研究論文

悪用者過多時代に即した堅牢なメイルサーバの構築

広瀬　雄二

概　　要

インターネットのもっとも基本的な通信プロトコルであるSMTPは、インターネットが性善説だけで暮らしていける時代に設計されたため、構造が非常にシンプルで普及の一助となった。しかし現在ではこれを逆手に取り、差出人を詐称したり、悪質な広告を送ったり、脆弱なOSを狙ったウィルスを自動的に流布したりする悪用者が増加し、本来望んで受け取りたいメッセージ数をはるかに上回る迷惑メイルが送信され続けている。また、迷惑メイル以前から観察されるメイルサーバプログラム自体への攻撃も後を絶たず、セキュリティ脆弱性の小さなMTAは依然求められ続けている。

そうした中、管理コストが小さく、セキュリティホールの可能性がもっとも低いとの評判高いqmailは、迷惑メイルの問題が顕在化する以前に公開されたものであるため、それらの問題対策手法を含めた体系的なシステム全体像が見えないという問題があり、堅牢なメイルサーバ構築を目指す上での障害となっている。本稿ではqmailと近代的パッチを統合したnetqmailをベースに、迷惑メイル拒絶のための対策をどのように取るべきかを、現状に即した手法を開発・採用し考察していく。

Abstract

We all realize SMTP is simple enough to implement it. That is a reason for email system had become essential infrastructure. But its simpleness has also been bringing the great social problem of attacking to or abusing of email system. Cracking and spamming. To prevent the system intrusion, we know that 'qmail' is one of best choice because it was designed with best care on security. But the one who want to introduce qmail from scratch inevitably come up

against the problem that there's few information about solving email abusing, because it was designed at the end of 20th century, when abusing of email was much less than these days.

In this paper, we consider the security performance of email system and examine the method of spam rejection at SMTP session. And we propose a methodical way of constructing a secure modern mail server with qmail.

1 背　　景

電子メイルは、SMTP[1][1]に基づいて配送が行なわれている。SMTPはその名のとおり単純なプロトコルであることから、実装も容易で多くのソフトウェアがこれに対応した。そのおかげで生活基盤ともいえるほどの普及を遂げた。

メイル送信希望を行なうhost-Aが、宛先ユーザを持つhost-Bに対して送信処理を行なうときのメッセージの授受を模したものを示す。

ホスト	送信文字列	意味
host-B	220 host-B ESMTP	Greeting メッセージ
host-A	HELO host-A	クライアント名の送出
host-B	250 host-B	成功
host-A	MAIL FROM: user1@host-A	送信者アドレスの通知
host-B	250 ok	成功
host-A	RCPT TO: user2@host-B	宛先アドレスの通知
host-B	250 ok	成功
host-A	DATA	本文送信開始
host-B	354 Go ahead.	送信待ち
host-A	～～本文が続く～～	本文送信中
host-A	.	本文終了
host-B	250 ok 1159591442	受理成功
host-A	QUIT	セッション終了
host-B	221 host-B	成功

[1] Simple Mail Transfer Protocol

このようにSMTPではHELOパラメータとして自己ホスト名、MAIL FROMパラメータとして送信者アドレス、RCPT TOパラメータとして受信者アドレス、DATAコマンドに続けて本文、を順に送信することで相手にメイルを届けることができる。ここで注意が必要なのは、HELOとMAIL FROMのパラメータに正しいものを入力することが期待されてはいるものの、不正であっても送信できるという点である。このため、インターネットに接続し、任意のホストのTCPの25番ポートに接続できる機械を持っている者なら誰でも送信者詐称メイルを送ることができる。あるいは、インターネットに繋がっているPC でコンピュータウィルス等に感染しているMicrosoft Windowsマシンなどから、外部のメイルサーバに詐称メイルや、迷惑メイル、ウィルス入りメイル等を送信し続けることができる。

ただし、詐称等ができる、というだけで、送信経路はメイルサーバに記録されるため、どの端末が発信源かは必ず特定できる。それゆえ、迷惑メイル送信者達は足の付くのを警戒し、接続毎に変わるIPアドレスを利用したり、世界中にある無防備なWindows-PCにワームを仕掛けては迷惑メイルの発信源を大量に作り出している。

2　メイルサーバに求められるもの

インターネット接続のためのコストが低くなった現在では、メイルサーバは常に悪意ある接続にさらされることになる。そうした「攻撃」から傘下の利用者、そしてサーバ自身を守るためにはサーバプログラムそのものの頑強性と、抗迷惑メイル性の２点が重要である。

