HBSTUDY #14
60分間スパムクッキング

株式会社サードウェア 滝澤隆史

私は誰？

- 氏名　滝澤隆史
- 所属　株式会社サードウェア
- twitter: ttkzw

- 仕事内容
  - ソフトウェアを開発したり
  - Linuxサーバを構築したり
  - ツッコミを入れたり

私は誰？

- オープンソース関連
  - http://www.emaillab.org/ の中の人
  - http://www.emaillab.jp/ の中の人
  - 昔はqmail使い（qmail-vidaの作者）
    - 今はPostfix + Dovecot使い
  - Muttの日本語／国際化対応関連
  - SpamAssassinの日本語対応パッチ
  - DNSキャッシュサーバUnboundの紹介

本日のレシピ

- 準備（15分）
  - 迷惑メール対策概論
  - SpamAssassin紹介編
- 本論（40分）
  - SpamAssassin概要編
  - SpamAssassin導入編
  - SpamAssassin実践編
- おまけ（5分）
  - 日本語ルール自動生成
  - 日本語対応ルール配布サイト

迷惑メール対策概論

迷惑メール対策（受信）

- 迷惑メール（スパム）と
  正常なメール（ハム）との区別
  - 人は文脈で判断できる
  - コンピュータは文脈では確実には判断できない

コンピュータによる判断

- 迷惑メールの特徴に基づいて行っている
  - 経路情報
  - エンベロープ情報
  - SMTPセッションの挙動
  - メールの内容

経路情報

- 送信元ホストのIPアドレスが逆引きできるか？
- 送信元ホストが動的IPアドレスであるか？
  - クライアントPCからの送信の可能性
  - botnetやワーム

経路情報

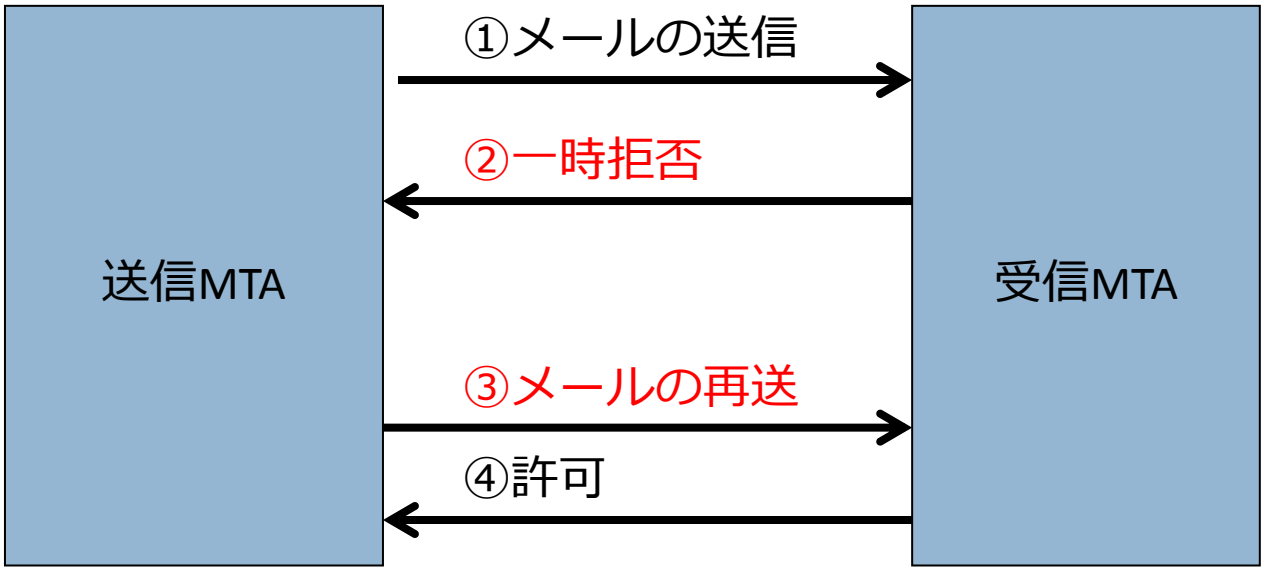
- DNSBL/DNSWL
  - 送信元ホストがブラックリスト／ホワイトリストに登録されているか？
  - リストの登録情報が正しく整備されているか？という問題がある
- →経路情報のみによる判断は「疑わしい」というレベルであり、誤判定の恐れがある

エンベロープ情報

- MAIL FROMのドメインパートのRR（MXレコードやAレコード）が存在するか？
  - バウンスメールを送ることができないから、拒否してよいんじゃないか？　という考えもある。
- HELO/EHLOのドメイン名のRRが存在するか？
  - クライアントPCは自身のコンピュータ名を名乗ることが多い
- →DNSのRRやMTAが正しく設定されていないことがあり、誤判定の恐れがある。

SMTPセッションの挙動

- Greylisting
  - 一時拒否した後に再送してきたら受け取る。
  - スパムは大量送信するために再送しない、という考えに基づいている

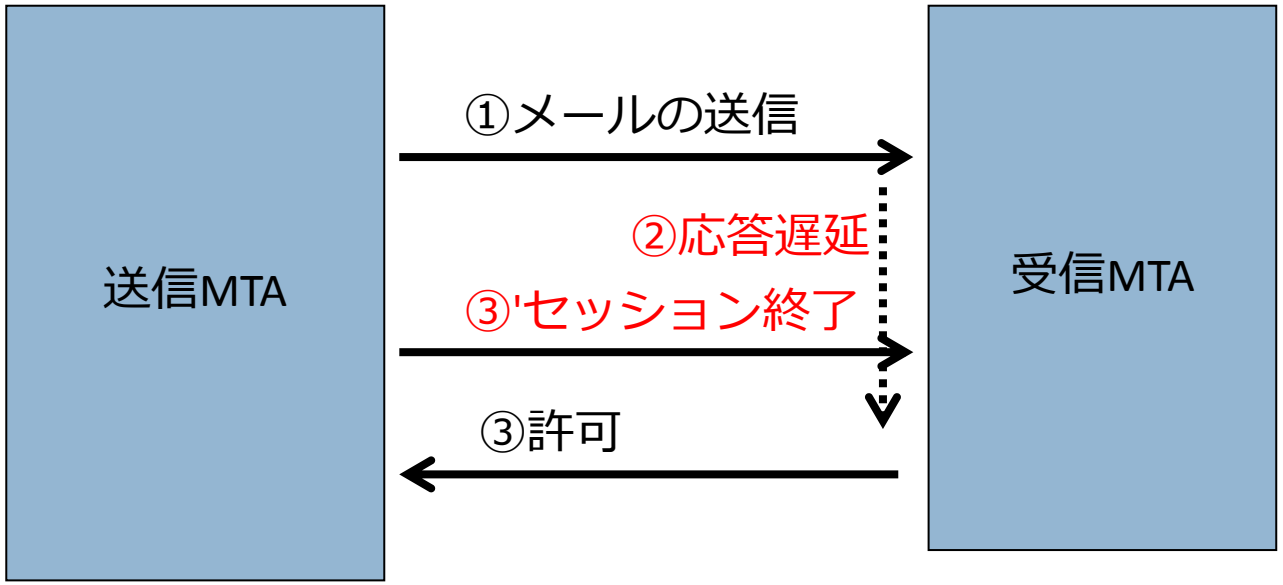


SMTPセッションの挙動

- Greylisting
  - →非常に効果はあるが、配送遅延の問題、再配送しないシステムの問題がある
  - →問題の影響を減らすために、Selective Greylistingを利用することもある
    - 送信ホストが疑わしいとき（S25R、SPF、DNSBL等の判断）にGreylistingを行う

SMTPセッションの挙動

- Tarpitting
  - SMTPセッション中の応答を遅らせる
  - スパムは大量送信するために、タイムアウトを短くして、自らセッションを切断する、という考えに基づいている



SMTPセッションの挙動

- Tarpitting
  - →送信側と受信側の双方に、セッション時間が長くなり、プロセス数が増加するという問題がある

メールの内容

- ルールベースフィルタ
  - 迷惑メールらしい特徴を持ったキーワードやパターンがメッセージに含まれているか
  - →ルールの整備が大変
  - →スパムの文面の巧妙化

メールの内容

- ベイジアンフィルタ
  - ベイズの定理を用いた統計確率的手法
  - 予め迷惑メールと正常なメールをそれぞれ学習させ、メールに含まれる単語の統計解析を行う。
  - 新しいメールが来たら、そのメールに含まれる単語を解析し、迷惑メールである確率を計算する。
  - この確率がある閾値（例えば、95%）を超えていれば迷惑メールであると判断する。
  - →確率の閾値を高くしないと正常なメールが引っかかる

