

# 試してガッテン、メールとspam

---

松田陽一  
yoh@fcl.org

## 迷惑メールとは

---

- 見ず知らずの相手から送信される広告メール
- spam とも云う

### spam とは

- 元々は、米国製ランチョンミート
- 英国コメディドラマ Monty Python の番組で連呼された
  - しつこい勧誘メール

## spam とは

---

□ 「受信者が望まない、一斉に送出されるメール (UBE: Unsolicited Bulk Email)」 または 「受信者が望まない広告・宣伝などの商業メール (UCE: Unsolicited Commercial Email)」 と一般的に定義される。

□ 大概は

○- 詐欺 (ex. Nigerian Scam)

▶ リンク名 Multimedia & Internet Dictionary

URI: <http://www.kaigisho.ne.jp/literacy/midic/data/k21/k211.htm>

○- えっち物物品販売 (ex. Viagra, Penis/Bust enlarge...)

○- 薬物、コピーソフト (warez) 等の違法物品販売

□ → 犯罪行為或はそれに近い、後ろめたい内容の営業行為

## spam の発信元

---

- 主に英語圏、中国語圏からのものが多い
- 日本語の迷惑メールはひところと比べると少なくなった

## 改めて spam とは

---

□悪意を内包し、ネットワークに撒き散らされる汚いデータ

○2003年5月に全世界のメール流通量の5割を越えた

○2004年4月には全世界のメール流通量の7割を占めると予測される

URI: <http://internet.watch.impress.co.jp/cda/news/2003/12/09/1413.html>

○→インターネットトラフィックの増加

○→回線業者やプロバイダ等の負担増

□内容の詐欺にひっかかる被害が絶えない

○→社会問題

## spam の特徴(1)

---

前野年紀先生の定義

リンク名 boycott spam mail

URI: <http://spam.qmail.jp/>

- 知らない相手からのメール
- 公開されているブラックリストにあるホストからのメール
- 公開されている spam メールの特徴リストにあるメール
- 発信者アドレスや From ヘッダを詐称しているメール
- 存在しない宛先へのメール
- 希望しない宣伝のメール
- spam フィルターが spam だと判定したメール
- しつこく何度も送ってくるメール

## spam の特徴(2)

---

### まつだの定義

- その内容の殆どが犯罪行為か、類似する後ろめたい内容
  - 発信元情報を隠蔽・詐称するケースが多い
    - ▷→→メールヘッダ情報の偽装
  - 甘い言葉、一目見てそれと判る内容
    - ▷通常メールとは異なる特徴がある
    - ▷→文章が使い回されており、決まり文句がある
    - ▷但し、文章はちよくちよく変更される
- web サイトへロボットアクセスを行い、機械的にメールアドレスを収集し、大量にメールを発信し、誰か一人でも引っかかるカモを待っている
  - 大量メール発信ツールを使用することが多い
    - ▷→→メールヘッダ情報或はメール本文にその特徴が記されていることがある

## spam はどうやって発信されるのか?

---

- どこからどうやってメールが来るの?
- どうして発信元情報を偽造できるの?
- どうして大量にメールを送信できるの?

spam と通常メール(ham: spamassassin ローカルな呼称)との違いはどうやって見分けるのか?

- ヘッダって何?本文って何?
- インターネットメールの仕組みを知ることが、この問題を解決する基本

## メールとは

---

- ヘッダと本文で構成される
- ヘッダと本文の間は改行のみの空行で区切られる
- ヘッダには本文に関する様々な情報が記される(実例)
  - ヘッダの詳細は後述。

# メールが配送される仕組み

---

メイラ(MUA)→smtpサーバ(MTA)→…  
smtp

…→smtpサーバ(MTA)→popサーバ→(MDA)→メイラ(MUA)  
smtp pop

□smtp: Simple Mail Transfer Protocol(RFC821,RFC2821)

▶電子メールを送信するためのプロトコル。

□pop: Post Office Protocol(RFC1939)

▶電子メールをスプールしているシステムから、TCP/IPプロトコルを使ってメールスプールの内容を読み出すためのプロトコル。

一般ユーザはこんなものの詳細を知る必要はない  
どういう仕組みなのかがわかればそれで良い

## メールは配送先メールアドレスを手がかりに配送する

---

yoh@flcl.org

「flcl.org というドメインに所属する yoh さんというユーザ」

flcl.org というドメイン宛へのメールはどこに送れば良い？

→DNSに問い合わせる(Domain Name System)