2.1　MTAのセキュリティ

プログラムのバグのうち、プログラムそのものの動作、あるいはシステム全体の安全性をおびやかすきっかけを与えるものをセキュリティホールという。プログラムに潜むバグはプログラムの規模、複雑さが増すほど多くなるのが普通で、それを裏付けるようにsendmail[2]は常にバグ修正が繰り返されている。

1990年代後半から、あまりに複雑化したsendmailを他山の石として、でき

るだけ簡潔な構造を持つ小さなプログラムの集合体として開発されるようになった。qmail[3]、Postfix[4]はともにsendmailを置き換えることを目標の一つとして設計・開発されたMTAで、現在では高いセキュリティを持つMTAとして高い評価を得ている。とりわけqmailは、セキュリティホールの発生する可能性を極限までなくすことを最大目標としていることもあり、1988年に発表されたqmail-1.03は現在までセキュリティホールが発見されていない。

クラッキングツールのはびこる現在では、ネットワークサービスデーモンプログラム自体のセキュリティホールが突かれ、サーバマシンを乗っ取られる事件が次々と起きている。これを未然に防ぐため、ネットワークサービスの中でも必要度の高いメイルサーバプログラムが強健な作りになっていることは必須といえる。

2.2 迷惑メイル対策

セキュリティホールを利用した攻撃だけでなく、電子メイルにはもう一つ危険性が伴う。利用者が望まないのにもかかわらず、一方的に広告やウィルス、害悪記事などを送りつけて来るいわゆる迷惑メイルである。これは1節で述べたように、SMTPが低コストで利用できる点、簡単に実装できる点、送信者の本人確認が不要である点[2]に起因する。

発信されているメイルの過半数が迷惑メイル[11]という現状は、放っておけば利用者のメイルボックスのほとんどが迷惑メイルになるという事態を引き起こし、必要な情報を抽出することが困難になる。サーバの機能として迷惑メイルを極力排除するように設定することはもはや避けられない要求となっている。

[2]実際のところ本人確認はコストがかかる上、複雑で、どの組織が認証すればよいかなど政治的な問題がも絡み、認証個体が偽装される可能性などを考えると現実的でない。本人確認は当事者どうしが当事者の責任で行なうという現状は批判されるべきものではない。

3　netqmailによるサーバ構築

インターネット接続コストの低廉価とPC-Unixを含めたUnixサーバの導入コストの低減化に伴い、小さな組織でも独自のメイルサーバを構築することが容易になった。しかし、先述のセキュリティ問題を考慮すると攻撃の標的となりやすいOS標準構成のものではなく独自にセキュリティ重視のサーバシステムを構築するのが望ましい。そのために、本稿ではqmailを利用した現状に即したメイルサーバを新たに構築するための必要事項を検証し、その指針を示すことにする。

qmailが高いセキュリティ性を持つことは既に述べたが、qmail-1.03が登場した時代に比べ、現在では様々な悪用防止策が必要になっている。そうした技術、あるいはそれを実現するプログラムはqmail-1.03よりも後発のものである。それゆえ、qmailの導入解説文献はそうした悪用防止策の実装が作られる以前に書かれたものが多く、それらには触れていない。もちろん文献がないからといってqmailを利用した管理者が悪用防止策を取っていないわけではなく、それらの防止策を漸増的に追加していっている。つまり、システム全体を統括的に導入したわけではなく、そのための指針を体系的に把握してからサーバ全体の構築に臨んだというわけではない。このような現象がqmailによる近代的サーバ構築の体系的情報の欠乏状態を生み出している。

3.1　netqmail

netqmail [5] は、qmail-1.03に対する現状に即した「必要最低限の厳選パッチ集」をネットワークコミュニティが監修の上構成してバージョン1.05としたパッケージである。インストール方法、設定・管理方法、動作機構といった基本的な部分はqmail-1.03と全く同じである。

3.2　メイルサーバ要件

メイルサーバでは、MTAが配送にかかわる動作をするだけでは不十分である。先述の迷惑メイル拒絶機能も重要になってきていることもあるが、それ以外にもメイルを利用者のメイルリーダに届けるための機能や、利用者がMTA

の持つ機能を最大限に利用できるようにするための様々なサービスを同時に揃えなければ実用できない。その意味で、メイルサーバが十分にサービスを提供するためには以下の機能が必要だといえる。

・基本配送機能

ユーザから送信依頼を受けて同一マシン内、あるいは外部ネットワークのユーザのところへ届ける基本的な機能は必ず必要である。ただし、これだけでは不十分で、以下に示す機能を併せて提供する必要がある。

