

ホワイトリストによる電子メール受信規制の緩和

2012SE185 野村 太郎 2012SE237 鈴木 ひかり

指導教員：後藤 邦夫

1 はじめに

近年では SNS が普及しており、メール離れと言われつつある。しかし、比率では電子メールの利用頻度はかなり高いと言える [4]。それに比例して、メールの送受信問題が発生しており、その中でも迷惑メールの受信規制による問題が起きている。

携帯会社の契約時の標準設定では、電子メールでの会話が一方通行になる。特に、携帯電話からのメールは一般のドメインに送信できるが、その返信を受信できない、または迷惑メールフォルダに分類され受信側がその状態に気づかないことが一方通行になる原因である。もし迷惑メールフォルダに分類された場合は、当然受信拒否を通知するエラーも発生しないため、返信した側はメールが届かなかったことがわからない。さらに、上記のようなエラーメッセージは、利用者が多い3社のうち1社しか出さないため返信を送ったのに返答が来ないといった事態が発生してしまう [1] [2] [3]。

その中でも、携帯電話ではないメールアドレスからの返信には、URL や電話番号が記載されている場合が多く、返信が迷惑メールとして扱われる可能性が大きい。

そこで本研究では、フィルタに入る前に以前メールを送信した宛先からの maillog の有無を確認して、whitelist を用いて必要なメールかを判断し、上記の問題を対処できないか考えた。したがって上記のようなメール送受信問題に対し、少なくともユーザが送信した宛先からのメールは受け取れるべきものとして、送信者がメールを出した宛先のリストを作り、リスト内の宛先からのメールは受け取れるシステムの構築を目的とする。先行研究は調べた結果見当たらず、また本システムを携帯電話会社に導入することで本研究で取り上げている問題が少しでも解決できるのではないかと思い本研究をすることに至った。

プログラムはメール log を読み取りリストに書き込むプログラム (以下プログラム 1)、作成したデータベースを読み取るプログラム (以下プログラム 2) の2つを作成する。上記のプログラムは SMTP サーバに組み込み、メールを受信する際に起動する。2つのプログラムを用いて、メールのログを読み取り、リストに上書きする。リストをデータベース化して、受信されたメールとデータベースの中の table にある宛先・送信者のアドレスが一致すればメールを受信し、アドレスが不一致の場合はエラーコードをサーバに返す。本研究では野村は主に実験環境とプログラム作成、鈴木は主に文章作成とプログラム作成を担当した。

2 本システムの概要

図 1 では、本システムの構造を示しており、図のように本研究では Postfix2 機を用いることで送信用と受信用の SMTP サーバとして実装する。

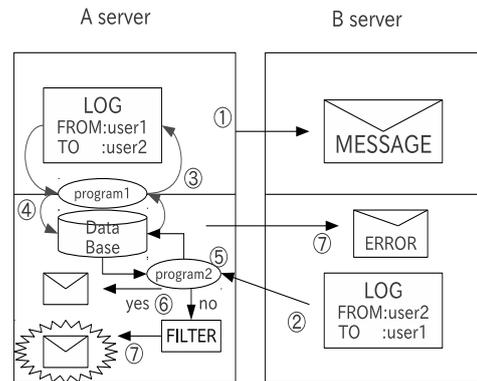


図 1 本システムの概要

一般的にユーザ間でメールを送信した時に、自分側のサーバの送信記録ファイルにメール log が記録される。本研究ではこのメール log を用いてシステムを実現する。自分側のサーバのメール log の中で必要な log をプログラム 1 を用いて抜き出し、リストを作成する。リストを参照し、プログラム 1 でデータベースに接続し table を作成する。

表 1 userlist の表

メールアドレス
usr1@mobile0.exa
usr2@mobile0.exa
usr3@mobile0.exa

表 2 maillog の表

送信者アドレス	宛先アドレス
usr1@mobile0.exa	ousr1@mobile2.exa
usr2@mobile0.exa	ousr2@mobile2.exa
usr3@mobile0.exa	ousr1@mobile3.exa
受信回数	最終受信時刻
2	Jan.03.10.25
4	Jan.02.08.23
3	Jan.03.08.56