DNSのMXレコード(Mail Exchanger)

送信相手が所属するドメインに対応するメールサーバの名前を取得したら、更にメールサーバのIPアドレスを問い合わせる。

DNSのAレコード(Address)

DNSには他にもドメインとホストに関する各種情報が登録されている。

---

Aレコード(名前→IPアドレスの定義)

PTRレコード(IPアドレス→名前の定義)

NSレコード(ネーム・サーバの定義)

SOAレコード(ドメインのオーソリティ情報の定義)

AAAA という問い合わせは、IPv6 特有のもの:

リンク名 v6word:AAAA 【クアッドエー】

URI: [http://www.v6start.net/word\\_v6/20010622/2/](http://www.v6start.net/word_v6/20010622/2/)

ここはこれ以上突っ込まれると困ります(汗)

# まつだの環境

---

```
+-----+ +-----+ +-----+
|INTERNET+--+プロバイダ+--+eth1+ +eth0+--+
+-----+ +-----+ +-----+ +-----+ |
          Yahoo!BB  | ルータ  ||
                | (Linux BOX)| |
                +-----+ |
+-----+
| +-----+
+--+ローカルマシン|
| +----Linux-----+
| +-----+
+--+ローカルマシン|
| +----Linux-----+
| +-----+
+--+ローカルマシン|
| +----Win2k-----+
```

## メール送信の実験

---

Linux BOX 上にて稼動する Exim に向けて、メールを送信する。

Exim は debian のデフォルトメールサーバ。

Exim はあらゆるメールサーバへの配送を行う設定になっている。(Internet siteの設定)

## ヘッダの種類(1)

---

### Unix From 行

- smtp** サーバのやりとり(セッション)の際に、"MAIL FROM:" で指定された内容が記される。必ずしも最後に記録されるとは限らない。":" が付かないヘッダはこの行のみ。他は全て ":" が付いており、"DATA" でやりとりされる。

### Received: 行

- 発信元から受信者に届くまでにメッセージを処理した MTA によって付加される。従って、複数の MTA を経由して来たならばそれらが全てここに表示される。

### X-xxxx: 行

- "X-" から始まるヘッダは、任意の文字列が記入可能「だった」。(RFC821)現在、この規格自体は廃れており (obsolete)、非標準ヘッダであれば任意の文字列が記入可能であるが、但し重複は許されない、という規格になっている。(RFC2821)

### Date: 行

- 発信日時

## ヘッダの種類(2)

---

**From:** 行

- 発信人のメールアドレス(と、氏名或は類する文字列)

**Subject:** 行

- 題名

**Message-Id:** 行

- 日時、ファイル名、マシン名、ドメイン名などから生成される一意の文字列。特に指定しない場合は、MTAが自動的に付加する場合もある。

**To:** 行

- 宛先メールアドレス(と、氏名或は類する文字列)

**Cc:** 行

- 同報配送先メールアドレス(と、氏名或は類する文字列)

**Content-Type:** 行

- メール本文の属性。(Text/Plain; charset=iso-2022-jp)

**Content-Transfer-Encoding:** 行

- メール本文の記述方式。(7bit)

## ヘッダの実態(1)

---

- これらヘッダ情報の殆どは、一部の制約を除いて偽造可能である。
- smtpは基本的に認証機構を持たないので、情報の正確さが保証されない。

その昔、インターネットが平和だった頃、全ての smtp サーバは内外を問わず全ての利用者に開放されていた。

誰もが任意のsmtpサーバに対してメールを投げれば、宛先に配送された。

しかし、このような牧歌的な仕組みを悪用する者が現れた。

大量の広告メールを発信し、顧客を得ようとする。

大量の詐欺メールを発信し、カモが引っかかるのを待つ。

これが spam の始まり。

## ヘッダの実態(2)

---

smtp サーバが誰でも使える設定になっていることの弊害に気付き、現在に至る迄様々な対策が取られている。→第三者中継の禁止

- 発信元IPのチェック (Open Relay DataBase 等: 自前smtpサーバが使えなくなる)
- pop before smtp

これらは全て、違法行為或は犯罪行為等を防ぐ為の、メール送信時における匿名性の排除である。

ヘッダ、特に Received: を良く見ると、第三者中継が行われているか否かが分かる

## これまでのまとめ:

---

- spamとは不特定多数に大量送信される詐欺や違法商法の勧誘メール
- spamは発信元情報を偽造していることが多い
- メールが配送される仕組みはユーザ認証の機構がないので発信元情報を簡単に偽造できる

