нужному методу. Ниже перечислены имена соответствующих переменных, а также приведены дополнительные настройки для конкретных методов.

3.1.1 spam-stat

При использовании метода spam-stat настройка может выглядеть так:

```
;; устанавливаем параметры, метод обработки и загружаем модули
(setq spam-install-hooks t)
(setq spam-use-stat t)
(require 'spam-stat)
(require 'spam)
;; загружаем файл со статистикой по словам
(when (file-exists-p spam-stat-file)
  (spam-stat-load))
;; задаем имя группы для спама
(setq spam-stat-split-fancy-spam-group "spam")

;; А эта функция нужна в основном для первоначального обучения
;; байесовского фильтра.
(defun my-spam-stat-learn ()
  "Learn about my spam and non-spam"
  (interactive)
```

[1]По собственному опыту могу заметить, что этот метод очень эффективно отлавливает спам. За неделю сквозь фильтр проходит 1-2 писем со спамом, при общем потоке спама в 100-200 писем в неделю (того, который не ловится внешним SpamAssassin).

```
(let ((starting (current-time-string)))
  ;; обрабатываем спам
  (spam-stat-process-spam-directory "~/Mail/spam")
  ;; обрабатываем нормальные письма из разных каталогов
  (let ((ham-groups '("inbox" "Gnus-ding" "Emacs")))
    (mapc (lambda (x)
            (spam-stat-process-non-spam-directory
             (format "~/Mail/%s" x)))
          ham-groups))
  (spam-stat-reduce-size)
  (spam-stat-save)
  (message "my-spam-stat-learn:  started at %s, ended at %s"
           starting (current-time-string))))
```

Часто у пользователя уже есть письма со спамом. Для обучения по ним, и предназначена функция my-spam-stat-learn, которую можно вызвать интерактивно. Предполагается, что спам хранится в каталоге `~/Mail/spam`, а нормальные письма берутся из каталогов `~/Mail/inbox`, `~/Mail/Gnus-ding` и `~/Mail/Emacs` (вы можете подставить пути к своим каталогам, или удалить лишние записи в списке ("inboxGnus-dingEmacs")). Функцию `my-spam-stat-learn` достаточно запустить один раз, при этом сформируется база с весами слов, и затем она будет дополняться по результатам ручной обработки почты — через соответствующие обработчики при выходе из группы.

3.1.2 Черные и белые списки

Для использования черных или белых списков надо присвоить истинное значение одной из переменных:

`spam-use-blacklist` для использования черных списков, когда письма от отправителя, чей адрес находится в черном списке, будут считаться спамом и отправляться в соответствующую группу;

`spam-use-whitelist` для использования белых списков. В этом случае, письма от пользователей, адресов от которых нет в указанном списке, передаются следующему обработчику спама;

`spam-use-whitelist-exclusive` для неявного использования белых списков. При использовании данного метода, письма от отправителей не перечисленных в белом списке, будут рассматриваться как спам. Осторожно используйте данный метод.

Вы можете явно задать значения в файлах с белыми и черными списками, отредактировав файлы, на которые ссылаются переменные `spam-whitelist` и `spam-blacklist`.

3.1.3 Белые списки из BBDB

Данный метод аналогичен по работе использованию белых списков, но белые списки при этом берутся из базы BBDB[2]. Аналогично обычным белым спискам, для настройки используются две переменных: `spam-use-BBDB` и `spam-use-BBDB-exclusive`, которые заставляют Gnus работать также, как и при использовании `spam-use-whitelist` и `spam-use-whitelist-exclusive`.

3.1.4 Внешние источники черных списков (blackholes)

Gnus может проверять адреса в письмах относительно внешних, распределенных систем обработки спама. Для включения данного метода, вам необходимо присвоить истинное значение переменной spam-use-blackholes, например так:

```
(setq spam-use-blackholes t)
```

Кроме этого, пользователь может указать список серверов, относительно которых будет производиться проверка (с помощью переменной `spam-blackhole-servers`)[3].

