透過型電子メールチェッカの導入に係る諸問題

三　谷　和　史

1　はじめに

電子メールはInternet以前に作られた古いアプリケーションであり，Internetの商用化以前の性善説に基づくため，現状では様々な問題点を抱えている。その一つがUCE（Unsolicited Commercial Email）やUBE（Unsolicited Bulk Email）と呼ばれる迷惑メール，いわゆるspamである。電子メールは差出人の名前を自由に名乗れることが事実上の標準であるため，迷惑メール送信者は差出人を詐称して迷惑メールを出すことが可能であり，迷惑メール送信者を特定して迷惑メールを止めさせることは難しい。現在，この迷惑メールのトラフィックが電子メールのトラフィック大部分を占めるようになってきており，ひいてはInternet全体のトラフィックをも圧迫している。さらに迷惑メールは，メールサーバの処理能力を無駄に消費し，メールスプールを圧迫し，ユーザの電子メール処理時間を無駄遣いさせている。また，これら迷惑メールの中にはウィルスメールと呼ばれるコンピュータウィルスが添付されたものもあり，十分な注意が必要である。コンピュータウィルスに感染した機器が組織内に発生した場合，組織内部，外部に対して感染を拡大し，それに伴いトラフィックが増大，また，暴露系のウィルスの場合は組織の機密が漏洩されて，組織自身に対する信頼の低下を招く。

一般に組織のセキュリティレベルはセキュリティが一番低い箇所のレベルとなる。組織のセキュリティレベル向上のためには，組織に存在するコンピュー

タ等の全ての機器に対し，適宜OSのパッチを当てる，ウィルス検出・削除ソフトウェアのパターンファイルを最新に更新する，セキュリティ的に問題が見つかったソフトウェアの更新を行うといった煩瑣な作業が必要となる。しかし，組織によってはそれを徹底することが難しく，結果として組織のセキュリティレベルを下げる結果となっていることも少なくない。

コンピュータウィルスの感染経路の1つが電子メールである。透過型電子メールチェッカは，ユーザになんら新たな設定を求めずして，組織に入る電子メール，組織から出る電子メールを一端横取りし，組織のポリシーに沿ったチェック，例えば出て行くメールの差出人が組織のアドレスであるか，入って来たメールが組織宛であるか等を行い，コンピュータウィルスであれば電子メールを削除し，迷惑メールであれば電子メールに警告を付けたり削除するといった動作を行う。これによって組織のセキュリティレベルの水準を保つことが可能となる。

しかし，透過型電子メールチェッカの導入によって引き起こされる問題が存在することも事実であり，またその問題は通常は表面化せず，ある条件の下で起こるものもあるため，導入時には組織内で電子メールの配送に関っている部署との調整が必要であることを述べる。

2　電子メールの配送方法

本節では電子メールの配送方法について，国内の歴史的経緯を踏まえて述べる。

2.1　過去の電子メール配送

Internet以前の国内の広域のネットワークは電話線とモデムを使ったものが主に使われており，1984年に始まったJUNET（Japan UNIX/University NETwork）がその代表であった。そこでの情報伝達は，UUCP（Unix-to-Unix CoPy）を使って行われており，電子メールとネットニュースが主たるア

プリケーションであった。JUNET での電子メールの配送は木構造に基づくものであり，各組織はこの木構造のノードもしくはリーフとして存在する。あるノードは自分の下位にあるノード及びリーフの全てを把握していて，それら組織宛の電子メールを下位のノードもしくはリーフに適切に送り出し，それ以外の電子メールは自分の上位のノードに送り出す。これによって国内の電子メールが配送されていた。JUNET では最上位のノード，木の根に当る部分を ccut という名前の東大大計センターのノードが担っていた。道内の組織の最上位ノードは北大情報工学科の wsclark というノードが担っており，その上位ノードは nttlab という名前の NTT 通研に置かれたノードであった。当時北大の wsclark では，道内の組織の全てを把握していて，各組織宛の電子メールを UUCP で送り出しており，本学もそのリーフの一つであった。