メールの内容

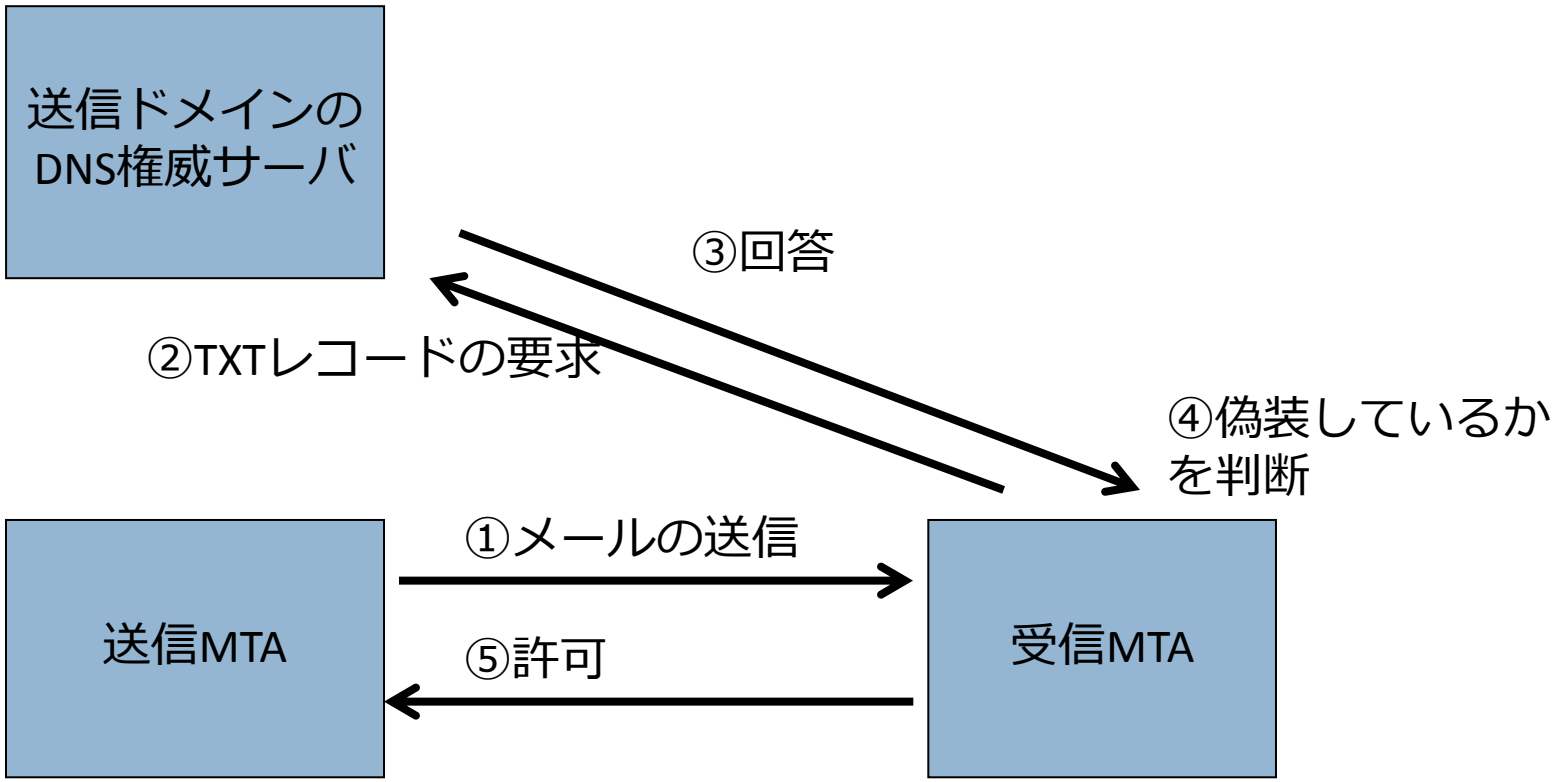
- 協調型フィルタ
  - 迷惑メールは同じ文面のメールを大量にたくさんの人に送りつけるため、そのメールを受け取った人が、そのメールが迷惑メールであると判断した結果を公開すれば、他の人もその結果を利用できる。
  - 判定結果を公開データベースに登録する仕組みと、公開データベースに問い合わせる仕組みを用意する。
  - Vipul's Razor、 Cloudmark(商用)

メールの内容

- URIDNSBL
  - 迷惑メールの目的はメールを読んだ人をウェブサイトに誘導すること。
    そのために、URIをメール本文に載せる。
  - 対策としてURIに対するDNSBLを用意する。
  - →リダイレクトや短縮URLによる回避の問題がある

送信ドメイン認証

- 送信者のメールアドレスに基づく認証方法
  - SPF, Sender ID, DomainKeys, DKIM



送信ドメイン認証

- 直接的な迷惑メール対策ではない
  - スパマー自身がSPFを宣言しているケースがある
  - SPFをpassしたからといってスパムではないとは言い切れない。
- 詐称を検出する仕組み
  - 詐称しているスパムは多いため、結果として迷惑メール対策になっている。

誤判定

		迷惑メール対策ソフトによる判定結果	
		正常なメール（ハム）	迷惑メール（スパム）
人の判断	正常なメール	True Positive（正判定）	False Positive（誤判定）
	迷惑メール	False Negative（誤判定）	True Negative（正判定）

迷惑メール対策を正しく運用するには

- False Positive（正常なメールをスパムと誤判定）を減らす
  - False Positiveが発生すると対策システムが信用できなくなる
  - False Negativeを許容する
- 1つの要因のみでは判断しない
  - 複数の要因の加算で判断する

SpamAssassinによる対策

- SpamAssassinのスパムらしさの判定の対象
  - 経路情報
  - エンベロープ情報
  - メールの内容
  - 送信ドメイン認証
  - →SMTPセッションの挙動以外はほとんどSpamAssassinによるテストの対象になる
- False Positiveを減らすために複数の要因の加算で判断する

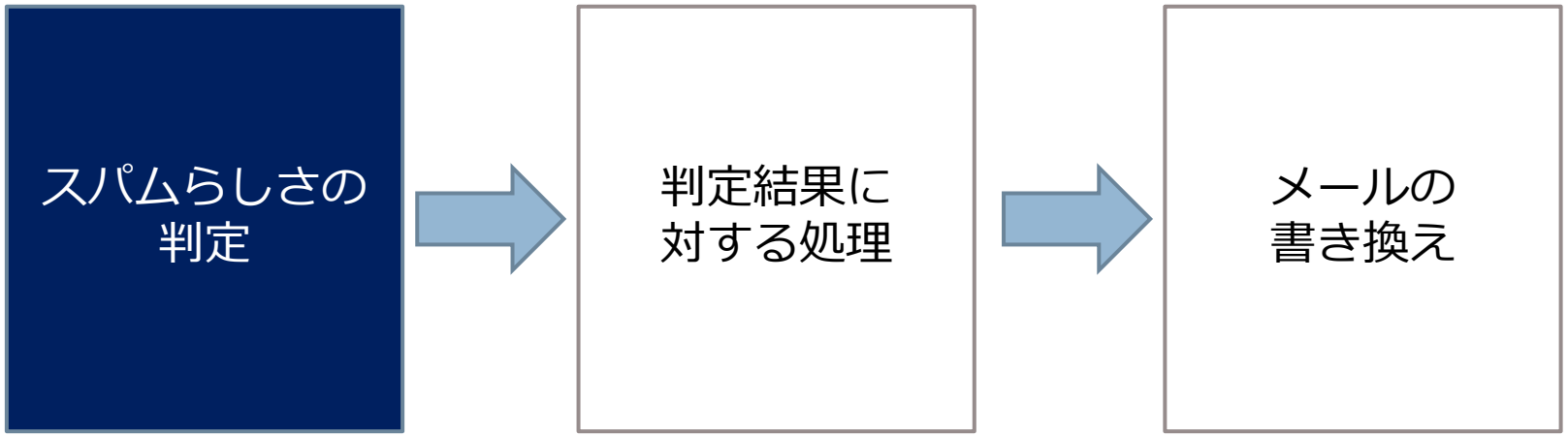
SpamAssassin紹介編

SpamAssassinとは

- スパムらしさを判定するメールフィルタ

SpamAssassinでできること

- メールのスパムらしさの判定
  - 様々な試験の実施
  - スパムらしさのスコアの計算および判定

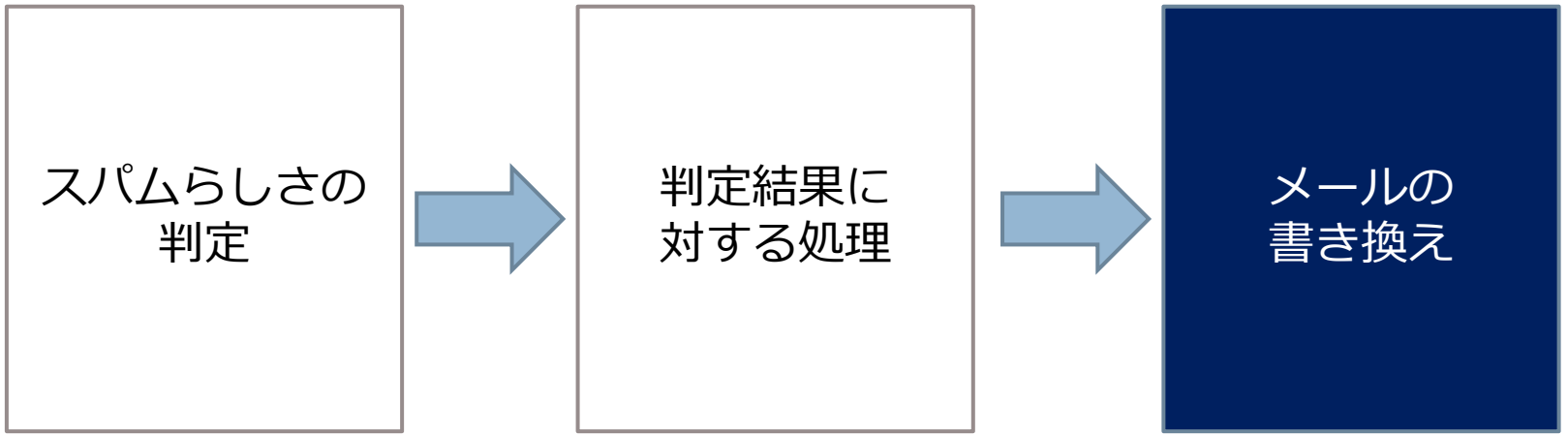


SpamAssassinでできること

- 判定結果に対する処理
  - ベイジアンフィルタの自動学習

SpamAssassinでできること

- 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への追加
  - スパムメールのカプセル化



SpamAssassinでできること

- スコアや判定結果のヘッダへの追加
  - X-Spam-Flag: YES
  - X-Spam-Level: *******
  - X-Spam-Status: Yes, score=7.3, ...
- 他のソフトウェアでこのヘッダフィールド値を利用できる。