本研究では userlist の table と maillog の table の 2 つ

の table を作成した。そしてデータベースはリレーショナルデータベースを用いる。userlist の表と maillog の表があり、userlist の表は送信者のメールアドレスで maillog の表は送信者アドレス、宛先アドレス、受信回数、最終受信時刻で構成されている。userlist の表の送信者のメールアドレスを maillog の表の送信者アドレスにリンクさせている。そのため送信者のメールアドレスより宛先アドレス、受信回数、最終受信時刻を確認することができる。

次に相手側がメールを送信すると仮定する。その場合相手側のサーバを経由して自分のサーバ側がメールを受信する。そのメールを受信する際にフィルタと同じディレクトリ内に存在するプログラム 2 を用いる。プログラム 2 はあらかじめ argv=/host/filter として、格納しておく必要がある。プログラム 2 は table を読み取るもので table にある送信者・宛先アドレスと送られてきたメールのアドレスが一致するかを判別するプログラムである。table にあるアドレスと一致するならばメールは受信させる。もしアドレスが不一致の場合はフィルタにかかり、プログラム 2 によってエラーメッセージが送信者に送られる。このシステムにより相手側は受信された場合は相手側からの返信メールが届き、またメールが紛失した受信されていなかったとしてもエラーメッセージを相手側が受け取ることができる。

例えば表 1 から、ousr1@mobile2.exa (to に格納されているアドレス) から usr1@mobile0.exa (from に格納されているアドレス) へ、または ousr1@mobile2.exa (to に格納されているアドレス) から usr1@mobile0.exa と usr2@mobile0.exa (from に格納されている複数のアドレス) へのメールは制作した table のアドレスと一致しているためメールの送信を許可する。しかし ousr1@mobile2.exa から someone@mobile2.exa (表に格納されていない宛先のアドレス) へ、または ousr1@mobile2.exa から user1@mobile0.exa と someone@mobile2.exa (表に格納されていない、いずれかの宛先のアドレス) へのメールは制作した table のアドレスと一致しないため、そのままフィルタにかかりメールを規制する。

3 環境設定

本節では、研究での環境構築設定すなわち SMTP サーバとして用いた Virtualbox、メールの送受信に用いた Postfix の設定について記述する。

3.1 VirtualBox

VirtualBox とは既存オペレーティング・システム上に追加のオペレーティングシステムを実行することができるアプリケーションである。本研究では SMTP サーバを Postfix2 機で代用する。そのために OS が 2 つ必要とする、PC 実機を 2 機使用して実験するよりも仮想 OS2 機を使用した方が安易にすみ、かつ VirtualBox は仮想 OS の設定がしやすいために VirtualBox を用いる。

3.2 Postfix

Postfix とは、フリーソフトウェアのメール転送エージェントである。Sendmail との操作上の互換性があり、かつ安全であることを指向して開発されている。本研究では、UNIX 用のメール転送エージェントとして設計されているために Postfix を用いることにした。

本研究では主な設定として/etc/postfix/main.cf と /etc/postfix/master.cf を設定した。

main.cf は主に Postfix の環境設定であり、Postfix1 上の /etc/postfix/main.cf のファイルに設定する。myhostname、myprogin には自分のホスト名、mydestination にはメールを受信するドメイン名をそれぞれ書き込む。smtp_host_lookup は SMTP クライアントがホストの IP アドレスを検索するのに使うメカニズムであり、DNS 検索が無効の場合無視される。native と書き込んだのはネイティブなネームサービスのみを使うためである。disable_dns_lookups は PostfixSMTP および LMTP クライアントでの DNS 検索を無効にする。無効になると、通常は/etc/hosts も検索する gethostbyname() システムライブラリルーチンを使ってホストを検索する。このままだと/etc/hosts で設定しているホスト名を宛先としたメールを送信すると DNS サーバにホスト名についての問い合わせパケット外とせず送られる。smtp_dns_support_level を disabled にすることで Postfix は独自に DNS を引かなくなるためシステム標準の方法でドメイン名の名前解決をする。