2．2　DNS に基づく電子メール配送

Internet が国内で普及しだしたのは，WIDE（Widely Integrated Distributed Environment） Project による1986年からであり，まずは教育研究用という AUP（Acceptable Use Policy）の基で動き出し，1988年には JAIN（Japan Academic Inter-University Network）がX．25等を使って国内のバックボーンの一翼を担った。国内で民間プロバイダがサービスを始めて，一般に商用利用が可能となったのは1992 年からである。

Internet 上での電子メールの配送は，DNS (Domain Name System)［2，3］に記された MX（Mail eXchange） レコードを参照して適切な MTA（Message Transfer Agent）へ直接配送するという形をとる。

MX には電子メールを受け取るノードの名前と共に preference という数値が書かれており，それが小さいものから順に直接配送が試みられる。

例えば，表 1 のような記述がある場合，example.jp 宛のメールは MX の preference 値が10 と最も小さい mail1.example.jp に対して配送が試みられ，何らかの原因でこの配送できなければ，次に値が20と小さい mail2.example.jp に対して配送を試みる。それでも配送できなければ，メールは送信側 MTA

表1：DNS の記述例

mail1.example.jp.	IN	A	10.0.0.1
mail2.example.jp.	IN	A	10.0.0.2
example.jp.	IN	MX 10	mail1.example.jp.
example.jp.	IN	MX 20	mail1.example.jp.

のキューに溜められ，送信側の設定に従った時間間隔で再配送が行われる。そして，送信側の設定に従ってある時間内に配送が完了しない場合，メールは廃棄される。

しかし，ネットワーク全体が一時に Internet に移行したわけではなく，UUCP と混在した時期が暫く続いた。

また，Internet に移行した後も，DNS の設定ミスや再配送の間隔を考慮して，UUCP 的上位ノードが UUCP による配送を止めた後も当該組織の MX に UUCP 的上位ノード自身を入れて貰い，受けたメールを IP で直接相手組織のメールサーバ（MTA）に向けて送るという設定を行っている場合があった。それ以外にも，自分の部署以外は組織内部の配送用 MTA に向けて送るだけという設定で動いている MTA も多く存在した。

表 2 のような記述がある場合，example.jp 宛の電子メールは mail1.

表2：組織外にも MX がある DNS の記述例

mail1.example.jp.	IN	A	10.0.0.1
mail2.example.jp.	IN	A	10.0.0.2
example.jp.	IN	MX 10	mail1.example.jp.
example.jp.	IN	MX 20	mail1.example.jp.
example.jp.	IN	MX 30	friend.example.com.
friend.example.com.	IN	A	10.10.10.10

example.jp 及び mail2.example.jp に配送出来なかった場合は，friend.example.com という別の組織に配送される。friend.example.com では，受け取った電子メールを自身でスプールすることも可能であるし，example.jp のしかるべきサーバに対して直接配送することもあり得る。

2.3 電子メール配送に関るプロセス

電子メールが作成されて相手に届き読まれるまでに関係するプロセスは，MUA（Message User Agent），MSA（Message Submission Agent），MTA（Message Transfer Agent），MDA（Message Delivery Agent）である。MSA と MTA は同じ場合もあり，以下区別せずに使うこともある。

MUA は Microsoft Outlook や Thunderbird といった，ユーザが電子メールを読み書きするプログラムである。MUA からユーザが電子メールを送る場合，SMTP（Simple Mail Transfer Protocol）［1］によって MSA に対して submission を行う。MUA は自身が DNS を引いて宛先に直接電子メールを送らずに，指定された MSA に一度送ることにより，宛先が受け取れない場合にキュー溜めておいて再配送を試み，一定期間配送されない場合は破棄するといった仕事を MSA に任せることができ，また，送信ドメイン認証［8，9］を使っている場合は正しい MS（T）A からのメールであることを示すことができる。

MSA は DNS MX に基づき，もしくは設定によっては直接に指定された MTA に SMTP により電子メールを配送する。そして，最終宛先ノードまで電子メールが届けられた時点で，電子メールは MDA に渡される。