SpamAssassinでできないこと

- スパムの除去
- スパムの振り分け
- バウンスメールの送信

SpamAssassinでできないこと

- スパムの除去や振り分けをしたい
  - 他のソフトウェアと組み合わせる。
  - →実践編へ

SpamAssassinの特徴

- 様々なテストを総合的に行う
  - 最新のルールファイルでは900個のテストあり
- パターンテスト
  - 文字列の一致（Perl正規表現を利用可能）
- ネットワークテスト
  - 経路情報（ヘッダのReceivedフィールド）
  - DNS/URIDNSブラックリスト
  - 協調型データベース
  - 送信ドメイン認証（SPF, DKIM）
- ベイズテスト

SpamAssassinの特徴

- 一つのテスト結果だけでは判断しない
  - 誤判定の要因
- 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する
  - 0.1 + 0.5 + 1.0 + 3.0 + 2.0 + 1.0 = 6.6
  - →スパムっぽいよ
- False Positive（正常なメールをスパムと誤判定すること）を少なくできる

SpamAssassinのバージョン

- 最新版
  - SpamAssassin 3.3.1 - 2010年3月19日リリース
- 旧バージョン
  - SpamAssassin 3.2.5 - 2008年6月12日リリース

動作環境 (SpamAssassin 3.3.1)

- Perl 5.8.5以降がインストールされたUNIX系OS（Linux/*BSDも含む）
  - Perl 5.8.8, 5.8.10, 5.10.1推奨
  - Perl 5.12では現バージョンは動作しない
- Windows環境でも動作する

日本語メールの判定

- 日本語対応パッチを当てるとよい
  - http://www.emaillab.jp/spamassassin/ja-patch/

SpamAssassin概要編

SpamAssassinの構成

- Perlモジュールライブラリ
- ツール
  - 判定ツール、ベイズ学習ツール等
- プラグイン
- ルールファイル

Perlモジュールライブラリ

- SpamAssassinの本体
  - Mail::SpamAssassin
- Perlのプログラムに組み込むことができる

```
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

spamassassin	メールがスパムであるかどうかを判定する。 フロントエンドプログラム。
spamc	メールがスパムであるかどうかを判定する。 spamdと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	ベイジアンフィルタの学習を行わせる。
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。
sa-comple	BODYルールのコンパイル

spamassassin

- スタンドアローンのスパム判定プログラム
- 標準入力からメールを受け取り、標準出力に結果のヘッダを付けて出力する。
- Perlのプログラムであるため、起動のオーバーヘッドがある。

spamassassinコマンド実行例

```
$ spamassassin < spam.eml
Return-Path: <ohydhitmyrpnr@example.com>
X-Spam-Checker-Version: SpamAssassin 3.3.1 (2010-03-16) on mercury.emaillab.jp
X-Spam-Flag: YES
X-Spam-Level: ***************
X-Spam-Status: Yes, score=15.6 required=7.0 tests=BODY_JA_TOSAITO,
    FROM_ILLEGAL_CHARS,FSL_HELO_BARE_IP_1,RCVD_IN_BRBL_LASTEXT,RCVD_IN_PBL,
    RCVD_NUMERIC_HELO,RDNS_NONE,SUBJECT_NEEDS_ENCODING,T_URIBL_BLACK_OVERLAP,
    URIBL_BLACK,URIBL_JP_SURBL autolearn=spam version=3.3.1
X-Spam-Report:
    *  1.4 FSL_HELO_BARE_IP_1 FSL_HELO_BARE_IP_1
    *  2.1 FROM_ILLEGAL_CHARS From: has too many raw illegal characters
    *  0.9 RCVD_NUMERIC_HELO Received: contains an IP address used for HELO
    *  3.6 RCVD_IN_PBL RBL: Received via a relay in Spamhaus PBL
    *      [119.48.195.193 listed in zen.spamhaus.org]
    *  1.9 URIBL_JP_SURBL Contains an URL listed in the JP SURBL blocklist
    *      [URIs: pinroom.com]
    *  1.8 URIBL_BLACK Contains an URL listed in the URIBL blacklist
    *      [URIs: pinroom.com]
    *  0.2 BODY_JA_TOSAITO BODY: TOSAITO
```

spamdとspamc

- クライアント/サーバ型のスパム判定プログラム
- spamdがデーモンとして常駐する。
- spamcはクライアントとして動作し、spamdにメールを渡してスパムの判定を依頼する。
- spamcはc言語で書かれているため、起動のオーバーヘッドが小さい。

sa-learn

- ベイジアンフィルタに手動で学習させるプログラム。
  - $ sa-learn --spam --progress ./spam/
    1% [=] 3.23 msgs/sec 09m34s LEFT

sa-update

- ルールファイルを最新のものに更新するプログラム
- スパムの手法は常に変化するため、対応する新しいルールが作られる。
- →最新のルールへの更新が必要
- 実行例
  - # sa-update

sa-compile

- BODYルールをコンパイルする。
- BODYルールの正規表現をC言語のプログラムに変換して、コンパイルする。
- ルール判定の高速化
- 残念ながら日本語には対応していない。

プラグイン

- SpamAssassinのテストエンジンはプラグインにより実装されている

プラグイン

- Mail::SpamAssassin::Plugin::*
  - AccessDB, AntiVirus, ASN, AutoLearnThreshold, AWL, Bayes, BodyEval, BodyRuleBaseExtractor, Check, DCC, DKIM, DNSEval, FreeMail, Hashcash, HeaderEval, HTMLEval, HTTPSMismatch, ImageInfo, MIMEEval, MIMEHeader, OneLineBodyRuleType, PhishTag, Pyzor, Razor2, RelayCountry, RelayEval, ReplaceTags, Reuse, Rule2XSBody, Shortcircuit, SpamCop, SPF, Test, TextCat, URIDetail, URIDNSBL, URIEval, VBounce, WhiteListSubject, WLBLEval,

プラグイン

- 自動学習関連
  - AutoLearnThreshold、 AWL
- パターンテスト関連
  - WhitelistSubject、 MIMEHeader、 ReplaceTags、 HTTPMismatch、 URIDetail
- 国、 言語関連
  - RelayCountry、 TextCat

プラグイン

- ネットワークテスト関連
  - Razor2、 SpamCop、 URIDNSBL
- 送信ドメイン認証
  - SPF、 DKIM
- ベイズ
  - Bayes
- その他
  - AccessDB、 AntiVirus、 FreeMail、 PhishTag

プラグインの制御ファイル

- プラグインの制御ファイル
  - /etc/mail/spamassassin/*.pre
- /etc/mail/spamassassin/init.preの例

```
# URIDNSBL - look up URLs found in the message against several DNS
# blocklists.
#
loadplugin Mail::SpamAssassin::Plugin::URIDNSBL

