メーリングリストにおけるスパムメール防止方式の研究

高　橋　健　一　　財団法人九州先端科学技術研究所情報ｾｷｭﾘﾃｨ研究室研究員

1 はじめに

グループ内での情報交換ツールとしてメーリングリストが利用されている。しかし、メーリングリストでも多量のスパムメールの発生が問題になっている。Symantec[1]の報告によると電子メール全体の 75%以上がスパムメールと言われている。また、特定のグループ内での情報交換を図るためにメーリングリストが利用されている。このようなメーリングリストでもスパムメールが発生しているケースが存在する。スパムメールを一時的になくすためにはアドレスを変更することが効果的である。しかし、既存のメーリングリストシステムではメーリングリストアドレスがメーリングリストメンバで共有される。このため、スパムメールの増加などの理由でメーリングリストアドレスの変更が必要になった場合でも容易にそのアドレスを変更することが難しい。そこで、メーリングリストのそれぞれのメンバごとに個別アドレスを発行する方法を研究開発する。メンバごとに異なる個別メールアドレスを割り当てるため、スパムメール増加の原因となったメンバを特定することができる。また、その原因となったメンバの個別アドレスの無効化・再発行によりスパムメールの増加を防ぐことができる。

2 関連研究

正当なメールとスパムメールに現れる特徴の違いによって、スパムメールをフィルタリングし遮断することが一般的に行われている[1]。しかし、これらのフィルタリング手法には false positive/negative の問題がある。また、投稿者や投稿元ドメインを限定することでスパムメールを排除することができるが、外出先などから一時的に gmail や yahoo メール、携帯電話などからメーリングリストに投稿することができなくなり、ユーザの利便性を損ねる。

SIDF (Sender ID Framework) [2]や DKIM (Domain Key Identified Mail) [3]ではメール送信元の認証を行うことでスパムメールに対処する。しかし、多数のメールサーバの協力やクライアントマシンに特別なソフトウェアライブラリのインストールが必要などの問題がある。また、メーリングリストでの利用やメールの転送を不得手とするという欠点がある[4]。

スパムメールは同一の送信者から大量に送信されることが多い。このため、メール送信時に Challenge & Reponse 型の負荷をかけることで大量のスパムメール送信を防止するための提案が行われている[5]。また、[6]ではメール送信者に一定量の課金を行うことで、スパムメール送信者に金銭的な負担を負わせることが提案されている。しかし、これらも SIDF や DKIM と同様の問題を持つ。

Privango[7]ではメールの受信条件を暗号化しメールアドレスに埋め込むことで、自動的に受信条件に合わないメールを排除することを提案している。[8]ではメールサーバが自動的に alias アドレスを生成し、スパムメールが発生したときにその alias アドレスを削除することでスパムメールを排除することを提案している。しかし、複雑なメールアドレスとなり覚えられないといった問題や受信条件を埋め込むための操作が必要といった問題がある。

そこで、ユーザに特別なソフトウェアのインストールを要求せず、既存のメールクライアントで利用可能で、かつ、ユーザに既存のメーリングリストと同様の操作しか要求せず、既存メーリングリストと同程度の利便性を備えたシステムを開発する。

3 システムの概要

スパムメールの発生は基本的にメールアドレスがスパマーに漏洩することで起こる。そこで、メーリングリストのそれぞれのメンバごとに個別の投稿先アドレスを発行することで、スパマーに漏洩したアドレスを無効化し、スパムメールの増加を抑えるシステム、Preventive Spam Mailing List (PSML) を開発した。

PSML ではメーリングリストのメンバにそれぞれ異なるメーリングリスト投稿用の個別アドレスが発行す

る。メーリングリストへの投稿はそれぞれに発行された個別アドレスにメールを送信することで行う。概要を図１に示す。各メンバが異なる個別アドレスをメーリングリストの投稿に利用しているため、メーリングリストの個別アドレスを漏洩させたユーザ、すなわちスパムメール増加の原因となったユーザを特定することができる。また、スパムメール発生の原因となっている個別アドレスを無効化し、異なる個別アドレスを発行することで、個別アドレスを漏洩させたユーザ以外に影響を与えることなく、メーリングリストへのスパムメールの増加を防ぐことができる。