尚，DNS MX で指定された宛先のノードは，自身に宛てられた電子メールを受け取るとそれを最終目的地として処理しなくてはならない。このノードが更に DNS MX を使って配送することは配送のループを招く可能性があるので禁止されている。

MDA は通常以下のような動作を行う。

スプール（spool）

受け取ったノードのファイルに閲覧に供するために電子メールを貯めておくことをスプールするといい，貯められたファイルを spool という。“/var/mail/ ユーザ名”といったユーザ毎のファイルが使われる。spool から MUA が電子メールを取り出すためには，POP3又は IMAP4といったプロトコルが用いられることが多い。また，MUA が spool のあるノードで動作するのであれば，直接 spool を取り込むこともある。初期の MUA は直接 spool から取り込むだけの物であったため，ユーザのノードが MTA となり，その管理が杜撰で無制限に relay を許す設定であったために，spammer のターゲットとされたこともあった。

転送（forward）

宛先ノードで電子メールの宛先を別の宛先に書き換えることがある。これを転送（forward）と呼ぶ。この場合，スプールしつつ転送するということも可能である。通常 MDA の上の MTA に対して submission を行い転送する。

Mailing List or program

Mailing List ドライバと呼ばれるプログラムに電子メールを処理させ，結果その Mailing List に登録された人々に電子メールを配ることが行われる。通常送られてきた電子メールに何らかの加工を行うことが多い。その他，様々なプログラム，例えばウィルスチェッカや procmail 等の電子メール振り分プログラム等に電子メールの処理が任される。

何れにしても，処理の結果新たな電子メールが submission されたり，指定のファイルに電子メールが spool されたり，場合によっては削除され，電子メールは消費される。

図1に電子メールの配送の様子を示す。

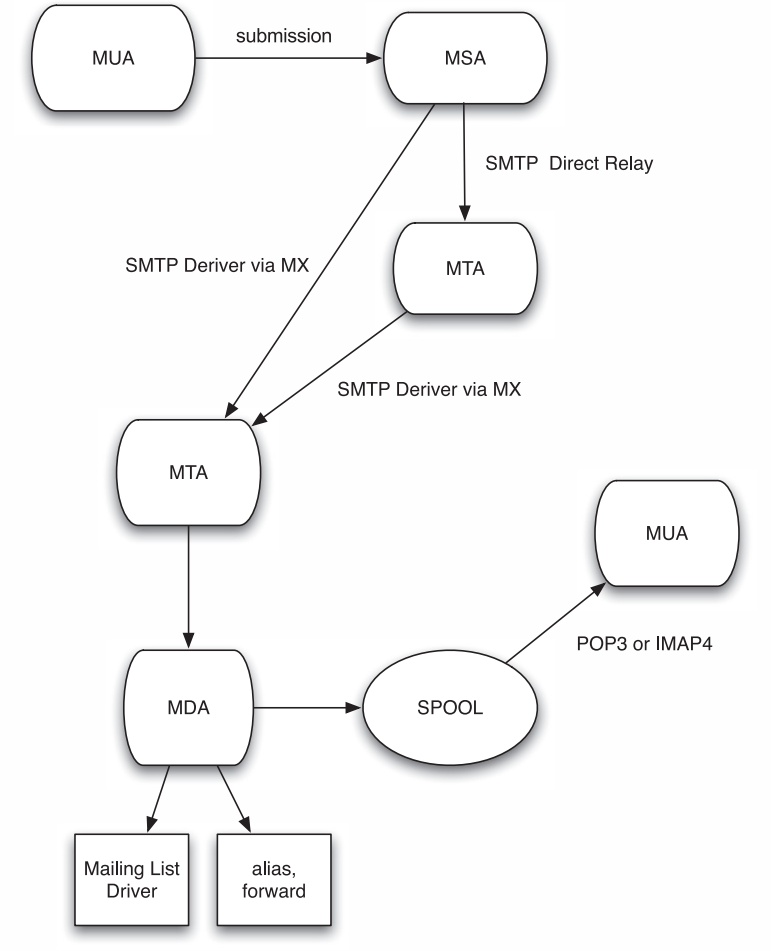

図１：電子メールの配送の様子

3　透過型電子メールチェッカ導入に伴い引き起こされる障害

3.1　透過型電子メールチェッカとは

組織内のスイッチと協調して，NAT（Network Address Translator）と同様にSMTPのコネクションの相手先アドレスを電子メールチェッカのアドレ