# SPF - perform SPF verification.
#
#loadplugin Mail::SpamAssassin::Plugin::SPF
```

ルールファイル

- 標準のルールファイル
  - /var/lib/spamassassin/
  - 約900個のルールが用意されている
  - sa-updateコマンドでダウンロードする
- ユーザー定義ルールファイル
  - /etc/mail/spamassassin/*.cf

ルールファイル

- ルールファイルに各種テストを実行するルールが記述されている。
- テストの実行エンジンはプラグインであるため、無効にしたプラグインに対応するテストやルールは実行されない。
  - 例えば、SPFプラグインを無効にしたら、SPFに関連したルールのテストは実行されない。

テストの種類

- パターンテスト
- ネットワークテスト
- ベイズテスト

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

パターンテスト

テスト	説明
header	ヘッダ （MIME復号化済み）
body	ボディのテキストパートのみ （MIME復号化済み、HTMLタグ等の除去あり）
uri	ボディに記述されたURI
rawbody	ボディのテキストパートのみ （MIME復号化済み）
full	生メッセージ全体 （MIME復号化なし）

パターンテストの例

- ヘッダのSubjectフィールドに「% off」を含む
  - header SUBJECT_OFF　Subject =~ /% off/i
    describe SUBJECT_OFF　Subject contains a word '% off'
    score SUBJECT_OFF　　2.0
- 本文に「出会い」を含む
  - body　　BODY_JA_DEAI　　/出会い/
    describe BODY_JA_DEAI　DEAI
    score　BODY_JA_DEAI　　0.5

パターンテストの記述上の注意点

- fullテストを使わない。
  - MIME復号化前の添付ファイルも含めた評価を行うため、非常に負荷がかかる。
- bodyテストにおいて /.*/のような行末までマッチするようなパターンを使わない。
  - 繰り返しには「{,5}」のように制限をかける
- →この2点を誤ると、過大なCPU負荷がかかる恐れがある。

ネットワークテスト

- IPアドレスやホスト名
  - Receivedヘッダの解析を実施し、経路情報やHELO/EHLOのドメイン名などの解析も行う。
- 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
  - 付属のテストは応答時間的に実質的に使い物にならないので利用しない方がよい
- 送信ドメイン認証（SPF, DKIM）

ネットワークテスト

- SpamAssassinの処理時間の長さはネットワークテストにおけるDNSクエリの応答によるもの
  - 専用にDNSキャッシュサーバを用意する
    - Unboundがおすすめ

ベイズテスト

- ハム・スパム共に200通以上学習したら判定開始
- 自動学習機能
  - デフォルトでスコア0.1以下でハムとして学習
  - デフォルトでスコア12以上でスパムとして学習
- 確率とスコア
  - score BAYES_00 -1.9
    score BAYES_05 -0.5
    score BAYES_20 -0.001
    score BAYES_40 -0.001
    score BAYES_50 0.8
    score BAYES_60 1.5
    score BAYES_80 2.0
    score BAYES_95 3.0
    score BAYES_99 3.5

特殊なテスト（プラグイン）

- メールが中継された国の一覧（RelayCountry）
- AS番号（ASN）
- URI（URIDetail）
- 画像情報（ImageInfo）
- バウンスメール（VBounce）

METAテスト

- META
  - 複数のテスト結果の組み合わせ

おまけ: 日本語対応パッチ

日本語対応パッチの機能

- normalize_charsetオプションの改良
  - 日本語でテストルールが書けます
    - body　HOGOHOGE　/ほごほげ/
  - オリジナルでは十分に機能していない
  - 文字エンコーディング推定処理の強化
- report_charsetオプションの改良
- ベイジアンフィルタのUTF-8の文字処理の改良
- ベイジアンフィルタの日本語対応
  - 日本語の分かち書き

ベイジアンフィルタの日本語対応

- 日本語は単語毎に区切られていない言語。
  - 「私の名前は中野です」
- ベイズ解析を行うためには分かち書きが必要。
  - 「私　の　名前　は　中野　です」

分かち書き処理プラグイン

- 分かち書き処理はSpamAssassinのプラグインとして実装。
- プラグインを２つ用意している。
  - Tokenizer::MeCab
  - Tokenizer::SimpleJA

