SpamAssassin の紹介

日本 SpamAssassin ユーザ会 滝澤 隆史

はじめに

前回のオープンソースカンファレンス(OSC2006 Tokyo/Fall)に日本 SpamAssassin ユーザー会として初めて出展しました。そこでは、「SpamAssassin って何ですか？」「スパムメールを削除してくるのですか？」「Windows で使えますか？」などの質問を頂きました。そこで、SpamAssassin とは何かを紹介する資料が必要だと考え、この資料を作りました。

SpamAssassin の紹介

SpamAssassin とは

SpamAssassin は様々なテストを行ってスパムらしさを判定するメールフィルタです。

スパムらしさを判定するテストにはテキスト分析、ベイズフィルタ、DNS ブラックリスト、協調型データベースなどを使います。

様々なテストの結果をスパムらしさのスコアとして加算していくため、正常なメールをスパムと誤判定することがが少なくなります。

さらに、プラグインで様々なテストを追加することができます。例えば、画像スパムに対しては OCR 技術を使ったテストを行うものが外部のプラグインとして開発されています。

なお、SpamAssassin は Apache Software Foundation の基に開発が行われています。

出来ることと出来ないこと

よく勘違いされますが、SpamAssassin 単体ではスパムを除去したり、振り分けした、バウンスメールを送ったりはできません。SpamAssassin が行うのはスパムらしさのスコアを付け、判定して、ヘッダに情報を追加したり、書き換えたりすることくらいです。

スパムの除去や振り分けなどを行いたい場合は、サーバとしては amavisd-new など、クライアントとしては procmail や maildrop などの他のソフトウェアと組み合わせて使います。

動作環境

Perl 5.6.1 以降が動く環境では SpamAssassin も動くはずです。そのため、最近のほとんどの Linux ディストリビューションや*BSD を含めて UNIX 互換 OS では動くはずです。

Windows 環境での動作については別の配布資料の「ウィンドウズ環境における SpamAssassin」をご覧ください。

なお、後述する日本語対応パッチは Perl 5.8 以降が必要です。

SpamAssassin に対する評価

Linux New Media Awards 2006

Linux Magazine を発行している LINUX NEW MEDIA が主催した Linux New Media Awards 2006 の Best Linux-based Anti-spam Solution 部門で SpamAssassin は２位以下を大きく引き離して１位を獲得しました。

Datamation Product of the Year 2006 Awards

米国 IT 誌 Datamation が主催する Product of the Year 2006 Awards のアンチスパムカテゴリにおいて、受賞候補として商用製品があがっている中でオープンソースソフトウェアの SpamAssassin が受賞しました。

日本語対応

SpamAssassin は残念ながら日本語には対応していません。しかし、日本 SpamAssassin ユーザ会の有志により開発された日本語対応パッチをあてると日本語対応になります。

この文書の執筆時点では日本語対応パッチは筆者のサイトで配布しています。

http://www.emaillab.org/spamassassin/

このパッチを当てると、日本語のルールを書くことができ、また、ベイズフィルタの学習精度も非常に良くなります。そのため、日本語のスパムの検出精度が非常に上がります。

将来は日本語対応パッチを本家の SpamAssassin にマージしてもらうように活動する予定です。

日本 SpamAssassin ユーザ会

日本における SpamAssassin に関する情報交換や日本語対応のルールの作成や日本語対応パッチの開発等を目的として 2006 年 3 月に発足して活動をしています。

日本 SpamAssassin ユーザ会のサイトは次の通りです。

http://spamassassin.jp/

SpamAssassin の概要

SpamAssassin が提供するもの

SpamAssassin は次のものを提供しています。

- スパムらしさを判定する Perl のライブラリ
- ツール
- 標準のプラグイン
- 標準のルールファイル

スパムらしさを判定する Perl のライブラリ

SpamAssassin をインストールすると Mail::SpamAssassin モジュールが Perl のライブラリとしてインストールされます。Perl のスクリプトからこのモジュールを呼び出して使うことによりスパム判定などを行うことが出来ます。