master.cf は主にメール受信後、配送前に filter に渡すようにするために必要であり、Postfix1 上の /etc/postfix/master.cf のファイルに設定する。危険なメール全てのコンテンツを扱うために filter という専用ローカルユーザアカウントを作る必要がある。また/var/spool/filter ディレクトリも作成する必要がある。argv=/home/fiter プログラム.pl は/home/filter にある filter.pl を実行させるものである。本研究ではここにプログラム 2 を指定する。また SMTP で到達したメールにコンテンツフィルタを使用するために、-o content_filter=filter:dummy を加える。

3.3 SQLite を用いたデータベースの作成

本研究では作成したメール log のリストをデータベース化させる必要があった。そのためデータベースを作成するのに SQLite を用いた。SQLite とは、サーバとしてではないアプリケーションに組み込むことで利用できる軽量のデータベースである。本研究で用いた PC が容量・スペック等の性能を考慮また大規模な実験をしないために使用した。

userlist の table 作成手順としてはまず、メールアドレスをプライマリーキーに設定。次に maillog を table として作成。必要であるメールアドレスの回数のカウント、最後に使った時間をデータとして抽出する。プライマ

リキーを from と to のアドレスに設定する . PRAGMA foreign_keys = true; は外部キー制約を ON にするために必要であるコマンドである .

4 プログラムの概要

本研究は perl プログラムを用いる . Postfix でメールを送るときに起動できるプログラムの 1 つであるためである . プログラムはメール log を読みリストを作り table に書き込むプログラムと table を読み取るプログラムを作成した . また本研究のプログラムには table にアクセスしなくては行けないのでデータベースに接続する必要がある . そのために以下のコードを perl プログラムに書き込むことでデータベースに接続ができる .

データベースへの接続・切断

```
use DBI;
接続
my $dbh=DBI->connect
("dbi:SQLite:dbname=database.db");
切断
$dbh->disconnect;
```

上記の例では , database.db というデータファイルへ接続または切断できる .

4.1 主要メール log を table に書き込むプログラム

プログラム 1(メール log を table に書き込むプログラム)
主要メール log を table に書き込むプログラムはメール log を抽出し振りリストを作成その後に必要なデータをデータベースの table に書き込むプログラムである .

プログラムの処理内容を以下に示す .

1. データベースと接続 (外部キー制約も有効にする)
2. field にメール log を分割して代入
3. 次のメールキューが保守されてるか from があるならば queueid は field[5] を代入 . queueid が ' ' (中身がない) と等しくない , つまり queueid があれば from{queueid} は field[6] を代入
4. ローカルのメール転送かつ status=sent ならば queueid は field[5] に代入もし queueid があれば to{queueid} を field[6] に代入また date{queueid} は getTimeStamp{field[0], field[1], field[2]} を代入
5. もし date , from , to の値が未定義値でなければ checkDB プログラムよりデータベースの table に from , to , 使用回数 (usecount) , date を格納する .
6. 15 を繰り返す .
7. データベースと切断

table の更新

更新

```
if ($row[1] < $date) {
print "Add Count: $row[0] add 1, lastused $row[1]
< $date";
my$count = $row[0] + 1;
$dbh-> do("UPDATE maillog SET usecount='&count', lastused = '&date'
where fromaddr='&from' and toaddr='&to'");
}
```

上記は table を更新する . row と date を比較し , table にあるならば使用回数 usecount 増やし , 最終使用時刻 lastused を最新の date に変更させる . table にない場合使用回数 usecount を 1 にして新規に送信者アドレス , 受信者アドレス , 最終使用時刻 lastused を書き込む .

4.2 table 読み取りプログラム

プログラム 2(table を読み取るプログラム) table を読み取るプログラムは読み取る際に table にあるアドレスを読み取ることでメールの受信をするか否かを確認するプログラムである . 処理内容を以下に示す .