## 私の独断と偏見:

---

□smtpは、現実の郵便物配送の仕組みを参考にして作られているプロトコル  
→現実でもピンクチラシや危険物/爆発物が送られるのだから、今後どのようにプロトコルが改変されても spam がなくなるとは思えない。

政策を待っていても何も変わらない

国境を越えれば法の効力は及ばない

過剰反応すれば相手の思う壺

spam は自衛するしか道がない

所詮はゴミ、ゴミはゴミとして扱えば良い

spam は送信相手に読まれることを目的としている

読まずに捨てる事が出来れば理想に一步近付く

だからこそ、 spam 自動フィルタリングが有効

## spam フィルタリングとは

---

- 受信するメールの中身を見て、spam なのか、通常メール (ham) なのかを判定し、判定した結果に従って振り分けを行う。

### spam 判定の基準は?

- メールのヘッダとボディから、それぞれ特徴を捉える  
→基本はパターンマッチング(文字列比較)

## spam フィルタリングの方法

---

- メールサーバ側で対処する方法
- 個人で対処する方法

## メールサーバ側の対処

---

- (1) Open Relay server からのメールを跳ね除ける

Open Relay server の情報を集めたサービス

ORBS: Open Relay Behaviour-modification System (Open Relay Blocking System)2001.6.6に閉鎖

Open Relay DataBase などの、ORBS 直系の子供を含め、現在多種多様な Open Relay server データ参照サービスが存在する。→ <http://spam.h1r.org/blacklists.html>

- (2) パターンマッチングによる spam 判断

## 個人の対処

---

□パターンマッチングによる spam 判断のみ

Open Relay server からのメールを除外する手立ではない

但し、殆どのプロバイダは多かれ少なかれ Open Relay server 対策を講じている

# spam フィルタリングの歴史(1)

---

□MDA (Mail Delivery Agent, Message Delivery Agent: メール配達機構(筆者は「ローカルメール振り分けソフト」と呼ぶ))による簡易フィルタリング

○→ procmail のパターンマッチング機能

▶ 個々人が各々独自にレシピ(recipe)を書いていた

○→ junkfilter (spam 除去に特化した procmail のレシピ(recipe))

▶ <http://junkfilter.zer0.org/>

▶ 紋切り型の判定メカニズム

▶ 誤認率が高く、すぐに抜け道が見つかってしまう

▶ 現在、作者はメンテナンスを中断(恐らく、もう開発は行われまいだろう?)

□独自プログラムによるフィルタリング

▶ procmail 等の MDA と組み合わせるフィルタ

▶ これ以降、これら spam フィルタリングを行うソフトウェアを、spam フィルタと呼ぶ。

## spam フィルタリングの歴史(2)

---

### □ルールベースのスコアリング

- ▶スコアリングを採用、複数のパターンマッチの組合せで判定
- ▶→ spamassassin の初期バージョン(2.4x迄)
- ▶TMDA(Tagged Message Delivery Agent), vipul's razor...

### □ベイジアン理論によるフィルタリング

- ▶→ spamassassin の後期バージョン(2.50以降)
- ▶bsfilter, bogofilter, SpamProbe, Bayespam.rb, scbayes, spamfilter.el, SpamBayes, ifile...

## spam フィルタはどれが良い?

---

あるもの全部試すのは大変  
誰かやってくれませんか?(笑)

## じゃあ、なんで spamassassin を選んだの？

---

ま、最初に出会ったフィルタだから、ってのはありますが、ハイブリッド型フィルタだからです。

つまり、

- ルールベースのスコアリングが出来ると共に、
- ベイジアンフィルタもスコアリングに加えられる
  - エンドユーザによる工夫の余地がある
  - 検出率はユーザの努力で改善できる
  - ソフトウェア自身のバージョンアップで検出率は更に向上が期待できる

# spamassassin の導入と設定(1)

---

## □(1) パッケージをゲット!

- spamassassin パッケージをゲット!
- fetchmail パッケージをゲット!
- procmail パッケージをゲット!