使い方は簡単です。Perl のスクリプトを書ける方であれば簡単に使えるでしょう。

```
use Mail::SpamAssassin;
my $sa = Mail::SpamAssassin->new();
my $mail = $sa->parse($message);
my $status = $sa->check($mail);
if ($status->is_spam()) {
    $message = $status->rewrite_mail();
    ....
}
$status->finish();
$mail->finish();
```

SpamAssassin のツール

次のツールが提供されます。

spamassassin	メールがスパムであるかどうかを判定するツール。
spamc	メールがスパムであるかどうかを判定するツール。spamd のクライアントとして動く。
spamd	メールがスパムであるかどうかを判定するデーモン。spamc をクライアントとして接続を受け付ける。
sa-learn	ベイズの学習を行わせるツール。
sa-update	最新のルールファイルをダウンロードしてきて更新するツール。

標準のプラグイン

標準のプラグインには次のものがあります。

- 自動学習関連（AutoLearnThreshold、AWL）
- パターンテスト関連
  （WhitelistSubject、MIMEHeader、ReplaceTags）
- 国、言語関連（RelayCountry、TextCat）
- ネットワークテスト関連
  （DCC、Pyzor、Razor2、SpamCop、URIDNSBL）
- 送信者認証関連（SPF、DomainKeys、DKIM、HashCash）
- その他（AccessDB、AntiVirus）

標準のルールファイル

標準で様々なルールが記述されたファイルがインストールされます。この標準のルールファイルを使うだけでもある程度のスパムの検出が出来ます。

テストの種類

スパムらしさを判定するためにプラグインで行うことも含めて次のような様々な種類のテストを行っています。

- パターンテスト
  - ヘッダ
  - ボディのテキストパート
  - URI
  - メッセージ全体
  - ホワイトリスト・ブラックリスト
- 国、言語のテスト
  - メールが中継された国の一覧
  - テキストから言語の判断
- ネットワークテスト
  - IP アドレスやホスト名
  - DNS ブラックリスト
  - URIDNS ブラックリスト
  - 協調型データベース
  - 送信者認証（SPF, DomainKeys, DKIM）
- ベイズフィルタのテスト
- META テスト

判定後の処理

メールがスパムであるかどうかを判定した結果として次の処理を行います。

- メールの書き換え
  - ヘッダの書き換え
  - スコアや判定結果のヘッダの追加
  - スパムペールのカプセル化（message/rfc822 形式）
- 協調型データベース等への報告
- ベイズフィルタの学習

SpamAssassin の使い方

インストール

SpamAssassin はパッケージシステム等でインストールすることもできますが、日本語対応パッチをあてる必要があるので、手動でインストールを行います。

次のサイトから SpamAssassin の tar ball をダウンロードします。

http://spamassassin.apache.org/

次のサイトから日本語対応パッチと説明文書をダウンロードします。

http://www.emaillab.org/spamassassin/

まず、必要な SpamAssassin の tar ball を展開してできるファイル INSTALL の"Required Perl Modules"と"Optional Modules"に記述されている Perl モジュールを必要に応じて予めインストールしてください。Perl モジュールが入っていないと利用できない機能やプラグインがあります。

次に日本語対応パッチの説明文書を読んでインストール作業を行ってください。

設定ファイル・ルールファイルの記述方法とプラグインの利用方法

OSC2006 Tokyo/Fall で配布した「SpamAssassin のプラグイン紹介」を読んでください。以下のサイトから PDF ファイルがダウンロードできます。

http://www.emaillab.org/spamassassin/

設定ファイル等を変更した場合は、必ず次のコマンドを実行してください。

```
$ spamassassin --lint
```

何も出力がなければ問題はないです。設定ファイルの記述に間違いがある場合はエラーが出ます。

ツールの紹介

SpamAssassin が標準で提供しているツールの紹介を行います。

spamassassin

spamassassin コマンドは Mail::SpamAssassin モジュールのフロントエンドとなる Perl のスクリプトです。このコマンドにメールを渡してスパムらしさの判断を行わせることができます。