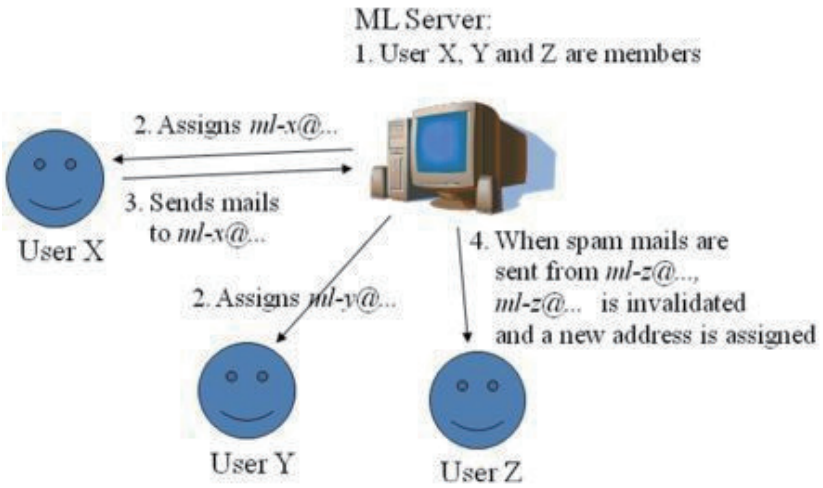

図１．システムの概要図

3-1 PSML の実装

PSML の実装の概要を図２に示す。

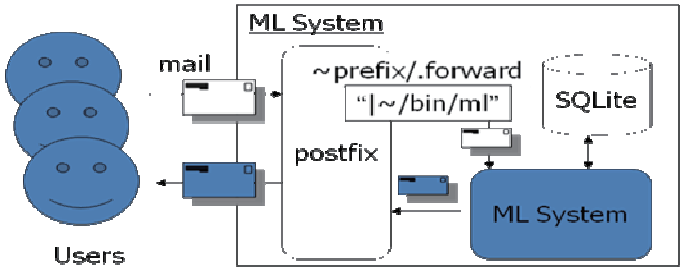

図２．実装システムの概要

システムの開発は perl5.8.8 を実装し、メールアドレス管理などには SQLite を利用した。また、実際のメール配送には既存の SMTP サーバ（postfix）を利用することとした。個別アドレスには prefix″-″ suffix という形式を利用した。prefix はメーリングリストアドレスとして固定し、suffix にメンバ固有の識別子を利用する。メーリングリストの作成時にはランダムな文字列を suffix として生成する。prefix-に届いたメールが。forward の設定により本メーリングリストシステムに標準入力として渡される。システムに渡されたメールには前節までに述べた処理が施され、メンバへ配信するメールを postfix に渡すことでメールの送信を行う。また、Subject ヘッダが下表に示すキーワードであった場合、個別アドレス変更要求やスパムメール報告などであることを判断する。PSML で利用可能なキーワード一覧を表１に示す。逆にこれらのキーワードが用いられないメールに対してはメーリングリストへの投稿として扱うこととした。

表１．コマンド一覧

Subject	説明
spamreport	引用元のメールがスパムメールであることを報告する
changesuffix *arg*	個別アドレスの *suffix* を *arg* に変更する
Summary *args*	*args* で指定した番号の投稿履歴を取り出す
bye	メーリングリストから退会する

3-2 PSML の性能評価

PSML の性能を実験により検証した。実験は CestOS 5.2、Intel Pentium D 2.8GHz、1GB メモリの計算機上で行った。メーリングリスト作成時には suffix として 8 文字のランダムな文字列を利用することとした。実験結果を図 3 に示す。実験の結果、20 人のメンバが参加するメーリングリストを作成した場合で 0.17 秒、50 人で 0.39 秒、100 人で 0.79 秒で処理できた。すなわち、メーリングリスト作成時、1 秒間に 100 人以上のメンバを処理可能であった。個別アドレスの変更は 46.1 回/秒を処理可能であった。また、1k バイトの本文を持つメールに対して、20 人が参加するメーリングリストでは 44.1 通/秒、50 人では 40.0 通/秒、100 人では 35.2 通/秒のメールを処理可能であった。このように百人規模以上のメンバからなるメーリングリストに対して十分な性能を持つシステムを実現できた。