## □(2) 各種設定ファイルを書く!

○~/procmailrc に以下の記述:

- ▷SHELL=/bin/sh
- ▷DEFAULT=\$ORGMAIL
- ▷SPAM=\$HOME/spam/spam/.
- ▷DOUBT=\$HOME/spam/doubt/.
- ▷# call spamassassin
- ▷:0fw: spamassassin.lock
- ▷\* < 256000
- ▷| spamassassin
- ▷:0H:
- ▷\* X-Spam-Flag: YES
- ▷\* X-Spam-Status:.\*autolearn=spam
- ▷\$SPAM
- ▷:0H:
- ▷\* X-Spam-Flag: YES
- ▷\$DOUBT

## spamassassin の導入と設定(2)

---

○ ~/.fetchmailrc に以下の記述:

```
poll www2.palnet.or.jp proto APOP
```

```
user "matsuda" password "wei_ha_oreno_tamashii_no_sakebi" options fetchall  
mda "/usr/bin/procmail -p -f %F"
```

○以下を実行

```
mkdir $HOME/spam/spam/;mkdir $HOME/spam/doubt/;chmod 600 ~/.fetchmailrc
```

○自分宛にメールを送信して、暫くしたら fetchmail を実行

```
cd ~/.spamassassin;wget -O user_prefs http://tlec.linux.or.jp/docs/user_prefs
```

□(3) spam アーカイブを取り寄せて、学習!

```
mkdir tempspam;cd tempspam;wget http://www.flcl.org/~yoh/spam9xxxx.tar.gz;tar  
xvzf spam9xxxx.tar.gz;sa-learn --spam *.*
```

詳細は <http://tlec.linux.or.jp/docs/spamassassin.html> を参照

本勉強会は設定を詳述することを目的としていません。悪しからず。

## spam フィルタリングの仕組み

---

□spamassassin 自体は、perl のパッケージであり、テキストフィルタプログラムである。

メールを標準入力から読み込み、判定結果に従って、読み込んだメールを編集し、標準出力へ出力する。

標準入力/標準出力/パイプ/リダイレクトを知らない人は、この機会に勉強しましょう。

これらは「データをストリーム(流れ)の形でプログラムやデバイスに渡す仕組み」。

ファイルとは「ストリームの形で得られるデータを貯め込んだ器」。

流れとは?

♪あ～川の流れのように～

データがある場所からある場所へ転送される。

その有り様を比喻している。

# 因みに、テキストエディタによるテキストファイルのセーブも、内部的にはストリーム。

## spamassassin を使いこなすには

---

デフォルトインストール時点での spamassassin は、検出精度があまり良くない  
デフォルトの user\_prefs は殆ど役に立たない。

そこでまっだは、日本語圏ユーザに特化した、過激な(?)フィルタリングルールを追加した。

[http://tlec.linux.or.jp/docs/user\\_prefs](http://tlec.linux.or.jp/docs/user_prefs)

## これまでまつだが user\_prefs でやってきたこと

---

1. junkfilter を使っていた頃に ~/.procmailrc に追加していたルールの移植  
X-Mailer 等
2. 日本語メッセージの認識  
鵜飼氏作成のルールを拝借
3. 新たな spam に出会う度にルール追加  
何人かのアドバイスやルールを取り込む
4. ベイジアンスコアリングの調整

## user\_prefs の記述方法

---

man Mail::SpamAssassin::Conf を参照

このマニュアルは未だ日本語訳がないので、誰か訳して下さい

記述例

```
header REPLY_TO_REMOVE Reply-To =~ /remove\@/ ←(1)
```

```
describe REPLY_TO_REMOVE Reply-To set to remove@... ←(2)
```

```
score REPLY_TO_REMOVE 2.0 ←(3)
```

(1) ルールの記述 (PRIVILEGED SETTINGS: 特殊な設定: spamd からは利用できない)

(2) ルールの説明 → 解析結果の単文説明に表示される

(3) ルールに対する配点

この他にも様々な設定、例えば WHITELIST や BLACKLIST も設定できるが、本勉強会では割愛する。

# ルールの種類(1)

---

沢山あるが、これだけ覚えておけばおおよそ事足りる  
詳細はマニュアル参照

## □header

- ▶ヘッダ文字列のマッチング、或は有無
- ▶デコード処理前の生のヘッダは扱えないことに注意

## □body

- ▶デコード処理後の本文のマッチング
- ▶base64 或は Quoted-Printable 等のエンコードに留まらず、html もプレインテキスト化される

## □rawbody

- ▶デコード処理前の本文のマッチング
- ▶htmlコメント等はこれでマッチングさせる
- ▶なお、body / rawbody の実際の文字列の取り扱いには時折バグに悩まされることがあるので注意が必要