SpamAssassin の tar ball にはスパムではないメールのサンプル sample-nonspam.txt とスパムメールのサンプル sample-spam.txt が含まれていますので、これを利用して試してみましょう。

spamassassin コマンドを使って次のように実行してください。

```
$ spamassassin < sample-spam.txt | less
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.1.8 (2007-02-13) on a.example.org
X-Spam-Level: **********************************************
X-Spam-Status: Yes, score=1002.5 required=5.0 tests=GTUBE,NO_RECEIVED,
        NO_RELAYS,RAZOR2_CF_RANGE_51_100,RAZOR2_CF_RANGE_E4_51_100,
        RAZOR2_CHECK autolearn=no version=3.1.8
X-Spam-Report:
        * -0.0 NO_RELAYS Informational: message was not relayed via SMTP
        * 1000 GTUBE BODY: Generic Test for Unsolicited Bulk Email
        *  1.5 RAZOR2_CF_RANGE_E4_51_100 Razor2 gives engine 4 level
        *      confidence above 50%
        *      [cf: 100]
        *  0.5 RAZOR2_CHECK Listed in Razor2 (http://razor.sf.net/)
        *  0.5 RAZOR2_CF_RANGE_51_100 Razor2 gives confidence level
        *      above 50%
        *      [cf: 100]
        * -0.0 NO_RECEIVED Informational: message has no Received headers
```

メールのスパムらしさの判定が行われ、X-Spam-で始まるヘッダが挿入され、判定結果が表示されます。それぞれのヘッダの意味は次の通りです。

X-Spam-Flag	スパムとして判定されたらこのヘッダが挿入されます。
X-Spam-Checker-Version	SpamAssassin のバージョン表示が記述されます。
X-Spam-Level	スコアの分だけが*が追加されます。
X-Spam-Status	判定結果の要約が記述されます。
X-Spam-Report	スパムとして判定されたら、このヘッダが挿入され、判定理由の詳細が記述されます。

spamd と spamc

メールを 1 通ごとに spamassassin コマンドでチェックしていたら毎回初期化のオーバーヘッドがかかります。そのため、たくさんのメールを効率よくチェックするためにデーモンとして動く spamd と C で書かれた軽量なクライアントの spamc というものが用意されています。

使い方としては、spamd をデーモンとして常駐させておき、spamc コマンドに標準入力からメールを与えます。spamc は UNIX ドメインソケットあるいは TCP/IP ネットワーク経由で spamd に接続して、メールのチェックを行わせます。

なお、spamd の rc スクリプトはソースコードの tar ball を展開したディレクトリの spamd ディレクトリにあるので、プラットフォームに応じて利用してください。

ベイズフィルタの学習 sa-learn

ベイズフィルタは spam（スパムメール）と ham（スパムではないメール）を標準ではそれぞれ 200 通以上学習しないとベイズフィルタのテストを行いません。ベイズフィルタの自動学習機能があるので、それぞれ学習するのを待つということも出来ますが、すでにスパムメールをたくさん持っているのであれば、sa-learn コマンドで強制的に学習させることができます。

次の例では Maildir 形式で溜め込んだスパムメールを学習させています。メールがたくさんある場合は--progress オプションを付けた方が進捗が見ることが出来て良いでしょう。

```
$ sa-learn --spam --progress ./
  1% [=                          ]  3.23 msgs/sec 09m34s LEFT
```

これと同じようなことをスパムではないメールについても--ham オプションを付けて学習させてください。

なお、ベイズのデータベースはユーザー毎に作成されることに注意してください。特にメールサーバで一括して学習させている場合は SpamAssassin を呼び出しているプログラムを実行しているユーザのデータベースに対して学習を行わせる必要があることに注意してください。

ルールファイルのアップデイト sa-update

私たちがスパム対策を行うと、スパムを送る側もその対策をすり抜けるようにスパムメールに工夫を行います。そのため、ルールファイルのメンテナンスを行わないと最新のスパムメールに対応できなくなります。