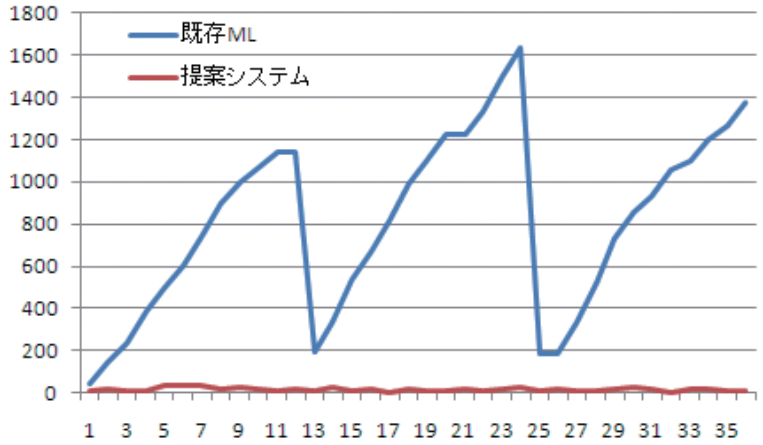

図 3．実験結果（縦軸：スパムメール数、横軸：月、メンバ数 50 人、既存 ML は年 1 回のアドレス変更、提案システムはスパムメール報告率 5%、3 通のスパムメール報告で個別アドレス変更）

また、メーリングリストのメンバ数の違いや、スパムメール報告率の違い、個別アドレス無効条件の違いなどにより、本システムが既存メーリングリストシステムと比べ、どれくらいのスパムメール削減効果があるかをシミュレーション実験した。実験の結果、どのような設定で実験を行ったとしても、本システムではスパムメール削減効果が確かめられた。また、メンバ数が多ければ多いほど、スパムメール報告率が高ければ高いほど、個別アドレス無効条件が緩ければ緩いほど、本システムのスパムメール削減効果が高くなることが分かった。

4 PSML の利用

PSML を http://itslab.csce.kyushu-u.ac.jp/~sakai/psml/index_ja.html で Apache ライセンスにてフリーウェアとして公開している。以下、公開しているシステムの利用方法を示す。

4-1 動作環境

- 動作 OS ： OS 依存なし
- メールサーバ：UserID-* のエイリアスが使えるもの（Postfix 推奨）
- データベースシステム：Perl の DBI に対応した物。

（データベースシステムが利用できない場合、ファイルをデータベース代りに使う SQLite で代用できます。）

- Perl 5.8 以降

Perl で必要なモジュールを事前にインストールしてください。インストールは全て cpan から行えます。

- HTML::Template
- Digest::SHA1
- Digest::MD5
- Jcode
- Date::Calc
- Net::DNS

- DBI
- DBD::*** (DBD::mysql or DBD::Pg or DBD::SQLite)

インストール例 (PostgreSQL の場合)

```
% su - root
# cpan install DBI
# cpan install DBD::Pg
```

4-2 インストール方法

インストールは以下の手順で行う。

- Postfix の場合、ユーザ ID-エイリアス@ドメイン(以下 example.co.jp をドメインとして説明) の形式を取り扱えるように設定を変更します。

(main.cf に以下を追記)

```
recipient_delimiter = -
```

- はじめに配布ファイル一式をサーバーに「アスキーモード」で FTP 転送してください。(文字コードは変換しないでください。EUC-JP でしか正しく動作しません。)
- check.pl のパーミッションを 755 にして実行してみてください。ここで必要な Perl 環境がそろっているかをチェックします。エラーが出た場合、2 節を参考にモジュールをインストールしてください。
- データベースシステム上に接続ユーザ(USERNAME、 PASSWORD)の作成、データベースの作成(DBNAME)をします。