スに書き換えて，電子メールの送信先MTAのふりをして電子メールを受け取り，様々なチェックを行った後，その電子メールを再び送信するという動作を行うのが，透過型電子メールチェッカである。スイッチと一体化した形態も可能である。透過型電子メールチェッカは，電子メールの出し元のMUA等に何ら特別な設定をせずに組織に出入りする電子メールをチェックし，ウィルスや組織のポリシーに反する電子メールを排除することが可能であり，組織のセキュリティレベルを一定に保つために導入されることが多い。

チェックには，電子メールの差出人，受取人のアドレスのチェックや，電子メールの本文がウィルスやspamではないかのチェックが含まれる。

また，チェックが済んだ電子メールは，通常のDNS MXに基づく配送を行うMTAに対してsubmissionされる。このMTAは電子メールチェッカと同一ノードで動作していてもよいし，他のノードであっても構わないが，チェック後の電子メールの配送を行うMTAからのコネクションはスイッチによって宛先を書き換えられてはいけない。

3.2　コネクションの横取り方法

TCP［4］のコネクションを表す5つ組を，（SRC，SPORT，DST，DPORT，PROTO）で表す。中身は順にSource IP address, Source port, Destination IP address, Destination port, Protocolである。通常MTAは25番ポートでSMTPのコネクションを受け取るべく待っている。IP addressがaのノードAがIP addressがbのノードBに対してSMTPでメッセージを送る場合，（a,x,b,25,TCP）というコネクションがノードAから見て張られることになる。“x”は通常OSに任される適当な値となる。このコネクションがIP addressがcの電子メールチェッカノードCに透過的に向けられるとは，組織内のスイッチでノードBに向かうコネクション（a,x,b,25,TCP）のDSTを（a,x,c,25,TCP）に書換えてやり，ノードCからノードAに戻るコネクション（c,25,a,x,TCP）のSRCをそのスイッチで（b,25,a,x,TCP）に書き戻すことである。つまり，NATと同じ動作であり，場合によってはport “x” も適

宜書換えることも可能である。

例として，図 2 に筆者のドメインの MTA に対して SMTP コネクションを張りにいった時のセッションを，図 3 に同じ MTA に対して北大内部から SMTP コネクションを張りにいった時のセッションの開始時のメッセージを示す。尚，アドレスやホスト名は一部書換えてある。これを見ると220から始まるコネクション開始時のメッセージが異なる。図 2 は MTA に直接接続されているが，図 3 では MTA と異なる別のホスト，この場合は mailgate3がコネクションを横取りしている。コネクションの横取りを曲げると表現することもある。

```
$ telnet YYYY.mit-s.otaru-uc.ac.jp smtp
Trying 150.83.XX.XX...
Connected to YYYY.mit-s.otaru-uc.ac.jp.
Escape character is '^]'.
220 mit-s.otaru-uc.ac.jp ESMTP Sendmail... 以下略
```

図 2 ：透過型電子メールチェッカが入らないコネクション

```
% telnet YYYY.mit-s.otaru-uc.ac.jp smtp
Trying 150.83.XX.XX...
Connected to YYYY.mit-s.otaru-uc.ac.jp.
Escape character is '^]'.
220 mailgate3.sys.hokudai.ac.jp ESMTP Welcome to ... 以下略
```

図 3 ：透過型電子メールチェッカに横取りされたコネクション

3.3 透過型電子メールチェッカが引き起こす障害

1 つの MTA を中心に考えて，そこにやって来る（Inbound）コネクション，そこから出て行く（Outbound）コネクションに対して透過型電子メールチェッカが働いた場合に，どのような問題が起こるがについて考える。

コネクションの種類としては, submission, DNS MX に基づくコネクション，DNS MX に基づかない Direct なコネクションの 3 つを考える。透過型電子