1. データベースに接続
2. i=0 として宛先 recipients を recipients-1 の数まで表示することを繰り返す .
3. maillog の table の fromaddr=recipients (宛先アドレス) かつ toaddr=sender (送信アドレス) のデータを取得する SQL 文を実行する環境を用意 , 実行 .
4. 取得したデータを項目の配列 (@row) として取り出す .
5. もし @row であるならば valid と表示違うならば invalid と表示 . (invalid の末尾に recipients[i] を追加)
6. データベースと切断
7. もし i が recipients-1 と同じならば verdict (判定結果) は 1 (true)
8. もし verdict でかつ debug でなければ sendmail コマンドで message を送信 . もし debug であればエラーコードを返す .

アドレス確認

```
if ($sender && $recipients[$i]){
$validRecipients .= "$recipients[$i]";
my $count = split($validRecipients,@row);
my $update = "UPDATE maillog set
usecount='&count',lastused='&date';
$dbh-> do($update);
}
```

上記はもし宛先アドレス sender かつ送信アドレス recipients[\$i] ならば validRecipients に recipients[\$i] を代入する . count に宛先アドレス @row がでた場合に分割し

たものを代入し受信回数 usecount 最終受信時刻 lastused をアップデートする。sender とすべての宛先アドレスが maillog の table にある場合そのままメールを受け取る。もし拒否された場合は filter から Postfix にエラーコードを戻す。

5 実験

本研究で使用したホスト OS は Ubuntu Linux14.04 を使用し VirutalBox のバージョンは 5.0.8 , 仮想 OS は Ubuntu14.04server とそのクローンの 2 機を使用。VirtualBox のネットワーク設定はゲスト OS だけで通信できればよいので内部ネットワークを使用。仮想 OS に postfix , SQLite , プログラムを組み込み, 仮想 OS 上の Postfix のネットワーク設定等する。Postfix1 のメールアドレスを taro@mobile1.private , Postfix2 のメールアドレスを user1@mobile2.private として telnet を用いてメールを送信し実際にメールが送信されるか確認する。また登録されていないメールアドレスを用いた場合に送信者側にエラーメッセージを送信するか確認する。

5.1 実験結果

実験の結果, Postfix1 から Postfix2 へのメールを転送することができた。図 2 はメールが送信時, 図 3 はメールが送信失敗時の maillog である。送信成功時は status=sent , 送信失敗時は status=bounced という表記で確認できる。

```
Feb 14 12:21:44 ubuntu postfix/cleanup[2169]: 43EFE456DD: message-id=<
Feb 14 12:21:44 ubuntu postfix/qmgr[2057]: 43EFE456DD: from=<user2@private2.private>, size=190, nrcpt=1 (queue active)
Feb 14 12:21:44 ubuntu postfix/local[2177]: 43EFE456DD: to=<user1@mobile1.private>, relay=local, delay=21, delays=21/0.01/0/0.01, dsn=2.0.0, status=sent (delivered to command: procmail -a "EXTENSION")
Feb 14 12:21:44 ubuntu postfix/qmgr[2057]: 43EFE456DD: removed
```

図 2 送信成功時の maillog

```
Feb 14 12:01:06 ubuntu postfix/qmgr[1694]: 3469A44A93: from=<user4@mobile1.private>, size=187, nrcpt=1 (queue active)
Feb 14 12:01:07 ubuntu postfix/pipe[2019]: 3469A44A93: to=<user1@mobile2.private>, relay=filter, delay=22, delays=21/0.03/0/0.64, dsn=5.3.0, status=bounced (Command died with status 255: "/home/taro/filterTest-20160126.pl". Command output: DBI connect('dbname=/pfFilter.sqlite3', '...', ...) failed: unable to open database file at /home/taro/filterTest-20160126.pl line 52. Can't call method "prepare" on an undefined value at /home/taro/filterTest-20160126.pl line 63, <STDIN> line 6. Sender: user4@mobile1.private Recipients: 0 user1@mobile2.private: )
Feb 14 12:01:07 ubuntu postfix/cleanup[2018]: 373D444D06: message-id=<20160214030107.373D444D06@postfix2>
```

