Greylisting - пора познакомиться с технологией «серых списков»

Опубликовано muff в Чт, 2009-09-24 13:51

Уже устал рассказывать пользователям и знакомым что такое Greylisting и "с чем его едят". А также отвечать на кучу вопросов "зачем", "как" и "почему". Поэтому и появилась эта статья.

Немного теории

Суть работы грейлистинга основана на предположении, что спамеры, осуществляя рассылку, далеко не всегда выполняют предусмотренные протоколом SMTP требования. В частности, этот протокол требует, чтобы при получении ответа с кодом 4xx, означающим временную проблему на сервере-получателе, сообщение помещалось в очередь отправителя, и спустя некоторое время предпринимались повторные попытки выполнить доставку.

Очевидно, что для спамеров выполнение этого требования обойдётся слишком дорого в плане затрачиваемых ресурсов потребуется вести очередь сообщений, а учитывая, что в базах адресов, как правило, многие получатели уже не существуют, а то и вовсе не существовали, дисковое пространство для такой очереди и ресурсные затраты на её обработку будут неоправданно высоки. Конечно, в случае «лобовой атаки», когда все адреса указываются в одном письме, такой проблемы не возникнет. Но поскольку подобная рассылка – слишком лёгкая добыча для различных фильтров, то спамеры всё чаще вносят в рассылаемые сообщения случайные фрагменты. А это уже будет означать отдельное сообщение на каждого получателя. Безусловно, можно модифицировать программу рассылки, чтобы она не вела очередь, а просто помечала проблемные адреса и выполняла на них повторную отправку, но в любом случае для спамера это означает дополнительные трудозатраты.

Как следствие многие спамеры просто игнорируют любые ошибки, в том числе и временные, продолжая отсылать сообщения дальше по своей базе. То есть если на первую попытку соединения возвращать код 4xx, то спамер его проигнорирует и оставит ваш сервер в покое (либо наоборот, если окажется очень настойчивым, будет непрерывно отправлять сообщения, не утруждая себя различными паузами). В то время как добропорядочный сервер, «выругавшись про себя», терпеливо положит письмо в очередь и чуть позже попытается ещё раз его отправить. На этом и основана фильтрация по «серым» спискам – «правильные» серверы будут предпринимать повторную попытку доставки спустя некоторое время (обычно это 30 минут или 1 час), а «неправильные» либо сделают повтор сразу же, либо не сделают вообще.

Преимущества

Как показывает практика, значительную долю нежелательной почты действительно удаётся отсечь с помощью грейлистинга. При этом почтовый трафик может ощутимо снизиться, поскольку в отличие от различных статистических и сигнатурных анализаторов предварительный приём спамерского сообщения не осуществляется.

В то же время грейлистинг практически полностью исключает ложные срабатывания, когда добропорядочное письмо блокируется фильтром и не попадает к адресату. Исключение может составить разве что случай, когда почтовый сервер отправителя по тем или иным причинам не вполне следует установленным стандартам. Но это, как говорится, не наши проблемы.

Также отмечу сравнительную нетребовательность к ресурсам вашего сервера. Конечно, всё не так просто, как в случае «чёрных» списков на основе DNS, но по сравнению с тем же SpamAssassin грейлистинг может считаться очень простым и незатратным методом, поскольку требует анализа только конверта сообщения и ведения несложной базы, в которой

будут фиксироваться состояния «триплетов» (IP-адрес, почтовый адрес отправителя, почтовый адрес получателя) – занесён ли этот триплет в «белый» список, была ли в недалёком прошлом попытка доставить такое же сообщение и т. д.

Ну и ещё можно отметить «юридическую» чистоту этого метода. Если блокирование почты на основании DNSBL может вступить в противоречие с обязательством провайдера обеспечивать надёжную и бесперебойную работу обслуживаемой сети, а различные анализаторы при известной сноровке можно рассматривать как покушение на конституционное право пользователей на тайну их личной жизни (особенно если копии сообщений, признанных спамом, доставляются администратору или помещаются в общедоступный карантин), то грейлистинг не нарушает ни того ни другого, работая строго в соответствии с техническими требованиями. Поскольку любой сервер может при тех или иных обстоятельствах вернуть временную ошибку, то ничего криминального в этом нет.

Недостатки

С другой стороны, грейлистинг при всей своей идеальности обладает и рядом отрицательных моментов. Прежде всего эта технология относится к «невежливым», поскольку вынуждает сервер отправителя тратить свои ресурсы на дополнительное обслуживание искусственно создаваемой очереди. Только представьте себе, что произойдет с сервером типа mail.ru, если каждое исходящее сообщение он будет вынужден ставить в очередь и отсылать повторно?