メールチェッカの導入以前には，これらのコネクションは正常に動作していたものとする。

Submission - Inbound

MS(T)A が submission を許可するポリシーと透過型電子メールチェッカのポリシーの違いが問題となる。submission を許さない場合は MUA から電子メールを送信できなくなる。通常このような場合は，通常使われる25番ポート以外の submission 用のポートを使って送信する必要がある。また，最近はプロバイダ等が spam 送信への対策として OB25（Outbound Port25 Blocking）の設定を行ってきており，直接外部の MSA に25番ポートで submission を行うことが難しくなってきている。そのような場合は通常そのプロバイダ用の submission ポートが用意されており，このポートの使用には通常認証を要する。

DNS MX - Inbound

組織内部に組織外のドメインの電子メールを受け取るノードが存在する場合，宛先ドメインのチェックをされ，組織外宛メールと認定されてメール受信を拒否される可能性がある。チェックが通れば DNS MX に基づく配送なので，予定されたノードに届く。しかし，MX で指定されたノードが停止している場合でも，電子メールチェッカはそのノード宛の電子メールを受け取ってしまい，チェック後に更に別の MX に指定されたノードに送るという場合もありうる。

DNS MX - Outbound

送信，受信ドメインのチェックで電子メールを拒否される可能性がある。チェックが通った後は DNS MX に基づく配送なので，通常通り相手に届く。また Inbound の場合と同様に，送り先のノードが停止している場合でも，電子メールチェッカが横取りしてしまうため，送り元のノードから電子メールは出て行ってしまう。

Direct - Inbound/Outbound

Direct なコネクションを透過型電子メールチェッカで横取りされた場合，チェックの結果正しいと認定されたとしても，再配送を行う MTA が DNS MX に基づくため，本来意図されていた宛先には届かない。本来意図されていた宛先はスイッチのアドレス書き換えテーブルの中にあり，電子メールチェッカや再配送用の MTA には判らないからである。これが障害の本質である。

Direct なコネクションを横取りされても障害が発生しない場合は，自身では直接配送を行わずに配送を依頼するための MTA へ張られた Direct コネクション等が考えられる。これは配送を依頼する MTA が電子メールチェッカのチェック後に使われる MTA に変わるだけであるからである。

それ以外の場合は障害が発生する。チェックの結果が正しいと認定されて MTA が DNS MX に基づいてメールの配送を行った場合，どのような事態が起こるかを次に考えてみる。

Direct なコネクションでの障害

ここでは Direct なコネクションによって DNS MX で指定されていないノードに電子メールを転送している場合を考える。この場合，送り元の MTA がその電子メールの宛先の DNS MX に指定されている場合と，指定されていない場合が考えられる。

指定がない場合の 1 つは，そもそも DNS MX に基づく配送を考えておらず，DNS 上で A も MX も設定されていない場合である。この場合は，Direct コネクションを横取りされた結果，次の MTA による転送が相手を見つけられずに失敗に終わる。指定がない場合のもう 1 つは，DNS MX の設定が別にある場合であって，これは横取り後の MTA が DNS MX に基づいて配送を行うが，それが成功したとしても，それは最初の意図とは異なる結果であろうからやはり障害となる。しかし，このような場合はあまり多くないであろう。

次に送り元の MTA がその電子メールの DNS MX に指定されている場合を考える。MTA 自身が DNS MX で指定されたノードであるから，そのメール

を更に DNS MX に基づいて送信してはいけないので，DNS MX ではなく Direct に配送を行っているセッションを横取りされたという状況である。しかし，横取り後の配送は DNS MX に基づいて行われるので，当然元の意図とは異なる所に向けて送られるが，それが自分に戻ってきてしまうと電子メールのループが発生する。

電子メールには，届かない電子メールを永久に配送し続けることを避けるために，一定回数以上 MTA による送信が行われた場合はループであると判断して電子メールを削除する機能が備わっている。もしこのような仕組がなければ，MTA は永遠に受け手に届かない電子メールを送り続けて資源を使い尽くしてしまうであろう。