Tokenizer::MeCab

- 形態素解析エンジンMeCabの利用
- 分かち書きの結果
  - 「私の名前は中野です」
  - →「私　の　名前　は　中野　です」
  - 「すももももももものうち」
  - →「すもも　も　もも　も　もも　の　うち」

Tokenizer::SimpleJA

- 文字種による区別
- 他のソフトウェアのインストールは不要
- 分かち書き結果
  - 「私の名前は中野です」
  - →「私　の　名前　は　中野　です」
  - 「すももももももものうち」
  - →「すももももももものうち」

SpamAssassin導入編

インストール

- 日本語対応パッチがあるので適応してインストールする。
  - http://www.emaillab.jp/spamassassin/ja-patch/
- 細かい話は省略
- インストールしたら、sa-updateを実行する。
  - SpamAssassin 3.3.0からルールファイルは同梱されなくなっている。
  - sa-updateによりルールファイルを取得する。

設定ファイル

- /etc/mail/spamassassin/local.cf
- /etc/mail/spamassassin/*.cf

必要最小限の設定

- UTF-8に変換してから評価する機能
  - normalize_charset 1
- 判定スコアの設定
  - required_score 5
  - 運用当初は高めに設定し、精度が上がってきたら徐々に下げる。
  - 最適な閾値が5になるように各ルールのスコアは調整されている。
  - 5から7の間がおすすめ

必要最小限の設定

- ネットワークの設定
  - internal_networks 192.0.2.0/24
- レポートオプションの設定
  - report_safe 0
  - これを設定しないとスパム判定されたメールはカプセル化されたレポートメール形式になる。

プラグインの選択

- 利用するプラグインを選び、有効にする。
  - /etc/mail/spamassassin/*.pre
  - init.pre
  - v310.pre
  - v312.pre
  - v320.pre
  - v330.pre
- 導入されたバージョン毎に設定ファイルがある

例：SPFの有効化

- /etc/mail/spamassassin/init.preを編集し、次の行を有効にする。
  - loadplugin Mail::SpamAssassin::Plugin::SPF

例：協調型データベースの無効化

- /etc/mail/spamassassin/v310.preを編集し、次の行を無効（コメントアウト）にする。
  - #loadplugin Mail::SpamAssassin::Plugin::DCC
  - #loadplugin Mail::SpamAssassin::Plugin::Pyzor
  - #loadplugin Mail::SpamAssassin::Plugin::Razor2

ユーザー定義ルール

- 必要に応じてルールを記述する。
  - ```
    body      BODY_JA_DEAI          /出会い/
    describe BODY_JA_DEAI          DEAI
    score     BODY_JA_DEAI          0.5
    ```
  - ```
    header TOO_DETAILED_DATE_TZ  Date =~ /¥s[-+]¥d{2}(?:[1-24-5]¥d|¥d[1-9])/
    describe TOO_DETAILED_DATE_TZ Date: timezone is too detailed
    score TOO_DETAILED_DATE_TZ   3.0
    ```
- ユーザー定義ルールはlocal.cfに記述するのではなく、別ファイルに記述するのがおすすめ。
  - 拡張子がcfであれば設定ファイルと認識する

設定ファイルの検査

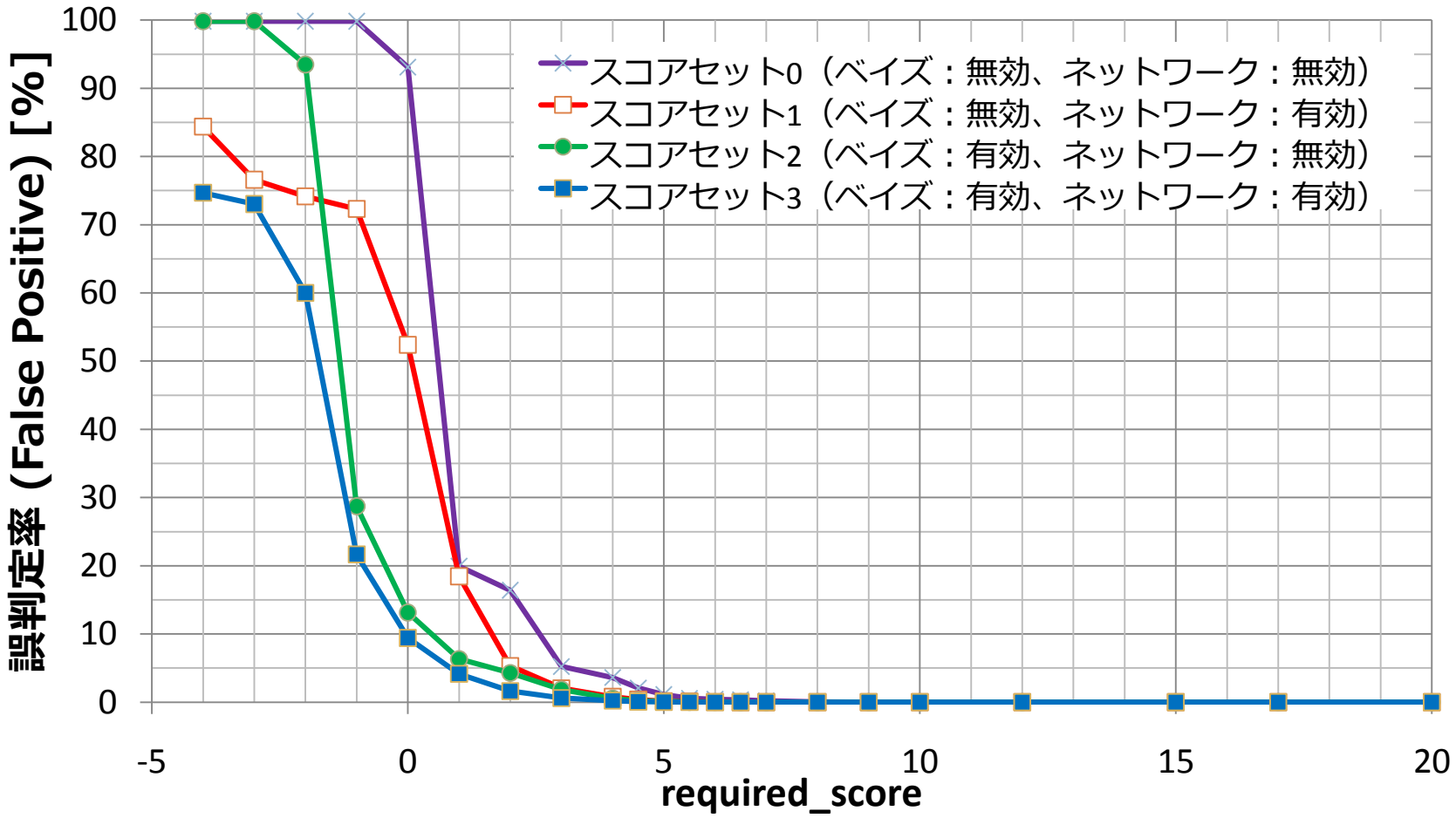
- 設定ファイルの記述を変えたら必ずspamassassin --lintを実行すること
- 何もエラーが出なければよい

おまけ: required_scoreの最適値

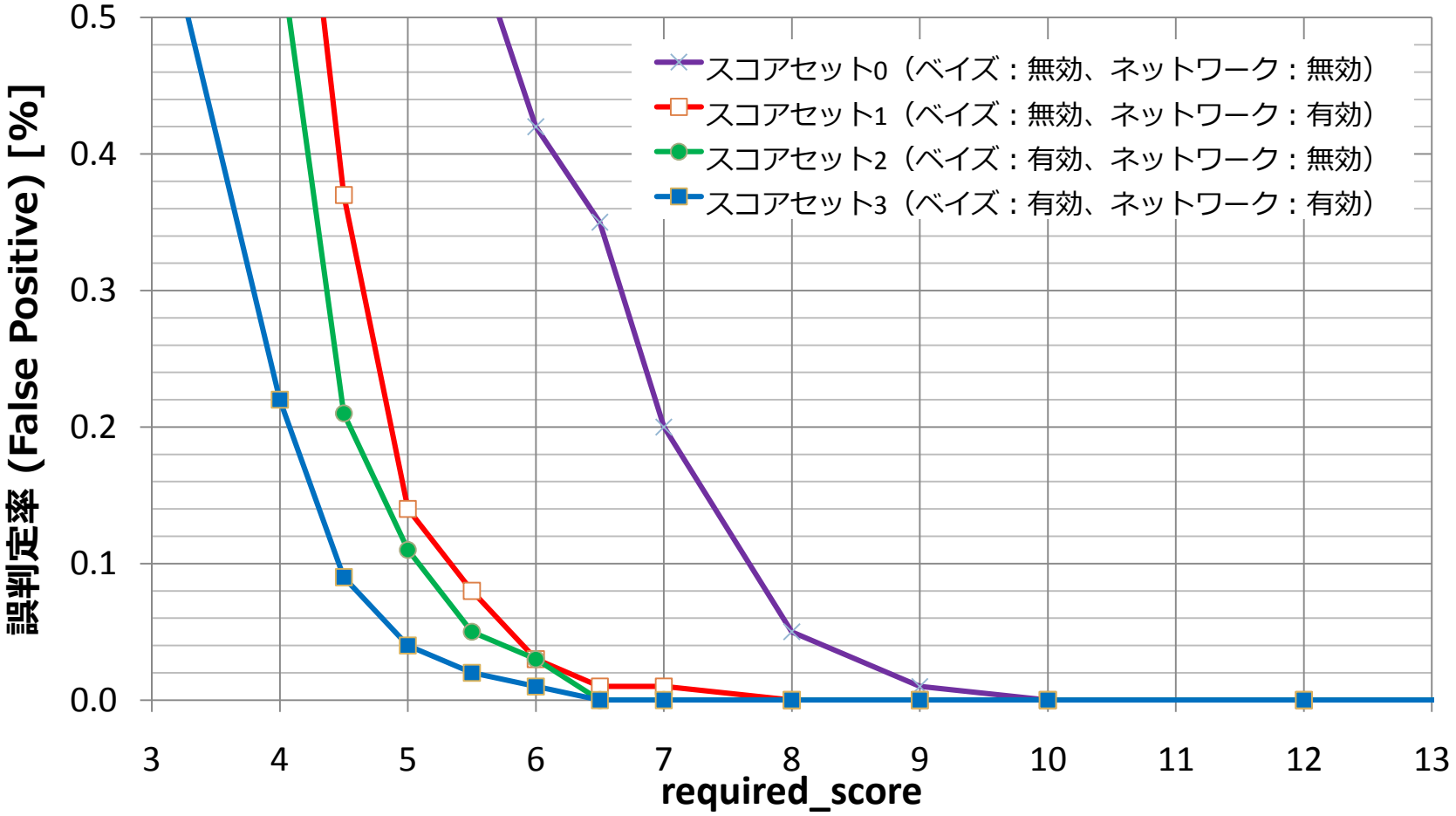
統計情報によるスコアの調整

- メジャーバージョンのリリース時に各ルールのスコアの調整を行うためにマスチェックを実施
- マスチェック:
  - 65320通のメール（ハム: 21186、スパム: 44134）
  - required_scoreの設定値毎に判定させる。
  - required_scoreが5.0のときに、False Positive 0.05%程度になるように調整
  - 統計結果がSpamAssassinの配布物に同梱

False Positive



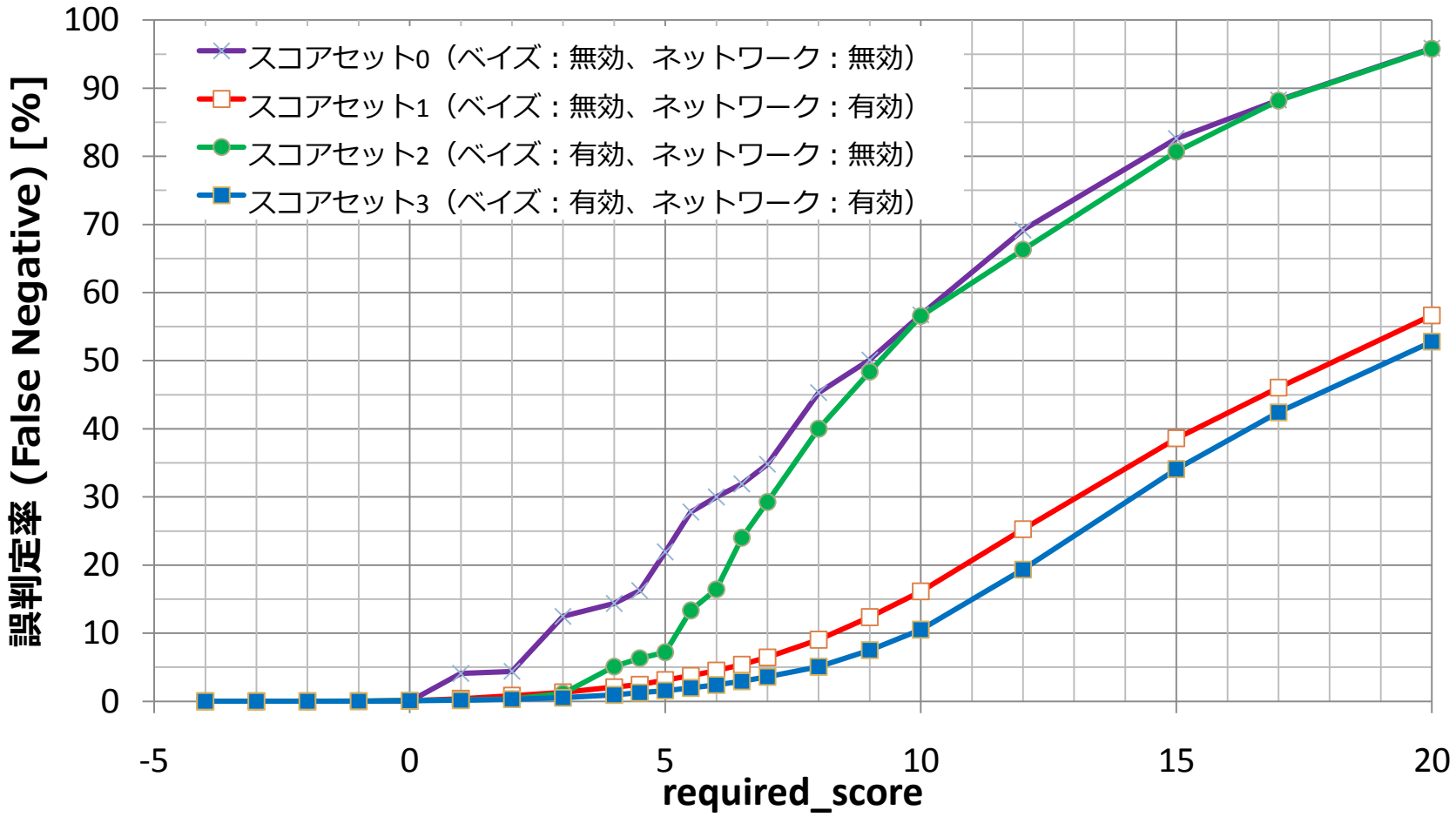
False Positive