(PostgreSQL)

```
%su postgres
%createuser -h HOSTNAME -P USERNAME
%createdb -h HOSTNAME -E EUC-JP -O USERNAME DBNAME
```

(MySQL)

```
%mysql -h HOSTNAME -uroot -p
mysql> create database DBNAME default character set ujis;
mysql> grant all on DBNAME.* to USERNAME@'%' identified by 'PASSWORD';
```

(SQLite)

```
%cp ./init_db/mldata.dbs
```

- conf-example.pm を conf.pm にコピーし、サーバーの環境に合わせて修正してください。各項目の表記方法等は、コピーされた conf.pm に説明が書いてあります。
- Database config section のセクションを上記で用意したデータベース環境に合わせた値を設定します。

(PostgreSQL)

- $DB_MODULE="Pg";
- $DB_USER $DB_PASS $DB_HOST はそれぞれ、接続ユーザ名(USERNAME)、接続パスワード(PASSWORD)、DB サーバのホスト名(HOSTNAME)
- $DB_NAME にデータベース名(DBNAME)をセット。

(MySQL)

- $DB_MODULE="mysql";
- $DB_USER $DB_PASS $DB_HOST はそれぞれ、接続ユーザ名(USERNAME)、接続パスワード(PASSWORD)、DB サーバのホスト名(HOSTNAME)
- $DB_NAME にデータベース名(DBNAME)をセット。

(SQLite)

- $DB_MODULE="SQLite";
- $DB_USER $DB_PASS $DB_HOST を全""にセット。
- $DB_NAME="./mldata.dbs";

- パーミッションの設定を行います。(tarball をサーバ上で直接展開した場合、パーミッションは自動的に適切に設定されます。)

ml : 755
conf.pm : 600
その他のファイル : 644
ディレクトリ : 755

- テーブルの準備(初期化)をします。
(PostgreSQL)
init_db/postgres.sql を phpPgAdmin 等で実行します。
- ログインして、データベースを選択する。
- SQL をクリックして「スクリプトをアップロード」で init_db/postgres.sql をアップロードして実行。シェルが使える場合は、上記で作成した接続ユーザと同じ値を用いて、次のコマンドを実行することもできます。

```
% psql -h HOSTNAME -U USERNAME DBNAME < init_db/postgres.sql
```

(MySQL) InnoDB を使用するため MySQL4 以上のみ
init_db/mysql.sql を phpMySQLAdmin 等で実行します。
- ログインして、データベースを選択する。
- SQL をクリックして「スクリプトをアップロード」で init_db/mysql,sql をアップロードして実行。 シェルが使える場合は、上記で作成した接続ユーザと同じ値を用いて、次のコマンドを実行することもできます。

```
% mysql -h HOSTNAME -u USERNAME -p DBNAME < init_db/mysql.sql
```

(SQLite)
- 初期化済みのため不要です。再作成したい場合、シェルから次のコマンドを実行します。

```
% sqlite3 ./mldata.dbs < init_db/sqlite.sql
```

- 初期設定では(init_db.sql で初期化したデータ)、管理者ユーザとして以下が追加される。
UserID : root
Password : root
- サーバ上に OS アカウントとして PSML を動作させるユーザ(MLAccount)を作成します。はじめに、ここで作成したユーザに Postfix から(通常の)メールが届くことを確認します。
- Postfix の転送設定ファイル(~MLAccount/.forward)を作成し以下の 1 行を記述します。ダブルクォート記号も必要です。
"|/(PSML をインストールしたディレクトリ)/ml "

以上で PSML を使用できる環境が整う。

4-3 利用方法（基本）

（１）ML の作成と購読者の作成

はじめに、ML と購読者の作成を行う必要があります。ML 名はインストール中に OS に作成したアカウント(MLAccount)です。

```
$ ./ml
```

コンソールにログインします。

```
PSML > login
Login User : root
Password : root
OK : 認証
```

次にメーリングリストを追加します。

```
PSML@root % addml
ML Name : MLAccount
Description : Test Mailing List
OK : ML 追加
```