そして，今述べたようなループが発生すると，この仕組によって電子メールは削除されて相手に届かない。

図 4 に電子メールがループする例を示す。この例は，MX に基づきノード A

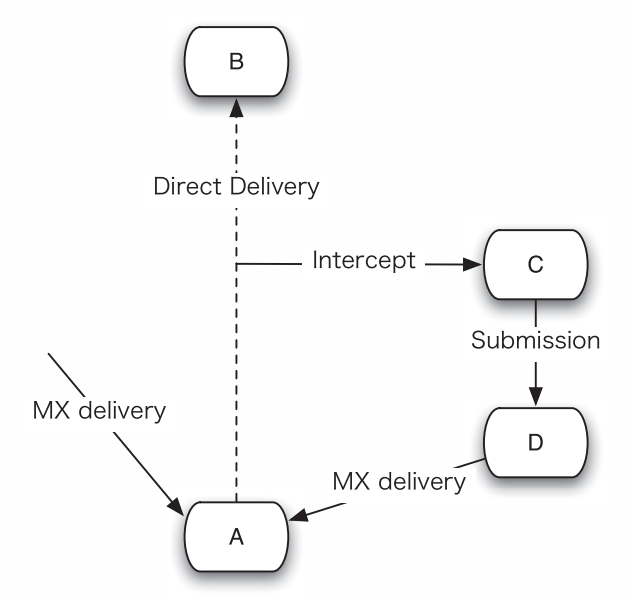

図 4 ：電子メールがループする様子

に送られた電子メールをノードAからノードBに対してDirectなコネクションにより配送していたものが横取りされて，ノードCが電子メールのチェックを行い，チェック後にノードDがDNS MXに基づく配送を行ったところ，又ノードAに対して配送してしまった所を示している。このメールをノードAはノードBに配送しようとしてループとなる。

このようなケースはまれではあるが，実際に起こりうる。通常はDNS MXで指定された上位ノードが停止や不通となることは少なく，組織の法定点検等で停止したり，ネットワークが不通となった場合に，外部にDNS MXで指定されたノードがあり，そのノードが実はDirectな配送を行っている場合に，その外部組織に透過型の電子メールチェッカが導入されると表面化する。

4　障害回避の方法

4.1　DNS MXに基づくコネクションの障害回避の方法

Inboundのコネクションを電子メールチェッカに横取りされる場合，外部ドメインの受け取りをする等，チェッカが拒否する場合は，チェッカに受け取りを許可するドメインに追加をしてもらい,拒否されることを避ける必要がある。また，このドメインのメンテナンスが煩瑣であって，受け取り側が別にメールチェック等を行っているのであれば，コネクションを横取りせずに内部の受け取りノードまで直接メールを通すのも一つの考えである。Outboundのコネクション場合は，チェッカで許可されればDNS MXに基づいて配送されるので問題はないが，差出人アドレスのチェックで拒否される場合は許可するべく設定をしてもらう必要がある。

このような例としては，組織外のドメインのDNS MXに指定されているノードが組織内にある場合がある。具体的には，学会のドメインのノードを学内にもつ大学等が考えられる。

4.2　Direct なコネクションでの障害回避の方法

25番ポートを使ってのSMTPコネクションが透過型電子メールチェッカに横取りされて障害は発生する。これを回避するには，チェッカ側に曲げているスイッチ等で横取りをしないように設定してもらう方法があるが，常に可能であるとは限らない。そこで，25番ポート以外を使ってSMTPを行うことを考える。当然のことながら，受信側がこれに対応していないとこの方法は使えないが，Directなコネクションで配送する間柄の相手であるから，その程度の手間をかけることは許されよう。

但し，これが組織のポリシーと相反する場合は，変更したポートに対してブロックが行われたり，電子メールチェッカ側に曲げられたりする可能性もあるので，組織の担当と協議を行って組織としてそのコネクションを認める方向で対処するべきである。そうなれば，場合によっては横取りしないという処置も可能となろう。