・SMTPサーバ機能

外部の別メイルサーバから来たメッセージをローカルユーザに届ける機能も必要である。かつてのsendmailのように、基本配送機能を行なうプログラムと同一プログラムで行なう場合もあるが、現在ではSMTPサーバはプログラムの動作権限を縮小した別プログラムで行なうのが一般的である。外部ネットワークからの接続は「悪意を持ったユーザ」による攻撃である可能性もあるので、サーバプログラムをスーパーユーザではなく一般ユーザ権限でなおかつ限られたリソースだけ与えて動かすのが安全性の観点から望ましい。

・アクセス制御(悪用防止)

別ホストからのSMTP接続要求があった場合、そのホストがどのネットワークに位置するかによって信頼度も変わってくる。そうした理由から、接続クライアントのIP アドレスによって接続を拒否したり、SMTPサーバの挙動を変更する機能が求められる。sendmailのようにSMTPプログラム自身がその機能を内包している場合もあるが、これもやはり「ネットワーク制御」に特化したシンプルなプログラムに任せた方がバグの可能性も低く、またメイルサービス以外での利用も可能になり効率的である。qmailではアクセス制御プログラムとしてucspi-tcpパッケージに含まれるtcpserverを利用する。

・SMTP送信時認証

上記と関連するが、利用者がメイル送信を目的としてSMTPサーバを使用する際に、正当な利用者であることを確認するためにSMTP-AUTHが用いられる。qmail ではqmail-smtpd-auth[7]が利用できる。ただし、そのま

まではqmailの持つ柔軟なアドレス拡張機能が利用できないため、それを実現したパッチ[8](qmapop-smtp-auth)を適用したものを利用する。

・迷惑メイルフィルタ

迷惑メイルは

- ウィルス・ワーム
- スパム

に大別され、それぞれ排除対策すべき場所と必要なロジックが異なる。一概にはいえないが、前者は概ねWindowsに感染したウィルスまたはワームが、他のPCへの感染目的でPC所有者のメイル用ソフトウェアに登録されている住所録、あるいはインターネットで公開されているWebページから無作為に集めたメイルアドレスに対して自己増殖を試みる感染プログラムなどを送りつけるものである。このため、人間が普段使っているメイル用ソフトから、人間が普段行なっている手順を模して送信することが多い。そのためプロバイダによる正規のメイルサーバを介して送信されることが多いので、これを検出したい場合はメイル本文を解析する必要がある。逆に後者（スパム）は、マルウェア[3]が仕組まれたPCから直接手当たり次第に直接送信されることが多い。つまり、正規のメイルサーバを経ずに届くことが多いため、メイル送信をしているホストを調べたり、正規のメイルサーバとは思えないSMTPパラメータを検出するだけで（メイル本文を見ずに）済む可能性がある。

ウィルス・ワームに関しては既に商業ベースで多くの検出プログラムが供給されている。それらの導入に関しては本論の域を出るためここでは扱わない。いっぽう後者の、メイル送信時の手順の特徴と送信者の身元情報を利用した選別は、筆者が開発したantibadmail[10, 11, 12]で行なうことができる。antibadmailの利用により、不審なIPアドレス、明らかに偽造と判定できる送信などをSMTPの層で直ちに遮断できる。

・POP/IMAPサーバ

[3]malware; 危害を加えることを目的として動く悪意あるプログラムを指す一般名称。ウィルスやワームの場合もある。

メイルサーバに届いたメイルを、利用者の手元に届けるためにはPOPやIMAPのような伝送サーバが必要となる。これは利用者の利用形態に即したものであればどれを選択しても問題ないが、qmailと親和性高く振る舞う拡張の施してあるUW-imapd extensions［13］を利用する。

以上を踏まえ本稿では以下のソフトウェアの導入の指針を示す。

daemontools, tcpserver, netqmail, antibadmail,
qmail-smtpd-auth+cmd5apoppw+qmapop-smtp-auth, imapext

4　サーバプログラムの導入

メイルサーバは単一のソフトウェアだけでは成り立たない。複数のサーバプログラムを導入する必要があり、それぞれ互いに関連しあって動作する。導入時にはひとつひとつのソフトウェアの挙動を試験しながら進めなければならない。ここでは、動作確認の取りやすい順番で進める方法を示す。ただし、本稿は指針を示すのみという位置付けであるから、手順に失敗したときの具体的な問題解決方法は示さない。

また、各ソフトウェアの導入方法・設定はただ一つに決まるものではない。ほとんどの設定は管理者の裁量で変えられるが原則として、一つのソフトウェアパッケージの独立性を高め更新や置き換えをしやすくすることを考慮した例としている。もう一点、サーバプログラムの設定は、動作定義が複雑になると設定ミスが起きやすくなり、どんなに高品質なプログラムを使っていても障害を引き起こしやすくなる。そのため、一つの設定の分担範囲を絞り込み、障害が起きたときの問題の切り分けがしやすくなることも重要な方針としている。