そのため、SpamAssassin はルールファイルを更新する sa-update コマンドを用意しています。ウィルスやワームのように時間単位で新しいものが出てくるのとは違うので、１週間に１回程度、短くても１日に１回、実行すればよいと思います。

使い方（クライアント編）

UNIX 系 OS のクライアント側で SpamAssassin の利用方法として一番やりやすいのは、fetchmail と procmail や maildrop との組み合わせで用いる方法でしょう。

ここでは fetchmail と maildrop を SpamAssassin と組み合わせて使用した例を紹介します。

SpamAssassin の準備

まず、先に紹介した資料を基にして SpamAssassin の設定を行ってください。設定が終わったら spamd をデーモンとして起動してください。

次に、サンプルのメールを spamc に与えて正常に機能しているか確認してください。

```
$ spamc < sample-spam.txt |less
```

maildrop の準備

maildrop は procmail ほどは知られていませんが、Maildir 形式のメールボックスをサポートしたメールの振り分けができるメール配送エージェント（MDA）です。procmail とは異なり、AWK や Perl の文法に似ているフィルタリング言語を使うため、スクリプトやプログラムを普段書いている人にとっては直感的にフィルタを書くことができます。

フィルタは.mailfilter というファイルに記述します。慣れないうちは万が一のメールの消失を防ぐために、次の記述例のように cc コマンドで全てのメールのバックアップを取るようにしたらよいでしょう。運用が安定したら cc 行を削除したらよいでしょう。

```
DEFAULT=$HOME/Maildir/

cc $HOME/Mail/backup/
if (/^X-Spam-Level: ¥*{15,}/)
        to $HOME/Mail/spam/
if (/^X-Spam-Level: ¥*{5,}/)
        to $HOME/Mail/spammy/
to $HOME/Maildir/
```

なお、この例では SpamAssassin により追加された X-Spam-Level ヘッダの*マークの数により、振り分けを行っています。確実にスパムと思われるスコアを 15 として、スコアが 15 以上のメールを spam フォルダに格納します。スパムではないメールも時々入ってしまい、確実にスパムとは言えないスコアを 5ー15 としてそのスコアのメールを spammy フォルダに格納します。こうすると、サルベージ作業は spammy フォルダだけを見ればよく作業が楽になります。この例で示したスコアの設定値は、運用当初はルールファイルのチューニングやベイズの学習状況により調整が必要です。筆者のお薦めとしてはそれぞれの値を最初は 20 と 10 に設定して、様子を見ながら徐々に下げていくことです。

筆者の経験上では 15 以上のスコアはほぼ確実にスパムで、SpamAssassin の設定パラメータの required_score のデフォルト値である 5 以上になるとほとんどスパムですがときどきスパムでないメールが混じる状態になります。

なお、振り分けするフォルダは予め maildirmake コマンドで作成する必要があります。

fetchmail の準備

fetchmail については説明不要だと思いますが、POP サーバや IMAP サーバからメールを取ってきて、他のサーバに転送したり、ローカルに配送したりするソフトウェアです。

fetchmail の設定ファイルにおいて、次の例のように mda として"spamc | maildrop"を指定するようにしてください。そうすると、POP サーバ等から取得したメールは spamc に渡され、スパムの判断結果のヘッダが追加されます。さらに、そのメールは maildrop に渡され、振り分けが行われます。

```
poll pop.example.org proto pop3
        port 995
        user foo
        pass secret
        ssl
        mda "spamc | maildrop"
        fetchall
        fetchsizelimit 0
```

実行

以上の準備が終わったら、fetchmail コマンドを実行してください。エラーがでなければ、それぞれのフォルダに振り分けが行われているはずです。

使い方（サーバ編）

amavisd-new と連携ができます、とだけ書いておきます。構築はちょっと大変ですが、興味のある方はがんばってください。

詳細な説明をしたいところですが、時間の関係上できませんでした。次回にでも機会があれば資料を作るかもしれません。

SpamAssassin の紹介

発行日　2007 年 3 月 17 日

著　者　滝澤 隆史