続いて、メーリングリストのメンバを作成します。

```
PSML@root % adduser
User Name : Akihiro Sakai
Mail Address : user1@user1domain。com (ユーザの実際のメールアドレスです)
Identity (Suffix) : user1
Identity (Password) : MyPassowrd
Privilege (1:Admin,  2:Oper, 3:User) : 3
OK : ユーザ追加
```

続いて、登録したメンバを購読者リストに加えます。

PSML@root % addsubscribe
ML Name : MLAccount
Target User Name : Akihiro Sakai
OK : ML 購読追加

ここまで終わったら exit コマンドを実行し、変更を反映させます。

PSML@root % exit

（２）メールの投稿

ユーザによるメーリングリストへの投稿は、MLAccount-Suffix@Domain 宛にメールを送信することで行います。 Suffix はメンバ作成中に指定した Suffix 値、Domain は ML サーバのドメインを指定します。

（例）

From: user1@user1domain.com
To: MLAccount-user1@example.co.jp
Subject: ○○の件

各メンバには以下のようなメールが配信されます。

From: user1@user1domain.com
To: MLAccount@example.co.jp
Reply-To: MLAccount-userXX@example.co.jp (userXX は受信者自身の Suffix 値になります)
Subject: [MLAccount:00001] ○○の

受信者はメーラの通常の返信操作により、メーリングリストに返信することができます。

（３）スパムが届いた場合のレポート

作成した ML にスパムメールが届いた場合､そのメールがスパムメールであることをシステムに報告できます。報告の手順は、メーラで返信操作を行い、その際、件名を spamreport と変更してください。

（例）

From: user1@user1domain.com
To: MLAccount-userXX@example.co.jp
Subject: spamreport

このメールはシステム側で処理され、配信はされません。

（４）ユーザによる投稿先アドレス（Suffix）の変更

ユーザの投稿先アドレスがスパムメール送信に一定回数(conf.pm の設定による)以上利用された場合、システムからユーザに対し、アドレス変更を要求するメールが届きます。ユーザは、件名を changesuffix としたメールを送ることにより、自身の Suffix を変更することができます。

4-4 その他

（１）管理者ユーザの復活方法

万一、管理者のユーザ編集で、管理者権限をはずしてしまった場合、 PSML を管理できるユーザがいなくなってしまう場合があります。その場合は、次のようにして復活できます。

(PostgreSQL)

psql コマンド又は、phpPgAdmin で次のクエリを実行。

> update ml_user set user_privilege=2147483647 where user_name='(管理者とするユーザ名)';

(他の DBMS)

DB にアクセスして上と同じクエリを実行してください。

（２）他の DBMS の利用について

運用実績等から、MySQL か PostgreSQL を推奨しています。しかし、PSML は他のデータベースでも利用できるように設計しています。現在の動作確認状況は次の通りです。

PostgreSQL 確認済み
MySQL(InnoDB) 確認済み
SQLite 確認済み(ただし、外部参照制約の検査を行わない)
Oracle 未確認

（３）プラグインについて

Conf.pm で有効無効を切り替えられます。各プラグインはコメントアウトすることによって無効になります。

[標準添付のプラグイン(必須)]
- Login - ログインログアウトの処理をします。

[標準添付のプラグイン(任意)]
- Distribute - メールの配信を行うモジュールです。
- UserOper - ユーザ追加削除等の管理を行います。
- MLOper - ML の追加削除等の管理を行います。
- SpamReport - スパム報告の受け付けを行います。
- Backup - メールのバックアップを行い、get により投稿履歴の取り出しを可能にします。