表1　仮定するサーバ情報

ホスト名	mail.example.com
IP　アドレス	192.168.1.25
ネットマスク	255.255.255.0
メイルドメイン	example.com

実行例4.2.1　daemontools の導入

```
% ftp http://cr.yp.to/daemontools/daemontools-0.76.tar.gz
% gzip -dc daemontools-0.76.tar.gz | tar xpf -
% cd admin/daemontools-0.76
% package/compile
% su
# mkdir -p /usr/local/daemontools
# cp -r command /usr/local/daemontools/bin
# (cd /usr/local/daemontools/bin
 > chmod +w svscanboot
 > printf ',s,/command,/usr/local/daemontools/bin,\nwq\n' |
 > ed svscanboot
 > chmod -w svscanboot)
# package/run
```

4.1　設定サーバ環境の仮定

単純化のためメイルサーバを構築するホストに関する情報を表1のとおりと仮定する。またメイルドメインに関するDNSの設定は完了しているものとする。また、作業例中の%は一般ユーザでの作業を、#はスーパーユーザでの作業を意味するものとする。ユーザやグループの追加等、一部のシステム管理コマンドはNetBSDに準拠したもので示す。

4.2　daemontoolsの導入

daemontools[14]は、各種デーモンプログラムの起動や停止をOSによらない統一的な手順で管理できるようにするためのツールである。/usr/local/daemontoolsディレクトリ以下にこれを導入する例を実行例4.2.1に示す。以上の導入作業によりデーモンプログラムの起動を、svcコマンドで行なえる。以下の手順例では/usr/local/daemontools/binにコマンド検索パスが通っているものとする。daemontoolsの細かい設定や運用法については文献[15]が詳しい。

実行例4.3.1　ucspi-tcpの導入

```
% cvs -d :pserver:anonymous@yatex.org:/qmail co -r paranoid+bsd ucspi-tcp
% cd ucspi-tcp
% echo /usr/local/ucspi > conf-home
# make setup heck
```

実行例4.4.1　netqmailのソース準備

```
% ftp http://www.qmail.org/netqmail-1.04.tar.gz
% gzip -dc netqmail-1.05.tar.gz | tar xpf -
% cd netqmail-1.05
% sh collate.sh
```

4.3　ucspi-tcpの導入

ucspi-tcpはTCPを利用した入出力処理を行なうプログラムを作成するためのプログラムインタフェースの集合パッケージだが、サーバ管理の用途としても利用できる。これを/usr/local/ucspi以下に導入する例を実行例4.3.1に示す。なお、リスト中で利用しているソースは、ucspi-tcp-0.88[4]をIPv6化[16]したものに、IPアドレスベースのルールがマッチした場合にも、ホスト名ベースのルールのマッチングを適用するよう筆者が修正を施したものである。この修正によりスパム送信者の利用しているホストの情報を、常にIPアドレス/ホスト名両側面から評価できるようになる。

4.4　qmailの導入

SMTP-AUTHを処理するメイルサーバを構築する場合、単一のqmailシステムを使うより、「通常処理用」と「SMTP-AUTHコネクション処理用」を分けて導入する方が管理手順が簡潔になり、設定誤りを軽減できる。ここでは、通常処理用を導入したのちSMTP-AUTH用を導入する手順を示す。また、最近ではインターネット接続プロバイダのスパム対策の一貫として、外部送信用SMTPポートとして25番ではなく、サブミッションポート（587番）の利用を

[4]http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz

強制するケースが増えているので、デフォルトのポート（25番）のみならず、サブミッションポート用のSMTPデーモンを用意する必要が出てきている。これも、SMTP-AUTH用のものを別個に導入する意義の一つである。

実行例4.4.2　SMTP-AUTH用パッチの適用

```
% ftp http://members.elysium.pl/brush/qmail-smtpd-auth/dist/qmail-smtpd-
 auth-0.31.tar.gz
% ftp http://www.gentei.org/~yuuji/software/qmapop-smtp-auth/qmapop-smtp-
 auth-3.diff
% gzip -dc qmail-smtpd-auth-0.31.tar.gz | tar xpf -
% cp qmail-smtpd-auth-0.31/base* netqmail-1.05
% patch -d netqmail-1.05 < qmail-smtpd-auth-0.31/auth.patch
% patch -d netqmail-1.05 < qmapop-smtp-auth-3.diff
```