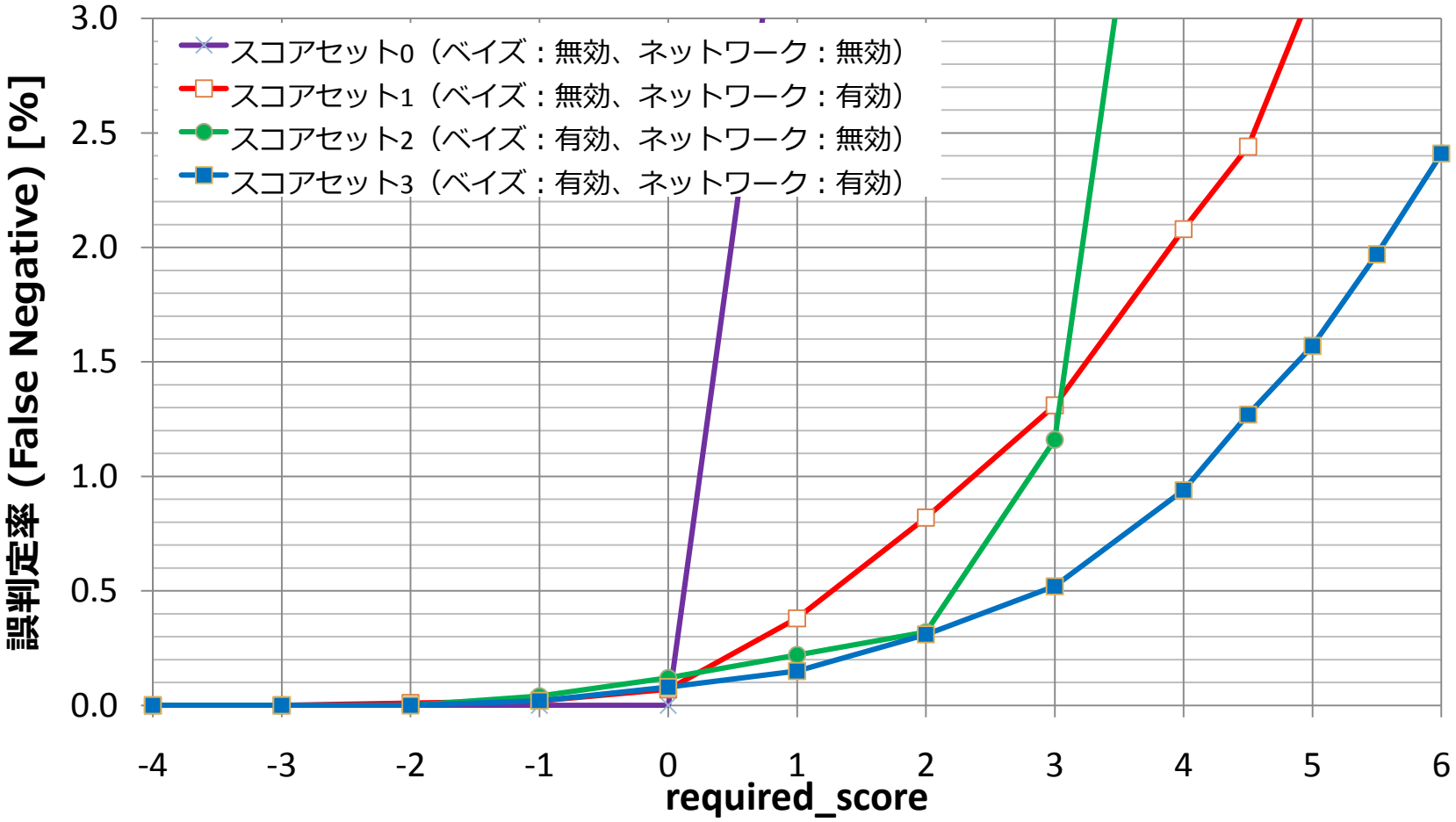
False Positive

- required_score=10
  - False Positiveなし
  - ベイズ自動学習はスコア12以上のときに働く
- required_score=5
  - False Positiveは0.04%（ベイズ＋ネットワーク）
- required_score=4
  - False Positiveは0.22%（ベイズ＋ネットワーク）
  - 急に増加する
  - 5未満にしてはいけない

False Negative



False Negative



False Negative

- required_score=-4
  - False Negativeなし
- required_score=0
  - False Negative 0.08%（ベイズ＋ネットワーク）
  - ベイズ自動学習はスコア0.1以下のときに働く
- required_score=5
  - False Negative 1.6%（ベイズ＋ネットワーク）

スコアのまとめ

- required_scoreの設定値の推奨値
  - 5.0～7.0
- False Positiveを0.05%未満に減らす
- False Negativeを許容する

SpamAssassin実践編

SpamAssassinの利用

- SpamAssassin単体では判定しかできない
- スパム判定されたメールの振り分けや削除を行うには他のソフトウェアと組み合わせる
- サーバ側
  - MTA
  - MDA/LDA
- クライアント側
  - MRA
  - MUA/メーラー

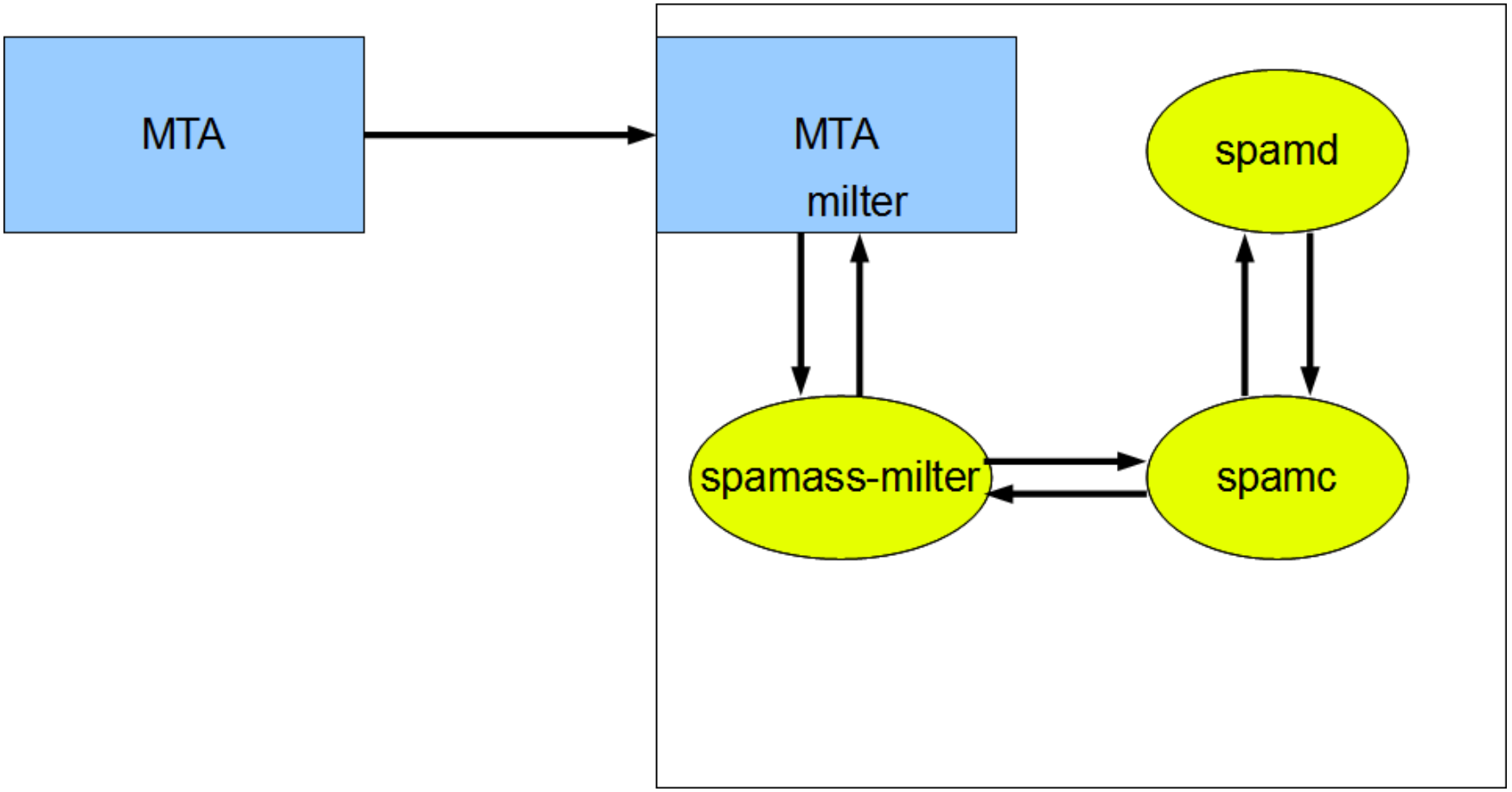
MTAでの利用

- spamass-milter
- amavisd-new

spamass-milter

- SpamAssassin専用のmilterプログラム
- できること
  - SpamAssassinの判定結果のヘッダを付与する。
  - 指定したスコア以上のものを拒否することもできるが、この機能は使うべきではない。
- セキュリティホールあり
  - 保守されていないのでおすすめしない。
  - 使う場合は修正パッチが当たってるものを使うか、パッチを拾ってきて自分で適応するか

spamass-milter



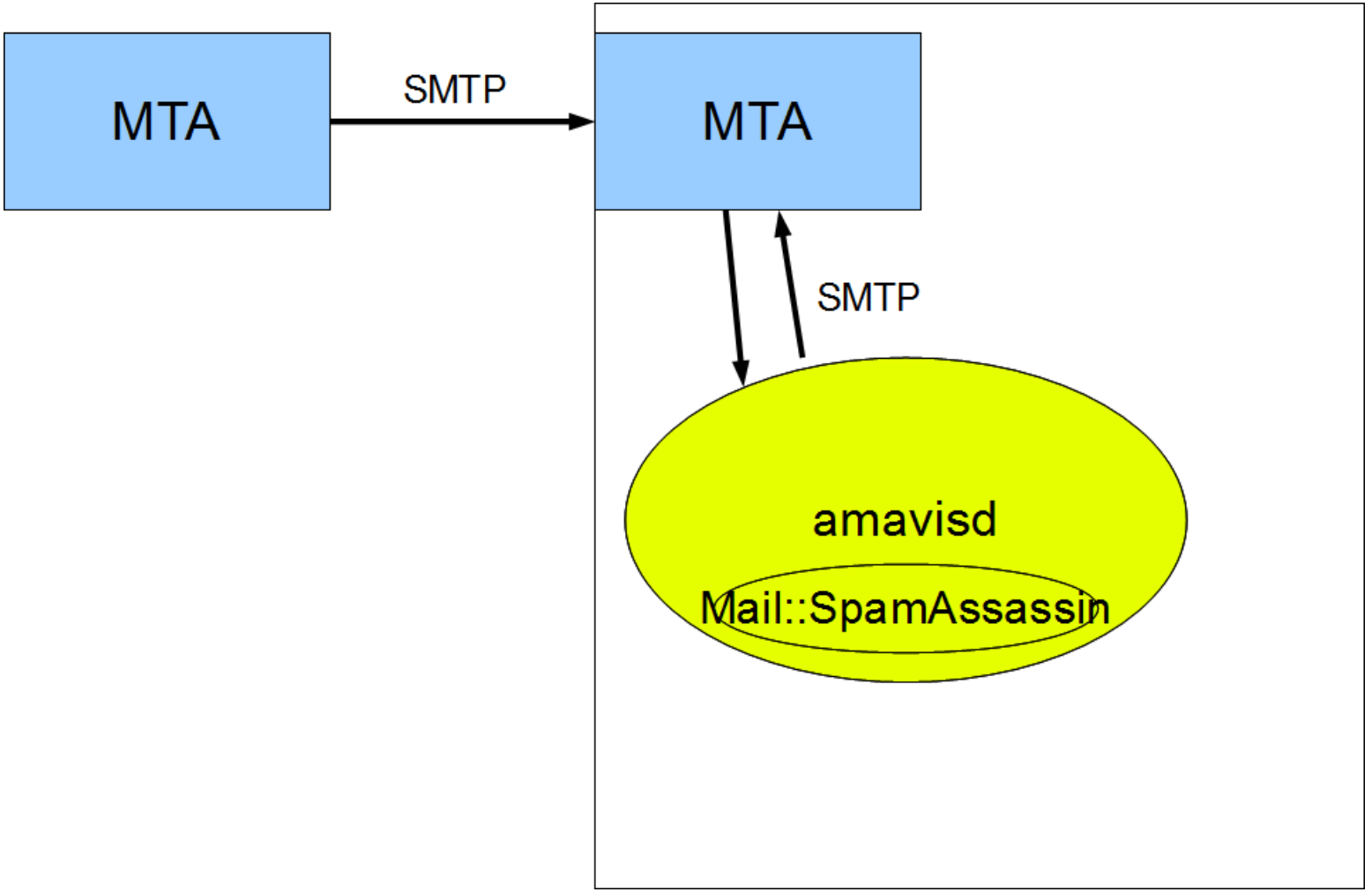
amavisd-new

- SpamAssassinを組み込んだ総合メールフィルタ
  - 不正なヘッダチェック
  - 添付ファイルの形式や拡張しのチェック
  - ウイルスチェック
  - スパムチェック（SpamAssassin）
  - ホワイトリスト/ブラックリスト

amavisd-new

- smtpサーバとして動作する。
- MTAと組み合わせて使用することもできる。
  - Postfixのcontents_filterなど
- amavisd-milterもある

amavisd-new



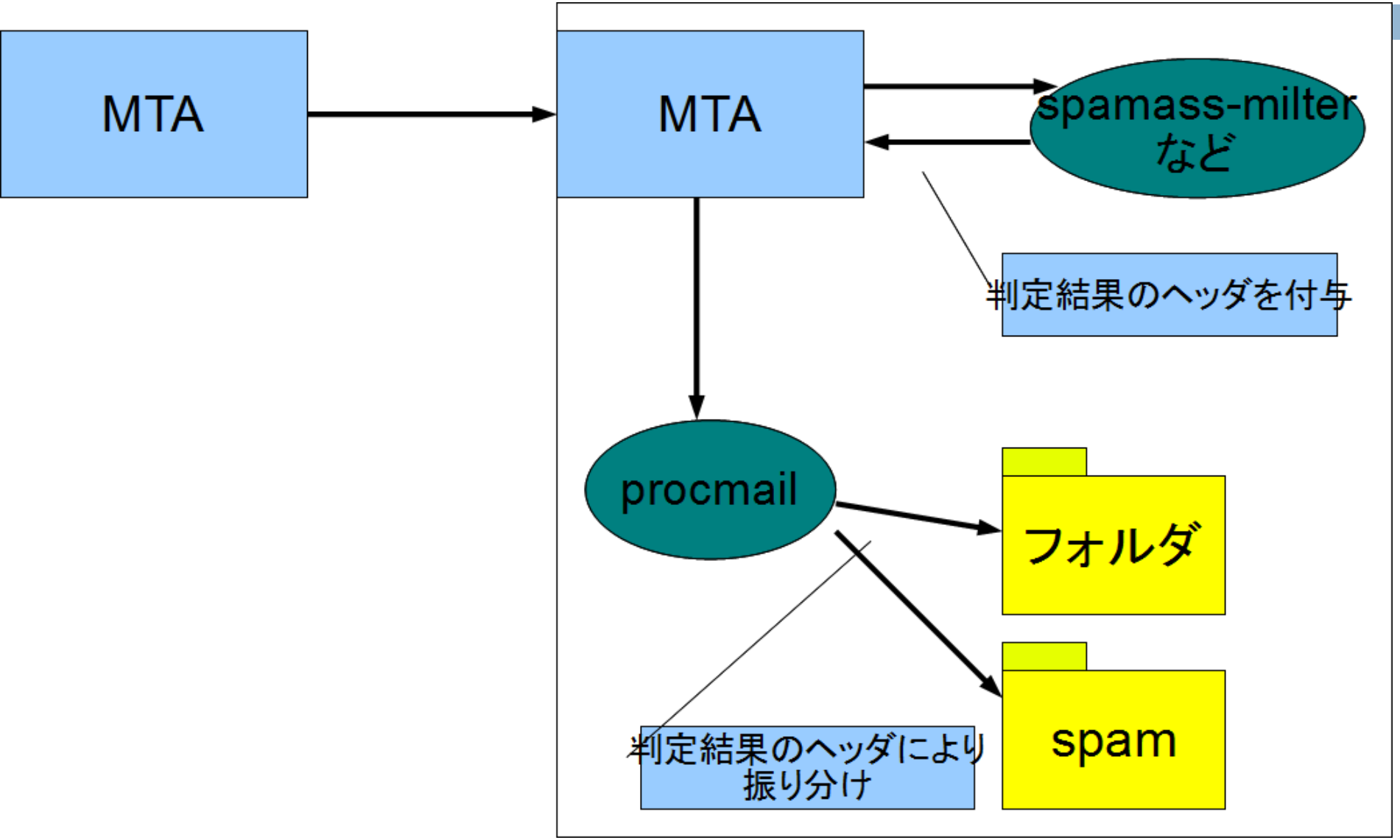
MDA/LDAでの利用

- MDA (Message Delivery Agent)
- LDA (Local Delivery Agent)
- 振り分けできるMDA/LDA
  - procmail
  - maildrop
  - sieve機能
    - Dovecotのdeliver+sieveプラグイン(pigeonhole)など

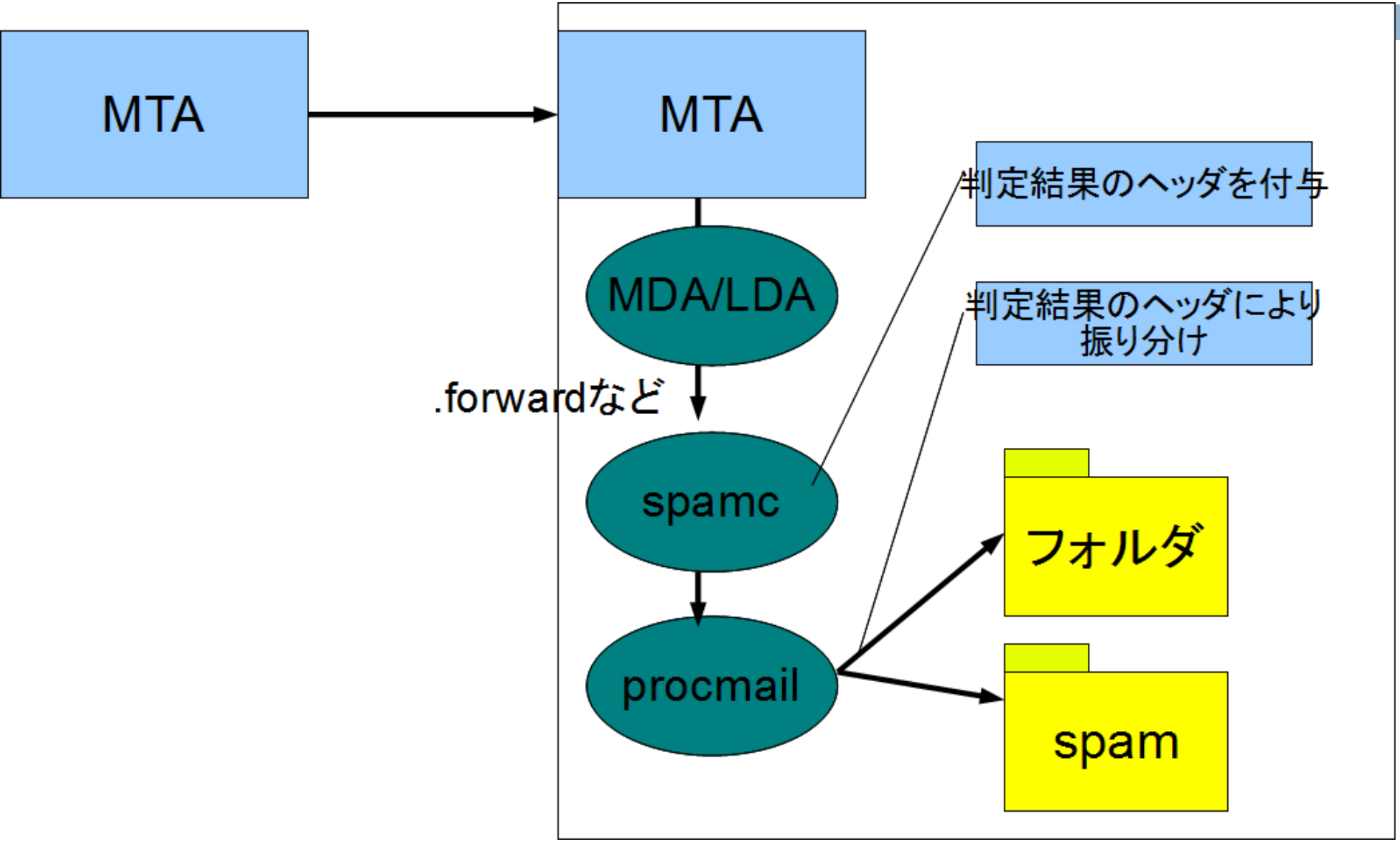
MDA/LDA

- 上流で付与したSpamAssassinの判定結果のヘッダによりMDA/LDAで振り分けする。
- 上流
  - MTAレベル（spamass-milter, amavisd-new）
  - メールボックスレベル.forwardなどでspamcやspamassassinコマンドを呼び出す
- 判断するヘッダ
  - X-Spam-Flag: YES
  - X-Spam-Level: *******

MDA/LDA


MTA
MTA
spamass-milter
など
判定結果のヘッダを付与
procmail
フォルダ
spam
判定結果のヘッダにより
振り分け

MDA/LDA


MTA
MTA
MDA/LDA
.forwardなど
spamc
procmail
判定結果のヘッダを付与
判定結果のヘッダにより
振り分け
フォルダ
spam

maildropの記述例

- .mailfilterファイル
  - DEFAULT=$HOME/Maildir/
    if (/^X-Spam-Level: ¥*{12,}/)
    　to $HOME/Mail/spam/
    if (/^X-Spam-Level: ¥*{7,}/)
    　to $HOME/Mail/spammy/
    to $HOME/Maildir/

sieveの記述例

- Dovecot 2.0+Pigeonhole の .dovecot-sieveファイル
  - require "fileinto";