（4）コマンドラインリファレンス

起動方法 : ./ml をシェルから起動します。

起動するとコマンドの入力を受け付けます。各コマンドは対話形式(引数は不要)になっているので以下のコマンドの中から 1 つを入力します。

login: コンソールにログインする。一番最初に行う必要がある。

adduser: ユーザを追加する。

deluser: ユーザを削除する。削除する前に、delsubscribe コマンドを使って削除対象のユーザを全ての ML から購読を解除する必要があります。

changesuffix: 指定したユーザの投稿先アドレスを変更する。

passwd: 指定したユーザのコンソールログインパスワードを変更する。

addml: ML を追加する。コマンド実行後、OS 上に追加する ML の名前と同じユーザを作成してください。但し、追加するユーザのホームディレクトリと UID の値は、3 節で作成したユーザと同じ値になるように作る必要があります。

delml: ML を削除する。ただし、OS のアカウントはこのコマンドでは削除されません。

addsubscribe: ML 購読者リストにユーザを追加する。

delsubscribe: ML 購読者リストからユーザを削除する。

listsubscribe: ML 購読者リストを表示する。

（5）メールの Subject タグによるアクションリファレンス

メールの Subject として以下の文字が指定されていると通常の配信動作のかわりにアクションが実行されます。各アクションは半角スペース区切りで引数を与えられます。

changesuffix $1: 自分の投稿先アドレスを MLName-$1@domain に変更する。

使用例:

To: MLName-MyCurrentSuffix@example.co.jp

Suject: changesuffix MyNewSuffix

spamreport: スパムの報告を行う。通常のメーラから元のスパムメールの返信の形で送信し(つまり、In-Reply-To ヘッダが必要)、件名のみを spamreport に変更する。

get $1 [$2・・・]: 投稿履歴の取り出しを行う。引数には取り出すメールの通し番号のリストを指定する。

使用例: 1～3 番の番号が付いたメールを取得

To: MLName-MySuffix@example.co.jp

Suject: get 1 2 3

unsubscribe: 現在の ML の購読を解除する。

5 おわりに

メーリングリストのメンバごとに異なる個別アドレスを発行し、スパムメール発生の原因となった個別アドレスを無効化・再発行することでスパムメールを削減するための仕組みを研究開発した。本提案システムでは、メンバに特別なソフトウェアのインストールを要求せず、既存のメールクライアントで利用可能な仕組みを実現している。また、既存のメーリングリストと同様の操作でメーリングリストへの投稿や返信を可能とする。シミュレーション実験による結果、既存のメーリングリストシステムに比べて高いスパムメール削減効果が期待できることが確認できた。本提案システムは Preventive Spam Mailing List (PSML) として http://itslab.csce.kyushu-u.ac.jp/~sakai/psml/index_ja.html にてフリーウェアとして公開している。

【参考文献】

[1] 田端 利宏, SPAM メールフィルタリング: ベイジアンフィルタの解説, 情報の科学と技術, Vol. 56, No. 10, pp. 464-468, 2006.

[2] The Sender ID Framework (SIDF), http://www. microsoft. com/mscorp/safety/technologies/senderid/default. mspx.

[3] Domain Key Identified Mail (DKIM), http://www. dkim. org/.

[4] G. Lawton, E-Mail Authentication Is Here, but Has It Arrived Yet?, IEEE Computer, Vol. 38, No. 11, pp. 17-19, 2005.

[5] R. Roman, J. Zhou, J. Lopez, Protection against Spam Using Pre-Challenges, Proc. of 2005 IFIP International Information Security Conference, pp. 281-293, 2005.

[6] R. E. Kraut, S. Sunder, R. Telang, J. Morris, Pricing Electronic Mail to Solve the Problem of Spam, Human-Computer Interaction, Vol. 20, No. 1\&2, pp. 195-223, 2005.

[7] K. Takahashi, T. Abe, M. Kawashima, Stopping Junk Email by Using Conditional ID Technology: privango, NTT Technical Review, Vol. 3, No. 3, pp. 52-56, 2005.

[8] M. Kawashima, T. Abe, S. Minamoto, T. Nakagawa, Cryptographic alias e-mail addresses for privacy enforcement in business outsourcing, Prof. of the 2005 workshop on Digital identity management, pp. 46-53, 2005.

〈発 表 資 料〉

題　　名	掲載誌・学会名等	発表年月
個別アドレス発行によるメーリングリストへのスパムメール削減方式の提案と評価	情報処理学会論文誌, Vol. 50, No. 9	平成 21 年 9 月
Spam Mail Blocking in Mailing Lists	Multimedia, Kazuki Nishi (ed.), IN-TECH	Feb. 2010