実行例4.4.3　ユーザ／グループの追加

```
# groupadd nofiles
# useradd -g nofiles -d /var/qmail/alias alias
# useradd -g nofiles -d /var/qmail qmaild
# useradd -g nofiles -d /var/qmail qmaill
# useradd -g nofiles -d /var/qmail qmailp
# groupadd qmail
# useradd -g qmail -d /var/qmail qmailq
# useradd -g qmail -d /var/qmail qmailr
# useradd -g qmail -d /var/qmail qmails
```

4.4.1　通常処理用qmail

qmailの導入は、まずnetqmail-1.05本体ソースの準備（実行例4.4.1）をした上で、同ディレクトリのソースにSMTP-AUTHパッチ［7, 8］を当てる。

パッチ当てが完了したら、qmail稼動に必要なユーザ/グループを追加（実行例4.4.3）し、そののち本体のコンパイルと初期調整を行なう。

次にローカルデーモンの起動スクリプトを作成し、/var/qmail/rcという実行属性つきのファイルとして保存する（リスト4.1）。システム再起動後に有効となるよう、/etc/rc.local等に追加したうえで、現行環境でも起動しておく（実行例4.4.5）。

実行例4.4.4　qmailのコンパイルと初期調整

```
# mkdir /var/qmail
# make setup check
# ./config
```

リスト4.1　/var/qmail/rc

```
#!/bin/sh
# Using splogger to send the log through syslog.
exec env - PATH="/var/qmail/bin:$PATH" qmail-start ./maildir/ splogger
qmail &
```

実行例4.4.5　/var/qmail/rcの起動

```
# /var/qmail/rc
```

4.4.2　SMTP-AUTH処理用qmail

SMTP-AUTH処理用のものは、別のディレクトリに導入する。netqmail-1.05のソースディレクトリに戻り、導入先ディレクトリを/var/qmail/authに変更してからコンパイルする（実行例4.4.6)。

これにより4.4節で導入したものと全く同じ構成が/var/qmail/auth以下に入るが、このうちSMTP-AUTH専用SMTPデーモンの挙動に固有のものは

- bin/qmail-smtpd
- control/me
- control/rcpthosts

だけであるため、それ以外は消去、またはシンボリックリンクに改める。この作業例を実行例4.4.7に示した。コマンド行の最後から2 行目では、SMTP-AUTH認証に成功しないクライアントからの受信を許可するメイルドメイン一覧をでたらめな文字列にして、未認証クライアントからのメイルを一切受けないようにしている。

実行例4.4.6　SMTP-AUTH用qmailのコンパイルと初期調整

```
# echo /var/qmail/auth > conf-qmail
# make clean setup check
# ./config
```

実行例4.4.7　SMTP-AUTH用qmail固有ファイルの構成処理

```
# cd /var/qmail/auth
# mv bin/qmail-smtpd .
# rm -rf */                          ディレクトリのみ全て消す
# ln -s ../bin .                     bin を共通に
# mkdir control; cd control
# echo acceptor.example.com > me
# echo detaramenara-nandemo-OK > rcpthosts
# ln -s ../../control/* .            他の control ファイルは共通に
```

4.5　antibadmailの導入

4.4.1節で導入した通常処理用qmailで起動するSMTPデーモン（qmail-smtpd）は、SMTP 標準ポート（25番）でリクエストを受けるため、望むメイルも迷惑メイルも送信依頼を受け付けることになる。SMTP接続時の情報だけで直ちにそれと分かる迷惑メイルを受信拒否するためのSMTPラッパーantibadmail[10]を導入（実行例4.5.1）し、SMTPデーモン起動スクリプトに追加する。このスクリプトはdaemontoolsのsupervise[5]に起動させるので、superviseが管理するディレクトリを作成し、その中にrunというファイル名でスクリプトを置く（実行例4.5.2、リスト4.2)。SMTPデーモンによるログを受け取るスクリプトを、やはりsupervise経由で起動するよう作成するのだが、ログ書き込みを行なうユーザも併せて作成する（実行例4.5.3)。このユーザ権限で動くようログ取得スクリプトを実行属性つきで作成する（リスト4.3)。

[5]daemontools に含まれるコマンドでデーモンプログラムの起動 / 停止を制御する常駐プログラム。

実行例4.5.1　antibadmailの導入

```
% cvs -d :pserver:anonymous@yatex.org:/qmail co antibadmail
% cd antibadmail
% make
# cp antibadmail f2d /var/qmail/bin
# cd /var/qmail/control
# printf 'local1.info\t/var/log/smtp-stat' >> /etc/syslog.conf
# pkill -1 syslogd
# printf '/var/log/smtp-stat\t\t640\t40\t7\t*\t$W0D00Z' >> /etc/newsyslog.
 conf
```