```
if header :contains "X-Spam-Level" "**********" {
  fileinto "spam";
  stop;
}
if header :contains "X-Spam-Level" "*******" {
  fileinto "spammy";
  stop;
}
keep;
```

クライアント側

- fetchmail + procmail/maildrop
- メーラーの振り分け機能

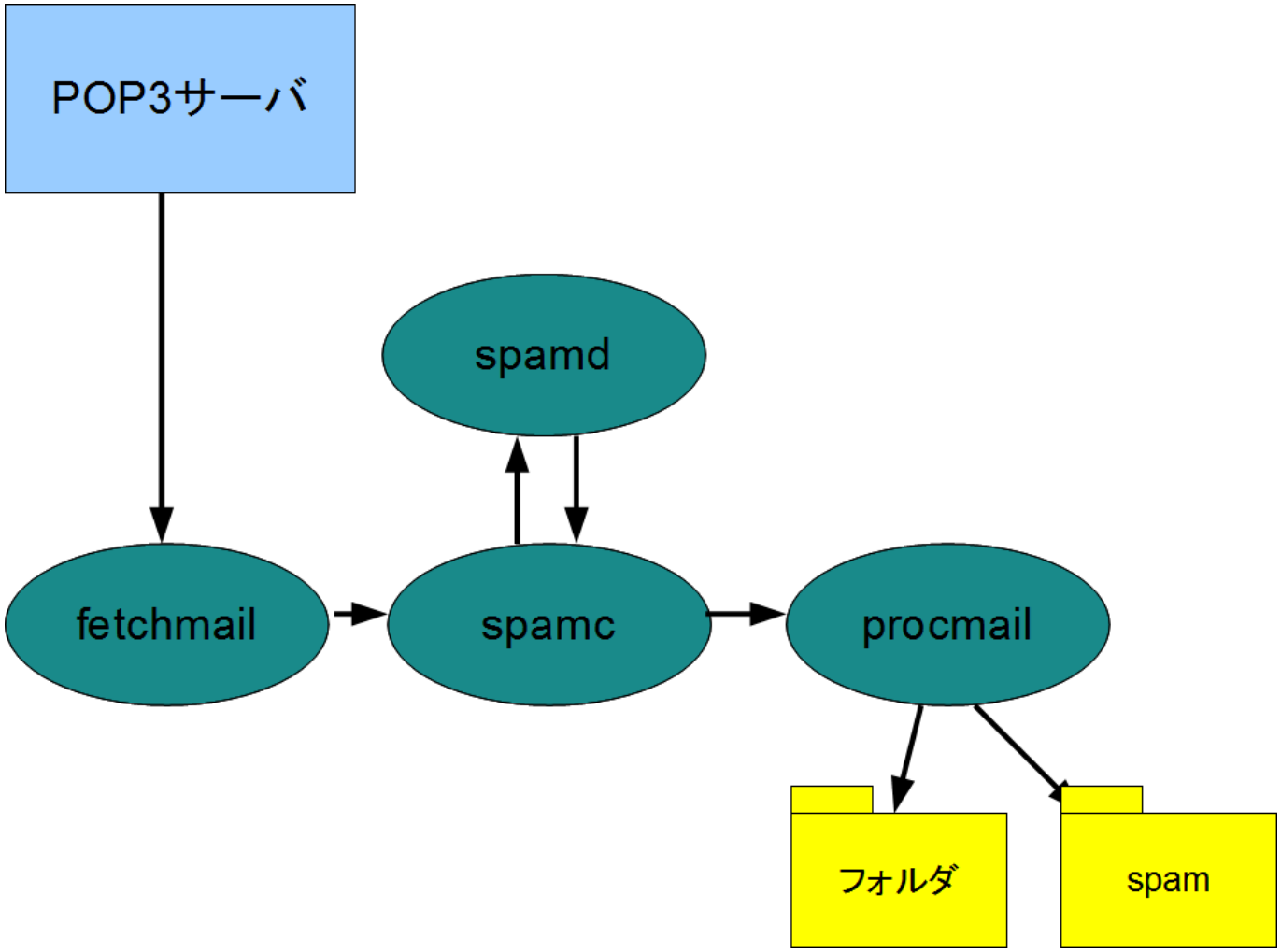
fetchmail

- POP3/IMAPサーバからメールを取得して、配送するプログラム
- spamc/spamdにより判定させて、procmailなどでメールボックスに配送させる。
- procmail/maildropなどで振り分けを行うことができる。

fetchmail

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

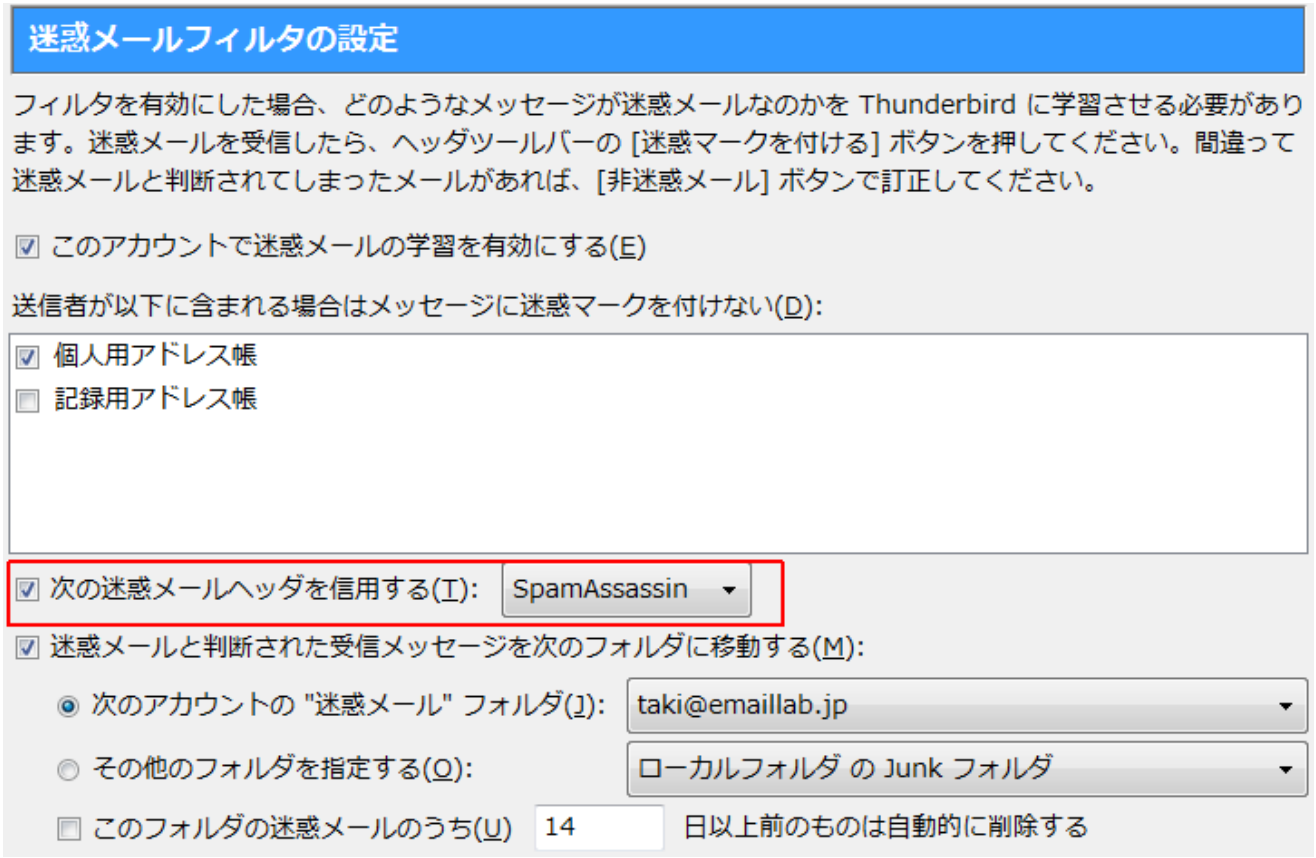
fetchmail



メーラーでの振り分け

- Thunderbird
  - SpamAssassinのフラグを信用するオプションあり
  - MTAやMDA/LDAなどで付与したSpamAssassinの判定結果を利用する。

SpamAssassin



日本語ルール自動作成

日本語ルール作成スクリプト

- 目的
  - 日本語のルールの自動作成
- スクリプトの所在
  - http://spamassassin.jp/download/experimental/taki/
  - sa-tokenizer.pl　　--- トークナイザー
  - sa-ja-testmaker.pl --- テスト生成スクリプト
- ハムとスパムを大量に食わせて統計処理する。

作成されたルール