図 3 送信失敗時の maillog

また, 図 4 よりデータベース内にあらかじめ登録していないアドレス, すなわち maillog の無いアドレス user4@mobile1.private へのメールが届かないことを示すエラー文が送信側である user1@mobile2.private に届くことも確認できた。

課題として, リストを作る際に送信側 (from) , 受信側 (to) の最終受信時刻 (last used) , 受信回数 (count) 等をそれぞれ分けて表記できるプログラムの制作が挙げられ

る。また両 server とともに本システムを導入した場合, 不具合が生じる可能性がある。

```
From MAILER-DAEMON Sun Feb 14 12:01:08 2016
Return-Path: <
X-Original-To: user4@mobile1.private
Delivered-To: user4@mobile1.private
Received: from postfix2 (mobile2.private [192.168.1.254])
  by postfix1 (Postfix) with ESMTP id 53B4A456DD
  for <user4@mobile1.private>; Sun, 14 Feb 2016 12:01:08 +0900 (JST)
Received: by postfix2 (Postfix)
  id 373D444D06; Sun, 14 Feb 2016 12:01:07 +0900 (JST)
Date: Sun, 14 Feb 2016 12:01:07 +0900 (JST)
From: MAILER-DAEMON@mobile2.private (Mail Delivery System)
Subject: Undelivered Mail Returned to Sender
To: user4@mobile1.private
Auto-Submitted: auto-replied
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;
  boundary="3469A44A93.1455418867@postfix2"
Message-Id: <20160214030107.373D444D06@postfix2>

This is a MIME-encapsulated message.

--3469A44A93.1455418867@postfix2
Content-Description: Notification
Content-Type: text/plain; charset=us-ascii

This is the mail system at host postfix2.

I'm sorry to have to inform you that your message could not
be delivered to one or more recipients. It's attached below.

user4
```

```
For further assistance, please send mail to postmaster.

If you do so, please include this problem report. You can
delete your own text from the attached returned message.

The mail system

<user1@mobile2.private>: Command died with status 255:
"/home/taro/filterTest-20160126.pl". Command output: DBI
connect('dbname=/pfFilter.sqlite3', '...', ...) failed: unable to open database
file at /home/taro/filterTest-20160126.pl line 52. Can't call method
"prepare" on an undefined value at /home/taro/filterTest-20160126.pl line
63, <STDIN> line 6. Sender: user4@mobile1.private Recipients: 0
user1@mobile2.private:

--3469A44A93.1455418867@postfix2
Content-Description: Delivery report
Content-Type: message/delivery-status

Reporting-MTA: dns: postfix2
X-Postfix-Queue-ID: 3469A44A93
X-Postfix-Sender: rfc822: user4@mobile1.private
Arrival-Date: Sun, 14 Feb 2016 12:00:45 +0900 (JST)

Final-Recipient: rfc822: user1@mobile2.private
```

図 4 送信失敗時のエラー文

参考文献

- [1] KDDI: オススメ設定 (accessed: Jan. 2016). <http://www.au.kddi.com/support/mobile/trouble/forestalling/mail/anti-spam/fillter/function-19/>.
- [2] NTTdocomo: 迷惑メールおまかせブロック (accessed: Jan. 2016). https://www.nttdocomo.co.jp/service/safety/omakase_block/.
- [3] softbank: かんたん設定 (accessed: Jan. 2016). <http://faq.mb.softbank.jp/detail.aspx?cid=2594id=2594>.
- [4] 情報通信政策研究所 調査研究部: 「平成 26 年情報通信メディアの利用時間と情報行動に関する調査報告書」の公表 (2015). http://www.soumu.go.jp/menu_news/s-news/01iicp01_02000028.html.