メール配送系における多様な迷惑メール対策の統合管理手法 *

佐々木琢磨 (学籍番号　200821654)
研究指導教員：阪口哲男

1　はじめに

インターネットにおける迷惑メールは, ユーザへの被害だけではなく, 同時多量送信によるメールサーバへの負荷も問題であり, メール配送系 [1] への迷惑メール対策の導入は必須である. 一方で, 実際に利用されているメールサーバソフトウェアは様々であり, 迷惑メール対策ソフトウェアは, 同じ原理や方式のものでも, メールサーバソフトウェアごとに個別にある. また, 様々なソフトウェアを組み合わせるために, 設定ファイルが散在し, さらに個々の設定ファイルが固有の記述形式であるなど, 管理者に負担がかかるという問題がある. その問題の解決のため, 本研究では, メール配送系と迷惑メール対策の設定を統合して管理する手法とその実現方式を提案する.

2　迷惑メール対策の統合管理

複数の迷惑メール対策をメール配送系で併用する場合, メールサーバソフトウェアと, 迷惑メール対策のために導入したフィルタソフトウェアの両方を設定・管理する必要がある. しかし, メールサーバソフトウェアは迷惑メール対策設定以外にも設定事項はあり, 場合によっては基本的なメールサーバの動作設定と迷惑メール対策設定が同一の設定ファイル内に混在することもある. また, フィルタソフトウェアを複数導入すると, 各フィルタソフトウェアの設定・管理も必要となる.

その結果, 迷惑メール対策を含めたメール配送系全体の設定内容を確認する場合, 散在した各設定ファイルを辿っていく必要があり, 管理者の手間となる. そこで, 迷惑メール対策の統合管理が必要と考えられる. 迷惑メール対策の統合管理とは, メールサーバソフトウェアの設定と迷惑メール対策設定の記述や記法が統合されたメール配送系の管理方式である. 統合管理に関する先行研究としては, MTA ソフトウェアで利用される外部ソフトウェアの規格統一のための API である milter API および milter manager[2] や, 電子メールのフィルタリング動作を記述できるプログラミング言語 Sieve[3] がある.

しかし, 前者は MTA の迷惑メール対策ソフトウェアの導入や設定手順の統合, 後者は MDA におけるメール振り分け条件記述の統一にとどまっている. そこで, 本研究ではメール配送系構成要素全体を対象とし, 迷惑メール対策の統合管理のための記述言語を提案する.

3　メール配送系の統合管理手法

3.1　統合管理言語と解釈系

メール配送系における迷惑メール対策の設定ファイルの散在およびソフトウェア固有の設定記法による管理の手間を低減すべく, 必要な設定項目の記述形式を統一したメール配送系統合管理言語を定義し, その記述から各ソフトウェアの設定ファイルおよびメール配送系構成要素の設定ファイルを生成する解釈系を開発した. これらの言語およびその解釈系を適用したメール配送系は, 全体の迷惑メール対策の構成を確認する場合に散在した設定ファイルを辿る必要がなくなり, 管理の負担が低減される.

本研究で開発したメール配送系統合管理言語を amaretto と呼ぶ. 図 1 に amaretto の記述例を示す. 図 1 は, MTA に postfix を用い, 迷惑メール対策として postgrey を導入して Greylisting を行う場合の amaretto の記述例である. amaretto では, `use` ブロックごとに管理対象ソフトウェアの設定項目を記述するようにし, 記法および設定ファイルを統合した.

次に amaretto の解釈系について述べる. 解釈系は, 多様に存在する迷惑メール対策に対して柔軟に対応させる必要がある. そこで, 解釈系は, amaretto 記述を

* “An integrated method for management of email delivery systems with various anti-spam tools ” by Takuma SASAKI

```
1  require "amaretto";
2    set "greyport" "60000";
3
4    use "postgrey" as GREYLISTING{
5      delay "300";
6      postgrey.port "${greyport}";
7    };
8
9    use "postfix" as MTA{
10     service "${greyport}";
11   };
```

図 1　amaretto の記述例（迷惑メール対策設定部分を抜粋）

各ソフトウェアの設定ファイルのテンプレートおよびルール記述に適用することにより各設定ファイルを生成させる方式とした. ルール記述とは, amaretto 記述をテンプレートに適用する際の制約, 時間の単位変換などの計算命令を記述したものである. テンプレートとルール記述の例を図 2, 3 に示す.

```
1  delay %{delay}m
2  autowhite %{max-age}d
```

図 2　milter-greylist 用テンプレート（抜粋）

```
1  using GREYLISTING
2  if delay :exists
3    then value.to_i / 60
```

図 3　milter-greylist のルール記述の例

3.2　解釈系の実現

解釈系は前処理として amaretto 記述とルール記述を入力し, ルールを適用した中間ファイルを出力するルール解釈部分と, その中間ファイルとテンプレートを入力して該当部分を置換処理する amaretto 記述解釈部分である.

解釈系の開発は, Java 言語を用いた. parser generator には SableCC[4] を用い, 生成された各クラスを利用して開発した. 入力された amaretto 記述は, ルール記述解釈部分により記述の計算や制約の確認がされ, amaretto 記述解釈部分に渡される. amaretto 記述の処理系も, 適用ルール記述の処理系と同様に SableCC を用いて開発を行った.

4　試験運用と評価

本研究のメール配送系統合管理手法を適用したメールサーバを試験運用し, 迷惑メール対策の設定・更新作業を行うことにより, 本手法における管理の手間を従来の場合と比較して評価を行った.

試験運用は, VMWare 仮想サーバ上に Debian GNU/Linux 5.0 を OS とし, MTA ソフトウェアとして Postfix 2.5, MDA・MRA ソフトウェアとして Cyrus IMAP Server 2.2 を用いた. 導入する迷惑メール対策方式としては, Greylisting, Whitelist, Cyrus IMAP Server がサポートする Sieve 機能を採用し, Greylisting に関しては, postgrey および milter-greylist を採用した.

上記の迷惑メール対策の導入・管理作業をいくつかのパターンで行ったところ, 従来では最大で操作対象ファイルが 5 つで計 500 行程度であったのに対し, 本手法では約 30 行の amaretto 記述のみで作業が終了するなど, 手間の低減が確認された.

5　おわりに

メール配送系統合管理言語 amaretto とその解釈系を用いた本手法は, 迷惑メール対策の設定・更新作業において, 従来よりも管理者の作業の手間が低減されることが示された.

また, 今後新しい迷惑メール対策方式が実現された場合でも, その方式のテンプレートおよびルール記述を用意することで本手法は対応できると考えられるが, その検証については残された課題となっている.

参考文献

[1] David Wood. “インターネット上の電子メール”. 電子メールプロトコル. 佐々木雅之ほか監訳. オライリー・ジャパン, 2002, p. 1-24.

[2] クリアコード. milter manager. http://milter-manager.sourceforge.net/index.html

[3] RFC 5228. Sieve: An Email Filtering Language. http://tools.ietf.org/rfc/rfc5228.txt

[4] SableCC project. SableCC. http://sablecc.org/wiki