## ルールの種類(2)

---

### □full

- ▶生のメール全体に対するマッチング
- ▶本文中のMIME宣言文やバウンダリの指定はこれではできない
- ▶行頭/行末という概念がなくなる(メール全体が1レコードとして扱われる)ので注意→正規表現の^ (キャレット: 行頭指定)と\$ (ドルー: 行末指定)が使えない

### □uri

- ▶メール全体に対する URI マッチング

### □meta

- ▶条件文の複合論理指定

## バウンダリとは

---

マルチパートバウンダリ文字列(multipart boundary)

boundaryとは「境界」の意味

メールに様々なファイルを添付する、マルチパートメールの肝となる、区切り文字列行

世の中の多くのメーラは、ヘッダやこれの存在を隠蔽して、ユーザに見せる工夫がされている

一部のウィルスメールやspamは、ここに特徴を見出すことが出来る

## spam 判別のコツ

---

□ヘッダと本文の両方から、特徴的な文字列を探す  
英語圏spamの特徴の大多数は、既にspamassassinに組み込まれている  
今、それらを列挙する必要はないが、敢えて過去を勉強する為に幾つか取り上げる  
(forge: でっち上げる、偽装する)

## spamassassin の日本語対応(1)

---

これ自身は日本語をうまく取り扱わない

但し、ヘッダでは **7bit JIS** にデコードする処理が行われる

→**7bit JIS** としてエスケープシーケンスを除外して扱えば良い

▷漢字IN: ^[ \$ B

▷漢字OUT: ^[ ( B

▷^[ B \$ ascii文字 ^[ B (

```
$ echo 超過激|nkf -j|awk '{gsub(/\x1B[$(]B/, "");print}'
```

```
D62a7c
```

↓

```
body CHOUKAGEKI /D62a7c/
```

```
$ echo もろ見せ|nkf -j|awk '{gsub(/\x1B[$(]B/, "");print}'
```

```
$b$m8+$;
```

↓

```
body MOROMISE ^$b\m8\+\$;/
```

▷perlの正規表現中、特殊文字はすべからく\でエスケープする

## spamassassin の日本語対応(2)

---

ヘッダはISO-2022-JP(7bit JIS)なのに本文がシフトJIS(MS漢字)な日本語spamがたまにある

例: 一時期有名になったロリムトーなど  
→直接シフトJIS文字列を正規表現に指定する

```
$ echo -n 送信者|nkf -s|od -toC  
0000000 221 227 220 115 216 322
```

```
# special thanks to: R.Takashi ISHIOKA-san! 2003/07/16  
body SJIS_SOSHINSHA ^221\227\220M\216\322/
```

```
$ echo -n 最新流出|nkf -s|od -txC  
0000000 8d c5 90 56 97 ac 8f 6f
```

```
body SJIS_SAISHINRYUSHUTSU ^x8d\xc5\x90\x56\x97\xac\x8f\x6f/
```

## user\_prefs のルール追加方法

---

- (1) spamの特徴を見付ける
- (2) 見付けた特徴を正規表現で書く
- (3) ルールが正しく機能するか、spamを読み込ませて実験する

```
$ spamassassin -d < target_spam|spamassassin -t -D 2>&1|v
```

▷ spamassassin は標準入力からしか入力を受け付けない

## spamから特徴を見つけるには

---

□ 先ず、ヘッダと本文を透過的に見ることのできるメイラが必要

▷ mew 或は wonderlust 等

▷ 或は、MH フォルダ形式等で1メール1ファイルで閲覧できる環境を用意する。

□ spamから特徴的な文字列を探すには、最低でも2通以上同じspamが来る必要がある

▷ 1通だけではどのような特徴なのかがわからない

▷ 2通以上来たら、それらspamを比較する

▷ エディタで比較するのが一番良いと思う

□ できるだけヘッダを優先

▷ 本文よりもヘッダの方が、特徴を特定し易いし、安全

□ 本文から特徴を抽出する場合は慎重に

▷ 少ない単語だけでルールを作ると、通常メール(ham)と誤認識し易い

▷ 友人がメール中に spam を引用したらどうするか?