3.1.5 Соответствие заголовков регулярным выражениям

Данный метод позволяет пользователю использовать регулярные выражения для проверки заголовков писем. Пользователь может использовать регулярные выражения как для определения писем со спамом, так и для определения нормальных писем.

Для использования данного метода нужно установить в истинное значение переменную spam-use-regex-headers. Переменные spam-regex-headers-spam и spam-regex-headers-ham должны содержать в себе списки регулярных выражений для спама и обычной почты[4].

3.1.6 Обработка спама с помощью bogofilter

Для данной программы возможно использование двух взаимоисключающих вариантов подключения. Первый вариант, который управляется переменной `spam-use-bogofilter`, запускает bogofilter для обрабатываемого письма и использует накопленную базу статистики. Второй метод, который включается переменной `spam-use-bogofilter-headers`, при фильтрации использует уже установленный заголовок `X-Bogosity` (например, он может быть установлен во время обработки письма с помощью procmail). Для использования одного из методов, необходимо установить соответствующую переменную в истинное значение.

Кроме основных переменных, включающих данный метод, пользователь имеет возможно задать расположение баз bogofilter. Оно задается с помощью переменной `spam-bogofilter-database-directory`.

[2] http://bbdb.sf.net

[3] Для данного метода не предусмотрено обработчиков спама и нормальной почты.

[4] Для данного метода не предусмотрено обработчиков спама и нормальной почты.

3.1.7 Обработка спама с помощью ifile

Использование ifile выполняется путем присвоения истинного значения переменной `spam-use-ifile`. Кроме того, как и в случае с bogofilter, с помощью переменной `spam-ifile-database-path` пользователь может задать путь к базам ifile.

3.1.8 Обработка спама с помощью SpamOracle

Аналогично другим методам, использующим внешние программы, настройка SpamOracle производится установкой переменной с именем `spam-use-spamoracle`. Пользователь также может задать путь к базе SpamOracle с помощью переменной `spam-spamoracle-database`.

3.1.9 Дополнительные методы

Кроме перечисленных методов Gnus позволяет пользователю достаточно просто добавить свои обработчики спама. Для того, чтобы узнать как это делать, прочитайте раздел "Extending the Spam ELisp package"в руководстве Gnus.

3.2 Другие методы обработки

Кроме использования пакета spam, можно фильтровать спам и с помощью других средств. Например, так выполняется фильтрация спама при помощи SpamAssassin.

3.2.1 Обработка спама с помощью SpamAssassin

Данный метод напрямую не использует пакет spam, а использует стандартные возможности Gnus, такие как выполнение команд при заборе почты, и стандартные средства разбиения почты по заголовкам.

Сначала необходимо внести изменения в настройку источников почты. Нужно добавить обработчики `:prescript` и `:postscript`, используя команды, аналогичные приведенным в примере:

```
(setq mail-sources
      '((file :prescript "formail -bs spamassassin < /var/mail/%u")
        (pop :user "testuser"
             :server "myhost"
             :postscript
             "mv %t /tmp/testuserfile; formail -bs spamc < /tmp/testuserfile > %t")))
```

Это приведет к тому, что у пользователя окажется почта, в которой вставлены дополнительные заголовки SpamAssassin.

Затем пользователь может использовать стандартные средства разбиения почты, проверяя наличие заголовка `X-Spam-Flag`, как это показано в примере:

```
(setq nnmail-split-methods '(("spam"  "^X-Spam-Flag: YES")
                             ...))
```

4 Благодарности

Хочется выразить благодарность всей группе разработчиков Gnus за многолетнюю работу по созданию такого замечательного продукта. А также подписчикам списка рассылки Gnus, где в обсуждениях рождаются новые идеи.

Отдельное спасибо моей жене, за терпение и понимание!