実行例4.5.2　supervise管理用ディレクトリの作成

```
# mkdir -p /var/qmail/sv/smtpd/log
```

リスト 4.2　/var/qmail/sv/smtpd/run

```
#!/bin/sh
px=/usr/local
ul=$px/bin
dt=$px/daemontools/bin
ut=$px/ucspi/bin
qm=/var/qmail/bin
exec env - \
PATH=$qm:$dt:$ut:$qm:/bin:/usr/bin:/sbin:/usr/sbin:$ul \
envuidgid qmaild softlimit -d950000 \
tcpserver -vR -c40 -p -U -x smtp.cdb 0 smtp antibadmail qmail-smtpd 2>&1
```

実行例4.5.3　ログ書き込みユーザの作成

```
# useradd -g nofiles maillog
# chown maillog /var/qmail/sv/smtpd/log
```

リスト 4.3　/var/qmail/sv/smtpd/log/run

```
#!/bin/sh
px=/usr/local
dt=$px/daemontools/bin
exec env - \
PATH=$dt:/bin:/usr/bin:/usr/sbin: \
setuidgid maillog \
multilog t !tai64nlocal s999999 n10 t ./main
```

4.6 tcpserverルールファイルの作成

リスト4.2ではSMTPデーモンプロセス（qmail-smtpd）を起動する際のアクセス制御を行なうルールファイルとしてsmtp.cdbを指定している。これはSMTP接続クライアントに、接続自体を許可するか、メイル送信要求を許可するか、リレーを許可するか、などの制限を掛けるためのルールを登録する。

一般的には全てのクライアントからの接続を許可し、そのうちローカルホスト（127.0.0.1）とLAN内のホストからの接続ではリレーも許可するという方針を取る。しかし近年の電子メイル悪用者の増加により、信頼できないホストを見定めて、メッセージの受け取り自体を拒否する必要も出てきた。このためにtcpserverルールファイルは、

1．LAN内のクライアントにリレーを許可するルール

2．迷惑メイル拒絶のためのantibadmail用ルール

の2つに分けて考える必要がある。ここでは1に関するものをsmtpというファイルに記述するものとする。なお、2については筆者らが作成・公開している、日本向け迷惑メイル対策公開データベース（以下spamdb）が利用できるのでそれを用い、1と2を合成したものをsmtp.cdbとする。

本稿で仮定した環境の場合smtpファイルはリスト4.4のようになる。また、spamdbは実行例4.6.1にならい公開cvsサーバより取得しsmtpファイルと合わせる。実行例では同時にantibadmailが迷惑メイル判定時に利用するデータベースディレクトリのシンボリックリンク作成も行なっている。

以上、全てが完了したうえで/serviceディレクトリにシンボリックリンクを張ると、5秒以内にSMTPデーモンとログ取得プロセスが起動する（実行例4.6.2）。

リスト4.4　/var/qmail/sv/smtpd/smtp

```
127.0.0.1:allow,RELAYCLIENT=""
192.168.1.:allow,RELAYCLIENT=""
:allow
```

実行例4.6.1　spamdbの取得

```
# cd /var/qmail/sv/smtpd
# cvs -d :pserver:anonymous@yatex.org:/qmail co spamdb
# (cd spamdb; make)
# ln -s $PWD/spamdb/bad*dir /var/qmail/control
# cp spamdb/Makefile.tcprule Makefile
# ln -s spamdb/smtp-badhost
# make
```

実行例4.6.2　通常用SMTPデーモンの起動

```
# ln -s /var/qmail/sv/smtpd /service
# sleep 5; svstat /service/smtpd
```

4.7　SMTP-AUTH用SMTPデーモンの起動

通常処理用SMTPデーモンとほぼ同様の手順でSMTP-AUTH用のものを起動する。これもsuperviseに管理させるので専用のディレクトリを作成し、そこにrunという名前で起動スクリプトを置く。ただし、SMTP-AUTHでは送信者の認証用パスワードの合否判定を行なうプログラム（cmd5apoppw）が必要となるのでこれを先に導入し、同時にログ取得スクリプトも用意しておく（実行例4.7.1）。

導入後、cmd5apoppwを利用する形でSMTP-AUTHが有効化されたSMTPデーモンを起動するスクリプトを作成し、supervise管理ディレクトリに置く（リスト4.5）。

2つのスクリプトの準備が了したら、/serviceディレクトリにシンボリックリンクを張り、superviseに起動要求する（実行例4.7.2）。

実行例4.7.1　cmd5apoppwの導入

```
% cvs -d :pserver:anonymous@yatex.org:/qmail co cmd5apoppw
% cd cmd5apoppw
# make all install
# mkdir -p /var/qmail/sv/auth-smtpd/log
# chown maillog /var/qmail/sv/auth-smtpd/log
# cp /var/qmail/sv/{,auth-}smtpd/log/run
```