▷ 必ず複数のフレーズを採用し、META で掛け算の併せ技

# 上手なベイジアンフィルタの使い方

---

## □(1) 全てをベイジアンフィルタに頼らない

▶フィルタリングツールは万能ではない

○ベイジアンフィルタをすり抜ける spam はたまに現れる

▶→その都度ルール再検討

○無料メールサービス等に来る広告メールは、厳密には spam ではない

▶→比較的容易に特定可能、よって別のフィルタ(MDA)で除外

○ウィルスメールは巨大で、処理に時間がかかることが多い

▶→種類は少ないのでこれも比較的容易に特定可能、望ましくは別のフィルタ(MDA)で除外

## □(2) 定期的に検出結果をチェックする

▶望ましくは毎日、最低でも一週間単位で、振り分けた結果が正しいかを検証する

▶もし、通常メールを誤って spam と認識したら、内容次第では大きな損害を発生するかも

▶疑わしいメールを一つずつ確認後、sa-learn で手動学習させることも必要

## spamの今後

---

ベイジアンフィルタや第三者中継を禁じるフィルタリング技術が進めば、案外減少傾向に転じるかも？

既に私の環境ではspamは減少傾向にある

但し、ウィルスメールと結託するspamも現われているので油断は禁物

リンク名 スпам業者とクラッカーが結託：追跡不可能なサイトやスパム送信ウイルス開発

URI: <http://www.hotwired.co.jp/news/news/technology/story/20031014301.html>

世の中に騙される人が存在する限り、なくなることはないだろう

## まつだの独断と偏見

---

spamとウィルスメールは「騙す」点において共通性がある

騙されないことが基本中の基本

受信したメールに心当たりがなければ、必ず生の状態で見られるようにすること(MIME展開を行わない)

→メールを生の状態で見れる環境が望ましい、つうか必要でしょ。

## 何故spammerはspamを出すのか?

---

spamを出すこと、それは犯罪行為、違法行為、或は迷惑行為

しかし、それはspammerにとって利益を産む行為

spamによって騙される人が居るからこそ、彼らは生きて行けている

騙される人はspammerの生活の糧となっている

これは、ネットワークが富をも産み出し得る場となったことの現れ

電子ネットワークも現実社会と同等の機能を提供している

## spamから自己を守るには

---

spamに騙されないことは最低限大事なこと

その次に、spamからどう身を守るのか

人間は自己の幸福の追求の為に生きている

この幸福追求の目的を阻むものが、他人による犯罪行為、迷惑行為

自己の幸福追求を確保する為にどうしたら良いのか

自分の精神的安定を確保することは勿論の事、相手の欲求を充足しないことも考えなければならない

「相手の思う壺」になるような行為は逆効果

spamは受信者に読まれることを目的として作成されている

→spam送信相手に反応する行為は逆効果

反応する行為は、読んでいることを表わしている→spammerの目的が達成された

我々は法の執行者ではない。反応する相手は然るべき機関

spammerには反応しないことが第一

次に相手がspam送信を諦める環境を作ること

# 前野年紀先生に対する反論 - <http://spam.qmail.jp/user-defence.html>

---

## □ブラックリストへ通告しよう

▶spam が送られてきたら、自前のブラックリストに登録するほかに どこかの RBL へ通知しましょう。spam を送ってくるホストが動的割りあての IP アドレスなどだったら、ISP に連絡するのがいいようです。

▶razor などの協調的 spam 対策ネットワークに登録するのもいいでしょう。

→毎日舞い込んでくるspamの一つ一つにいちいちこんなことやってられない

あたさそんな暇人じゃありません

→あたさそんな高度な技術を持っておりません(メール解析、英文作成のコストがバカにならない)

そんな高度な技術を持ったら他の事に使います

## □メールアドレスをさらさないこと

▶最近の spam はアドレス収集ロボットを使って集めた実在アドレス宛に送られてきます。アドレス収集に利用されそうな場所を書きこんだり、メールしたりするときには専用のアドレスを使いましょう。整理するときにも便利です。

→オープンな場にて連絡を取り合うようになった人とのコミュニケーションはどうする？

相手に『こちらのアドレスを使ってください』と言うのか？

→捨てアドレスとは「捨てても良いコミュニケーション」を表してはいないか？

## 前野年紀先生の提案『お馴染さん』方式 - <http://spam.qmail.jp/onazimi/>

---

リストにない相手に『一時エラーを返事』するだけで、多数の spam を受け取り拒否できます。spam (おそらく open proxy 経由) は再送してこないことが多いからです。つまり、相手が簡単に送信をあきらめてくれることを利用しています。

### まつだの考え

多分、低コストで大きな効果が得られる、すばらしい方法

もしもこれが完全なら、エンドユーザが行うフィルタリングは不要になるかも？

→否、英語圏 spam の大多数を排除できても、根本的にはなくならないだろう

anonymous provider の存在、web mail service の存在は spam の温床になりがちである

→抜け道は存在するし、破られるだろう