```
# BODY_JA_HITOZUMA: 人妻 spam=2583/1325054, ham=1/1841092, ratio=0.00194
body     BODY_JA_HITOZUMA            /人妻/
describe BODY_JA_HITOZUMA            HITOZUMA
score    BODY_JA_HITOZUMA            0.6

# BODY_JA_ANATA: 貴方 spam=2645/1325054, ham=11/1841092, ratio=0.00193
body     BODY_JA_ANATA               /貴方/
describe BODY_JA_ANATA               ANATA
score    BODY_JA_ANATA               0.6

# BODY_JA_ICHIHACHIMIMAN: １８未満 spam=2446/1325054, ham=0/1841092,
ratio=0.00184
body     BODY_JA_ICHIHACHIMIMAN      /１８未満/
describe BODY_JA_ICHIHACHIMIMAN      ICHIHACHIMIMAN
score    BODY_JA_ICHIHACHIMIMAN      0.6

# BODY_JA_DEAI: 出会い spam=2444/1325054, ham=9/1841092, ratio=0.00179
body     BODY_JA_DEAI                /出会い/
describe BODY_JA_DEAI                DEAI
score    BODY_JA_DEAI                0.5
```

作成されたルール

- テスト名称（ローマ字）を自動生成
- 出現頻度によりスコアの割り付け

```
# BODY_JA_DEAI: 出会い spam=2444/1325054,
#    ham=9/1841092, ratio=0.00179
body      BODY_JA_DEAI                  /出会い/
describe BODY_JA_DEAI                  DEAI
score     BODY_JA_DEAI                  0.5
```

日本語対応ルール配布サイト

sa-update

- sa-updateはチャネルを指定することにより、標準以外のサイトからルールを更新できる。
- 日本向けのルールを試験的に公開中
  - チャネル名: spamassassin.emaillab.jp
  - 説明サイト
    - http://www.emaillab.jp/spamassassin/sa-update/
- 現時点では日本語の単語のBODYルール300個＋α

sa-update

- 公開鍵のインポート
  - # wget http://spamassassin.emaillab.jp/updates/GPG.KEY
    # sa-update --import GPG.KEY
- 更新方法
  - # sa-update --channel spamassassin.emaillab.jp
    --gpgkey 22B8A63A
- spamdを利用しているときはspamdを再起動
  - # /etc/init.d/spamassassin restart

おわり

- 資料等
  - http://www.emaillab.jp/spamassassin/ にて公開