リスト4.5　/var/qmail/sv/auth-smtpd/run

```
#!/bin/sh
px=/usr/local
ul=$px/bin dt=$px/daemontools/bin ut=$px/ucspi/bin qm=/var/qmail/bin
exec env - PATH=$qm:$dt:$ut:$qm:/bin:/usr/bin:/sbin:/usr/sbin:$ul \
envuidgid qmaild softlimit -d950000 \
tcpserver -vR -c40 -p -U -x smtp.cdb 0 smtp \
   qmail-smtpd mail.example.com cmd5apoppw /bin/true 2>&1
```

実行例4.7.2　SMTP-AUTH用SMTPDの起動

```
# ln -s /var/qmail/sv/auth-smtpd /service
# sleep 5; svstat /service/auth-smtpd
```

4.8　POPサーバの導入

qmailを基盤としたメイルサーバシステムに適用可能なPOPサーバは複数あるが、qmailのもつ豊富な拡張アドレスを有効に活用できるものとして、imapext[13]を導入する。imapextはワシントン大学版IMAP[17]をベースに、

・APOP対応
・Maildir対応
・拡張アドレス対応
・POP before SMTP対応

など、パスワード漏洩防止機能とqmailの利用形態に即した拡張を筆者らが施したものである。CVSリポジトリに登録された最新版を導入する例を実行例4.8.1に示す[6]。

この実行例では、LAN内のホストのみにパスワードを暗号化しないPOPアクセスを許可させるためのtcpserverルールファイル（pop.cdb）を生成している。ここまで作成したら、supervise用runスクリプトを作成し実行属性を付け、/serviceディレクトリに管理ディレクトリのシンボリックリンクを張る（リスト4.6、実行例4.8.2）。

[6]make bsf の 'bsf' の部分は OS により異なる。OS に応じた make ターゲットの名前はトップディレクトリの Makefile に記載されている。

実行例4.8.1　imapextの導入

```
% cvs -d :pserver:anonymous@yatex.org/imapext co -r imapext-2006 imapext
% cd imapext
% printf ',s/-DPOPBEFORESMTP//\nwq\n' | ed src/osdep/unix/Makefile
% make bsf
# mkdir /usr/local/sbin
# install -m 700 APOPtools/deapop /usr/local/sbin
# install -m 755 APOPtools/apoppasswd /usr/local/bin
# mkdir /var/qmail/pop
# cp ipopd/ipop3d /var/qmail/pop
# cd /var/qmail/pop
# cat << _EOF_ | tee pop | tcprules pop.cdb tmp$$
127.0.0.1:allow,INTRANET=""
192.168.1.:allow,INTRANET=""
:allow
_EOF_
```

リスト 4.6　/var/qmail/pop/run

```
#!/bin/sh
exec env - \
PATH=/var/qmail/pop:/usr/local/ucspi/bin:/usr/bin:/usr/sbin \
tcpserver -vR -c40 -x pop.cdb 0 pop3 ipop3d 2>&1
```

実行例4.8.2　imapextの起動

```
# ln -s /var/qmail/pop /service
# sleep 5
# svstat /service/pop
```

5　メイルサーバの運用

メイルサーバの構築が完了し、DNS情報も登録できていればメイル配送が始まる。その先に「セキュリティパッチ」などを当てる心配が皆無なのがqmailによるサーバ運用の最大の利点だろう。その他、サーバ運用を考える上で配慮すべき点について考察する。

5.1 セカンダリサーバ

旧来メイルサーバは最終的にメッセージを受け取る「プライマリサーバ」と、プライマリサーバがダウンしているときに臨時で預る「セカンダリサーバ」を用意するのが常識であった。セカンダリサーバを配備することで、メイル送信者側のサーバに配送待ちのメッセージキューが残ることがないので、依頼者に対して負担を掛けずにすむという配慮であった。

しかし最近ではセカンダリサーバが「なんでも受け取る」という性質を逆手に取り、迷惑メイルの送信先としてわざわざセカンダリサーバを選び、ランダムな宛先アドレスを大量に送りつけて来る[7]事例が増えている。最終的な受け取りアドレスが実在するかどうかはプライマリサーバに問い合わせる必要があるが、そのような問い合わせシステムを導入する手間がかかる上、仮に導入したとしてもその問い合わせによる負荷上昇は免れない。

また、セカンダリサーバが本来活躍すべき状況である「実際にプライマリサーバが落ちているとき」に、ランダム辞書攻撃が来たとしたら実在確認の問い合わせができないため宛先不在の可能性の高い迷惑メイルも全て受け取らざるを得なくなってしまう。

