あなたのコンピュータは大丈夫ですか

–情報システムの安全対策–

只木進一
理工学部知能情報システム学科

1　はじめに

1980 年代始めに、日本では N1 ネットが構築され、ようやく遠隔地の大型計算機センターを相互利用出来るようになりました [1]。それから 10 年後には、佐賀大学を含む各大学がインターネットへの接続を開始しました。今や家庭からの接続を利用する人も多数となり、インターネットは日常的なものになりました。一部の研究者によって利用され始めたコンピュータネットワークが、急速に家庭を含む社会全体に入り込んだため、普通の社会が抱える様々な負の側面がコンピュータネットワークに入り込んで来ました。コンピュータ及びネットワークへの不正攻撃です。

コンピュータへ不正に攻撃を行う不正攻撃者は、利用資格のないコンピュータやネットワークに入り込み、そこを利用するだけでなく、大切な情報を抜き出したり書き換えたり、あるいはそのシステムを破壊します。盗まれる情報には、そのコンピュータ或は同じネットワーク上のコンピュータの利用者情報、パスワード、ホスト情報などが含まれ、不正利用の被害が他のコンピュータやネットワークへ拡大することもあります。

また、不正攻撃者は、一度入り込んだコンピュータを足場に、他のコンピュータやネットワークへ不正攻撃を繰り返します。足場に使われるコンピュータが公的機関のコンピュータのようなもの、あるいは同一組織内であれば、攻撃を受けた側が不正攻撃であることを判断するのが遅れ、被害が拡大する恐れがあります。このように、あるコンピュータが不正攻撃を受けると、単に自分のコンピュータのデータが破壊されるなどの自分の不利益だけでなく、次の攻撃への足場にされることで、不正利用に荷担することになってしまいます。

残念ながら、このような不正利用は、決して稀な現象ではありません [2]。佐賀大学では、毎日のように不正攻撃の事例が報告されています。後述のようなシステムのモニターを行っている組織だけで毎日のように不正攻撃の件数を数えることが出来ます。このような組織は学内ではまだまだ少数であることを考えると、佐賀大学には、毎日のように非常に多数の攻撃が行われていると考えるべきでしょう。もしかすると、既にあなたのコンピュータは、攻撃者の餌食になっていて、攻撃者はそこを足場に学内外への次の不正攻撃を行っているかも知れません。学内のコンピュータが乗っ取られて、学内への不正攻撃の足場となった事例が既に報告されています。

このように、佐賀大学のネットワークには毎日のように不正攻撃が行われ、情報処理センター、ボランティアでシステムを支えている人々、及び各サブネット管理者は、その対応に追われています。甚大な被害を被ったシステムもあり、その管理者 (多くの場合はボランティア) が復旧に貴重な時間を取られました。たとえば、情報処理センターは、1998 年の 11 月末にメールを使った攻撃を受け、約 4 日間メールシステム停止に追い込まれました。

残念ながら、こうした事態にあることへの、全

学的な理解はまだまだ不足しています。システムの安全性への責任ある取り組みが出来なければ、様々な情報化を行っていくことは不可能です。民間企業の場合、ネットワークとシステムの安全性確保の為の専門の部署を置いています。本稿は、システムの安全性に関する簡単な紹介をすることで、全学的にネットワークの安全性への議論が行われ、適切な対応体制を構築することの一助となることを目的としています。

2 不正攻撃の技術的背景

なぜ、インターネットに接続すると不正攻撃を受けるのでしょうか。その理由を知ることによって、不正攻撃を防ぐ手掛かりが得られます。

現在、インターネットに接続されているコンピュータは、大きく二種類に分けられます。一つは UNIX に代表されるような OS を持つコンピュータです。このようなコンピュータは、複数の人が利用することを前提としていて、利用者管理のシステムを OS が有しています。もう一つはパーソナルコンピュータで、個人が個人の目的で一人で利用することを前提とするものです。後者の場合は、利用者管理という概念が脆弱です。

UNIX、とくに BSD 系の UNIX の場合、ネットワークとともに OS を進化させて来ました。コンソール (コンピュータに直接継った画面とキーボート) からの利用も、ネットワークからの利用も、原則として利用者認証が必要になっています。同時に匿名 FTP や HTTP のように広くデータを公開することを目的とする場合には、利用の記録を残すような仕組みが採り入れられています。

このように UNIX の場合、一見、安全なように見えます。しかし、全ての利用の記録が残されるわけではありません。また、開発者が予想出来なかった方法での不正利用の可能性が常に残っています。そのような不具合は次々と発見され、対応策が講じられています。そのため、UNIX のネットワーク機能は、常により安全なものに進化しています。UNIX という OS が小さな機能の集合として構築されているため、管理者は部分的な手直しを繰り返して、システムの安全性を向上していくことが出来ます。

パーソナルコンピュータの場合、ネットワークに継ることが当初は想定されていませんでした。ネットワーク機能が付いてからも、AppleTalk や NetWare 、NetBIOS のように、同一部署内のような小さな組織を想定したネットワーク機能が実装されてきました。同一組織内では、プリンタ共有やファイル共有などが容易に行えることが重要です。その半面で、安全性については、十分に考慮する必要が無いことが前提になりがちです。また、パーソナルコンピュータでは、OS と個々の機能が密接に関連しているため、部分的手直しが困難な側面があります。つまり、パーソナルコンピュータのネットワーク機能は、ネットワークの安全性については、大きな危険をもって出発したと言えます。

パーソナルコンピュータを単なる端末として利用する場合には、受動的にコンピュータウィルスを拾うことが、ネットワーク利用上の危険の主なものでしょう。しかし、そのウィルスが、他のコンピュータに不正攻撃をする可能性があります。パーソナルコンピュータ上で、匿名 FTP や HTTP のサービスをする場合には、利用者認証に仕組みを欠いているため、危険は更に大きくなります。