Далее отправитель, получив временную ошибку, скорее всего попытается отправить сообщение на резервный сервер для вашего домена. Если Backup-сервер (т.е. хост с менее приоритетной MX-записью) у вас есть и он не включён в ваш «белый» список, то львиная доля нагрузки по обслуживанию очереди будет переложена на его плечи (в смысле на дисковую подсистему). И кстати говоря, ваш Backup-сервер наверняка будет скрупулёзно придерживаться протокола, так что любой спамер, додумавшийся использовать резервную MX-запись, может быть уверен, что его сообщение будет доставлено.

Если же резервный сервер занести в «белый» список, то нагрузка на него, конечно же, значительно снизится. Но вот сам он станет прекрасным способом гарантированно доставить вам любое сообщение (впрочем, он его в любом случае доставит, как было показано выше).

То есть, чтобы грейлистинг работал, все Backup-серверы тоже должны использовать эту технологию. А следовательно, нагрузка на отправителя возрастёт ещё больше, поскольку он будет не только держать сообщение в очереди, но и безнадёжно «ломиться» на несколько хостов, вместо того чтобы «успокоиться» после одной неудачной попытки.

Ещё одна проблема связана с серверами, на которых эксплуатируется «конкурирующая» система борьбы со спамом – так называемый обратный звонок (callback). Суть этого метода заключается в следующем: при получении входящего соединения сервер на стадии RCPT TO приостанавливает сессию и имитирует рабочую сессию с сервером, указанным в команде MAIL FROM. Если эта попытка из-за несуществующего адреса отправителя или по другим причинам завершается неудачей, то и приостановленное соединение разрывается без дальнейшей обработки.

Нетрудно догадаться, что если вы будете использовать грейлистинг, то у вас наверняка возникнут проблемы с отправкой почты на серверы, где настроен callback: в ответ на ваше подключение удалённый сервер попытается установить с вами «встречное» соединение, а вы его отправите «попробовать немного позже». Кому от этого будет хуже – неизвестно.

Ну и кто знает, как будут реагировать на ошибку 4xx службы «легальной» рассылки. Для них тоже не доставит удовольствия по полдня возиться с обработкой задержанных по тем или иным причинам сообщений.

Как это работает?

Технология Graylisting предельно проста. Для ее работы необходимы всего лишь три

составляющие SMTP-сессии, т.н. «триплет»:

1. IP-адрес хоста, пытающегося выполнить доставку сообщения.

2. Адрес отправителя сообщения, передаваемый в MIME-конверте

3. Адрес получателя, так же передаваемый в MIME-конверте

На основании данного триплета мы можем однозначно идентифицировать сообщение. И принцип работы Graylisting’a сводится к очень простому принципу:

Если мы впервые получаем сообщение, идентифицируемое данным триплетом, мы откладываем доставку в течение этой и нескольких последующих сессий, в течение заданного промежутка времени с передачей на сторону отправителя кода временной ошибки.

Т.к. протокол SMTP изначально был разработан, как ненадежный метод транспорта, вероятность временных сбоев заложена в саму его спецификацию (RFC 821), на чем и базируется применение данной технологии. Любой нормальный агент передачи сообщений (Message Transfer Agent, MTA) имеет в своей архитектуре заложенное требование повторять попытки доставки сообщения при получении определенных кодов временных ошибок.

При разработке данной технологии, происходившей в 2003 году, было проведено множество тестов, в ходе которых было обнаружено, что большинство приложений, используемых для рассылки спама, были именно под эту единственную цель и разработаны. Их работа основывалась на принципе хорошего снайпера: «Один выстрел – один труп!» Иными словами, производился пакетный выброс сообщений по адресам, которые брались списком из какой-либо спамерской базы, и задача считалась выполненной. Таким образом действует подавляющее большинство спам-ботов. Использование хотя бы одного отказа в приеме снижало долю спама на 95%. Конечно, спам – это целая индустрия, и технологии спама всегда на шаг опережают технологи борьбы с ним, поэтому вполне закономерно, что спамеры стали приспосабливаться к такому поведению своих жертв. Но Graylisting никогда и не позиционировался, как самостоятельное средство борьбы со спамом. Основная его цель – снизить потребление ресурсов принимающего сервера, затрачиваемое на обработку входящего потока различными контентными фильтрами, перенести боевые действия на сторону спамера, сделать спам более дорогостоящим занятием.