以下ではMTAの具体的な例としてsendmail［5，6］を仮定する。この場合SMTPで送信するMailerの設定でポート番号を書くことができ，書かなければデフォルトの25番が使われるので，例えば，

```
Mtcp,   P=[IPC], F=mDFMuXg8, S=17, R=27, A=IPC $h, E=\r\n
Mtcp2,  P=[IPC], F=mDFMuXg8, S=17, R=27, A=IPC $h 10025, E=\r\n
```

のように10025番ポート[1)]を指定したMailer tcp2を宣言し，

```
R$*<@$*example.jp>$*    $#tcp2$@[10.0.0.1]$:$1<@$2exapmle.jp>$3
```

といったルールをsendmail.cfのrule set 0に追加すればexample.jp宛のメールは10.0.0.1のアドレスのホストの10025番ポートに向けて送信される。ここで注意すべきは，宛先のアドレスは10.0.0.1と固定されており，DNSで

1）submission用の587番ポートを使用するのが一般的かもしれないが，それも横取りされる可能性を考えてあえて異なるポートを例として使っている。

MX を問い合わせているわけではないことである。

次に，このメールの受信側であるが，stone［7］等のパケットリピータを用いて10025番ポートを localhost の25番ポートへとリレーする方法が 1 つ考えられる。stone での記述は以下のようになる。

```
stone localhost:smtp 10025
```

他には sendmail.cf に

```
O DaemonPortOption=Port=10025, Name=MTA
```

といった記述を行い，sendmail が直接10025番ポートを受け取るようにしてもよい。

以上のような方法で，25番ポート以外での横取りされない SMTP の Direct なコネクションを張り，電子メールを配送することが可能となる。

5 ま と め

25番ポートを使った SMTP は電子メールの submission，DNS MX に基づく電子メールの配送，Direct なコネクションによる電子メールの配送の 3 つの用途に使われている。submission については専用のポートを使う場合もある。この 3 つを区別せずに透過型電子メールチェッカがコネクションを横取りしてしまうため障害が発生することを指摘した。特に Direct なコネクションが横取りされた場合は，横取り後の動作が意図されたもの異なることとなり，大きな障害となる。

電子メールチェッカによる送信，受信ドメインのチェックで電子メールが拒否される場合は，組織の担当と協議を行い，必要なドメインを認めるように設定してもらう必要がある。

Direct なコネクションの横取りに関する問題の解決方法としては，横取りをされないようにするのが最も簡単である。しかし，それでは透過型電子メー

ルチェッカを導入する意味が半減する。横取りをしないという設定ができない場合は，25番以外のポートを使って横取りされないDirectなコネクションを張るという方法が現実的である。そして，例としてMTAにsendmailを使った場合の，ポートを変更した運用方法について述べた。

謝 辞

著者宛の電子メールがこの現象によってループして落とされていることを最初に報告して頂いた中村素典先生（当時京都大学，現NII）に感謝します。

参考文献

[1] J. Klensin, *Simple Mail Transfer Protocol*, RFC 5321 (Oct. 2008)
http://www.rfc-editor.org/rfc/rfc5321.txt

[2] P. Mockapetris, *Domain Names - Concepts and Facilities*, RFC 1034 (Nov. 1987)
http://www.ietf.org/rfc/rfc1034.txt

[3] P. Mockapetris, *Domain Names - Implementation and Specification*, RFC 1035 (Nov. 1987)
http://www.ietf.org/rfc/rfc1034.txt

[4] J. Postel, *Transmission Control Protocol*, RFC 793 (Sep. 1981)
http://www.ietf.org/rfc/rfc793.txt

[5] http://www.sendmail.org

[6] Bryan Coastales, et al., *Sendmail, 4th ed.*, O'Reilly Media, Inc. (2007)

[7] http://sengoku.blog.klab.org/archives/50267303.html

[8] M. Wong, W. Schlitt, *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*, RFC 4408 (Apr. 2006)
http://www.ietf.org/rfc/rfc4408.txt

[9] E. Allman, et al., *DomainKeys Identi.ed Mail (DKIM) Signatures*, RFC 4871 (May 2007)
http://www.ietf.org/rfc/rfc4871.txt