その一方で、かつてはインターネット回線がダウンしメイルサーバが孤立することや、サーバ事態がダウンしてその復旧に時間がかかることなどが多かったが、最近では回線やシステムの品質が上がったためそうした障害が長時間に及ぶことが少なくなった。

以上を考慮すると、プライマリサーバの障害時の補佐としてのセカンダリサーバの重要性はきわめて低く、むしろ迷惑メイル攻撃の弱点となることを優先的に考える必要が出てきた。言い換えれば、現在では必ずしもセカンダリサーバを準備する意味は乏しいといえる[8]。

---

[7] 人名としてありそうな単語が記載された辞書をもとに機械的に何百何千ものメイルを送信する。辞書攻撃。

[8] もちろん大量ユーザをかかえる大規模サーバの負荷分散としてのセカンダリサーバ（群）の意味は失われない。

5.2 迷惑メイルサーバ情報の管理

antibadmailは迷惑メイルを効果的に撃退するが、迷惑メイルは常に送信者と管理者の「いたちごっこ」である。送信用として悪用しているプロバイダのアドレスからの送信が「防御」されるようになれば、送信者は別の場所を探す。そのため、antibadmail導入時に同時に取得したspamdbをそのまま使い続けていると新しい場所から送りつけて来る迷惑メイルに対応できない。そのため、迷惑メイル対策の最新データベースを常に利用するよう心掛ける必要がある。それには、spamdbのあるディレクトリで「`cvs up && make`」を継続的に行なう必要がある。

また、spamdbが完全に自己サイトに適合するものばかりを含むわけではないことに注意し、常にそこに追加されたものを観察する必要がある。なぜなら、迷惑メイルの判定には絶対基準があるわけではないからである。たとえば、ある悪徳業者がある人に勝手に送りつけたメイルと同一内容であっても、別の人がそれを送るよう事前に希望していたならそれは迷惑メイルではない。そのため、spamdbを利用する場合は自らの管理するサイトの利用者が希望しているメイルを拒絶するようなデータが登録されていないかに配慮する必要がある。

6 おわりに

電子メイルは根幹的なサービスであるだけに様々な「攻撃」から保護する必要がある。そのような場合に、MTAそのものにバグが出続けるようでは管理コストが無視できない。また、qmailをベースとしたシステムは小さなプログラムの集合体であるため、どこかに問題が起きてもその一部を置き換えるだけでよいため、将来に渡って環境を大きく変えずに済む点が利点といえる。本稿が最善最強のメイルサーバ構築の一助となることを望む。

参考文献

[1]Simple Mail Transfer Protocol; Netowork Working Group, Request for Comments-RFC2821

[2]The Sendmail Consortium; Sendmail;
http://www.sendmail.org/
[3]qmail: Second most popular MTA on the Internet; D. J. Bernstein
http://www.qmail.org/top.html
[4]The Postfix; Wietse Zweitze Venema;
http://www.postfix.org/
[5]netqmail; Charles Cazabon, Dave Sill, Henning Brauer, Peter Samuel, and Russell Nelson; http://www.qmail.org/netqmail/
[6]ucspi-tcp; D. J. Bernstein
http://cr.yp.to/ucspi-tcp.html
[7]qmail-smtpd-auth; Elysium deeZine;
http://members.elysium.pl/brush/qmail-smtpd-auth/
[8]netqmail SMTP-AUTH with qmapop schema; HIROSE, Yuuji;
http://www.gentei.org/~yuuji/software/qmapop-smtp-auth/
[9] cmd5checkpw; Elysium deeZine;
http://members.elysium.pl/brush/cmd5checkpw/index.html
[10]Anti BAD Mail; HIROSE, Yuuji;
http://www.gentei.org/~yuuji/software/antibadmail/
[11]spam対策に特化したSMTP wrapperの実装と検証；広瀬雄二；平成16年度情報処理学会分散システム/インターネット運用技術研究会研究報告
[12]メールサーバーで行なうスパム対策；広瀬雄二；ソフトバンクパブリッシング UNIXUSER 2004年11月号第1特集Part3
[13]UW-IMAPD extension; HIROSE, Yuuji;
http://www.gentei.org/~yuuji/software/imapext/
[14]daemontools; D. J. Bernstein
http://cr.yp.to/daemontools.html
[15]daemontools/tcpserverによるデーモン管理；広瀬雄二；ソフトバンクパブリッシングUNIXUSER 2002年7月号第1特集Part2
[16]Fefe's patches for ucspi-tcp; Felix von Leitner;
http://www.fefe.de/ucspi/
[17]UW IMAP software・IMAP Information Center; University of Washington;
http://www.washington.edu/imap/