Почему именно так и происходит? Разберем пример обычной SMTP-сессии. Итак при попытке передачи сообщения при помощи MTA, происходит следующий обмен данными:

```
-> HELO somedomain.com

<- 250 Hello somedomain.com

-> MAIL FROM: <sender [at] somedomain [dot] com>



<- 250 2.1.0 Sender ok

-> RCPT TO: <recipient [at] otherdomain [dot] com>



<- 250 2.1.5 Recipient ok

-> DATA

<- 354 Enter mail

...
```

```
<- 250 2.0.0 Message accepted for delivery
```

Понятно, что на этапе установки SMTP-сессии можно провести массу проверок, как то проверка по DNS SBL\RBL, проверку Reverse DNS Lookup, проверка на блокировку отправителя\получателя по Custom Blacklist\Whitelist. Ну а если спамер новенький? Письмо пройдет, а зачастую письма могут быть очень серьезных объемов из-за обилия графической информации, различных вложений, а зачастую и вирусов, и лишь после полного прохождения в дело вступят интеллектуальные контентные фильтры. Что мы имеем? Мы имеем гигантский трафик, мы имеем значительную ресурсоемкость приложений контентной фильтрации, и даже можем получить остановку почтового потока. Что в данной ситуации делает Graylisting?

Если сообщение от данного пользователя данному получателю приходит впервые – SMTP-сессия до обидного коротка:

```
-> MAIL FROM: sender [at] somedomain [dot] com

<- 250 2.1.0 Sender ok

-> RCPT TO: <recipient [at] otherdomain [dot] com>

<- 451 4.7.1 Please try again later
```

Т.к в течение столь короткой сессии получены все необходимые данные для формирования триплета, и, как следствие, идентификации сообщения. Что, естественно, приведет к минимизации почтового трафика, т.к. сессия обрывается в самом ее начале, минуя этап передачи собственно данных. Технология хранения триплетов тоже относительно проста. Неважно, будет ли это файл в локальной системе, либо какая-то база данных, возможно удаленная; для успешной работы Graylisting’а необходимо хранить следующие данные триплета:

1. Время, когда данный триплет был получен впервые (record create time)

2. Время прекращения блокировки данного триплета (block expiried time)

3. Время жизни записи (record age time)

4. Время последнего вхождения данного триплета (last entry time)

Дополнительно может храниться информация об общем количестве отложенных сессий и о количестве сессий, прошедших успешную передачу.

Если в течение времени жизни записи сообщений с данным триплетом получено не будет, запись будет удалена из хранилища, и последующее письмо с данным триплетом будет проходить полный цикл обработки.

В технологии Graylisting’а, одним из очень частых методов является ручное составление белых списков. Это часто используется для ускорения доставки писем от важных отправителей, от доверенных доменов, от надежных ретрансляторов.

Технология белых списков основана на абсолютном доверии доменному суффиксу, либо IP-адресу сервера, выполняющего отправку, либо конкретному отправителю. Этот метод тоже имеет свои недостатки, например при помощи подмены адреса, спамер может теоретически обойти Graylisting используя белые списки. Но на это есть другие проверки, о чем речь не в этой статье.

Итак, как же будет выглядеть полная картина проверки почтовых сообщений в системе,

использующей технологию Graylisting:

1. Проверка по белому листу IP-адреса сервера-отправителя, если сервер в белом списке – письмо принимается.

2. Проверка адреса отправителя по белому листу – если адрес в нем присутствует – письмо принимается.

3. Проверка адреса получателя по белому списку – если он там имеется, письмо принимается. (А что, есть общительные люди, горящие желанием принимать абсолютно всю почту, и с наслаждением в ней копаться. Чужие слабости надо уважать)

4. Проверка триплета на вхождение:

4.1. Если это первое вхождение, создается запись в базу\файл данных со штампом RCT (record create time), запускается счетчик BET (block expiried time) и отправляющему серверу отправляется код ошибки 451 4.7.1.

4.2. Если это не первое вхождение триплета, но счетчик BET еще не обнулился, отправляющему серверу уходит код ошибки 451 4.7.1., в базе создается запись со штампом last entry time

4.3. Если это не первое вхождение, счетчик BET равен нулю, письмо принимается сервером. Запись last entry time обновляется

5. При успешном прохождении добавляется единица к счетчику успешных попыток передачи, и время жизни записи сбрасывается на исходное значение.

6. Если поле MAIL FROM имеет значение null, что классифицируется как message with blank sender, после команды RCPT TO временная ошибка не отправляется, но она отправляется после команды DATA.

В статье испозовались материалы:

Сергей Супрунов. Greylisting: панацея от спама или «мыльный пузырь»? [1]

Олег Крылов. Graylisting – как метод борьбы со спамом. Общая теория. [2]

Ссылки:
[1] http://av5.com/journals-magazines-online/1/15/116
[2] http://okrylov.wordpress.com/2009/07/10/graylisting-spamfighter-method-general-theory/