3 安全性管理の一般論

「インターネットは危険だから使わないようにしましょう」と言うのが本稿の目的ではありません。利用に伴う危険を知って、安全性対策の重要性を知って貰うことが趣旨です。安全性対策の一般論については、いくつかのまとまった文献が利用出来るようになりつつあります [3, 4, 5, 6]。

システムの安全性を守るうえで最も基本的な事柄は、システムがどのようなサービスを提供しているかと、どのように利用されているか、更にそ

れぞれが管理者の意図と一致しているかを知ることです。

例えば大学のネットワークシステムの場合、利用者は大きく分けて学生、教員、事務系職員に分けられます。それぞれの利用形態は大きく異なり、特に扱う情報の安全性に対する要求度が非常に異なります。こうした利用内容の異なる利用者層を抱えるシステム場合、それぞれに応じた利用制限と安全対策を行うべきです。例えば、事務系のネットワークの場合、情報の安全性が重要ですから、外部と交信出来るサービスを整理し、必要のないものは制限すべきです。一方で、教員の中には、ネットワークの可能性そのものの研究を行っている人もいますから、その場合は、出来る限り制限をしないことが必要です。残念ながら、佐賀大学の現状では、こうした整理が行われずにネットワーク利用が始まってしまいました。

個々のコンピュータの場合にも、どのようなネットワークサービスを利用するのか、あるいは提供するかを整理すべきです。例えば、メールサーバをしていない UNIX マシンはメールを受け取らない設定にすべきです。NFS や Windows のファイル共有機能は、不要ならば停止し、必要ならば範囲を限定してサービスすべきです。

上記のように、ネットワーク、利用者及びコンピュータの利用内容を整理した後に、何を管理すべきかを決めます。つまり、サービスごとに利用できる利用者グループ、ネットワークを介して利用出来るサイトなどを指定し、制限をかけます。制限をかけると同時に、その制限が有効に働いていることのモニターも不可欠です。

ネットワークやマシンの管理責任者を明確にすることも必要です。誰がモニターするか、誰がトラブルに対処するのか、対外的責任者は誰かです。佐賀大学の場合、この点も非常に曖昧になっています。極めて小さな組織である情報処理センターや、周囲のコンピュータに詳しそうな人に責任を押しつけていませんか。

4 基本対策

個々のマシンの安全対策についての基本事項をまとめておきましょう。

まず、余計なプロセスは動かさないことが重要です。古い OS の場合は、コンピュータ資源を節約するために、システム起動時に動くプロセスは少数です。しかし、新しい OS の場合は、知らぬ間に多くのネットワークサービスが動いてしまう場合があります。インストールが容易である OS の場合は特に注意が必要です。あなたのマシンで、意図に反して、メールサーバ、NFS やファイル共有、HTTP、さらに匿名 FTP サーバなどが動いていませんか*。

第二に重要なことは、ネットワークからのアクセスの状況を定期的に検査することです。上述のように余計なプロセスと言われても、どれが必要で、どれが不要であるかの判断は容易に出来ないものです。そこで、どのネットワークサービスがどこから利用されているのかを知っておくことがアクセス制限の前に必要です。

どのネットワークサービスが利用されているかの調査は、システムの基本設定では出来ないことがあります。ネットワークサービスプログラムは、記録の残るものを選択しましょう。OS 標準のものに、そのような機能が無ければ、記録機能を有するものに入れ換える必要があります。

5 tcp_wrapper を使ってネットワークサービスの利用状況を見ましょう

具体的に、ネットワークサービスの利用状況を調査したり、その制限をする方法は、OS によって大きく異なります。前述のように UNIX はネットワーク利用と強く連動して OS を進化させて来ま

*例えば、パーソナルコンピュータに簡単にインストール出来る UNIX として Linux がありますが、幾つかの版では、様々なネットワークサーバが勝手にインストールされ、起動されてしまいます。

した。そのため、ネットワーク機能が最も充実しています。ここでは、UNIX のネットワークサービスの利用記録と制限のためのソフトウェアである、tcp_wrapper について簡単に説明します。

本題に入る前に、UNIX の場合のネットワークサービスの仕組みを簡単におさらいしておきましょう。ネットワークサービスを受ける場合、サービスを受ける方 (クライアント) は、サービスを提供する方 (サーバ) に対して、サービスごとに決まったチャンネル (ポート) への接続要求をします。つまりサーバは、常にポートを監視して、接続要求があればその要求に応じた動作をしなければなりません。

要求頻度の高いサービスの場合、サービスプログラムは、システムに常駐してポートの監視を行うべきでしょう。UNIX の場合、メールサーバプログラムの sendmail やファイル共有プログラムの NFS は常駐型のプログラム (daemon) としてシステム起動時に起動されます。しかし、利用頻度の低いものは、常駐してポートの監視をしたのでは、システム資源の無駄です。

UNIX の場合、利用頻度の低いネットワークサービスは、inetd というプログラムが代表してポートを監視し、要求のあったプログラムを、その都度叩き起こすという方式を取っています。例えば、telnet 要求は、一旦 inetd が受け取り、telnet 要求であることを判断して、telnetd へその要求を渡しています。tcp_wrapper は inetd の代わりに動作して、サーバ接続要求の記録と制限を行います。

tcp_wrapper のインストールは簡単です。まず、最新のパッケージを入手してください*。ソースを展開したら

```
make REAL_DAEMON_DIR=/foo/bar sys-type
```

を入力します†。/foo/bar には、telnetd などのネットワークプログラムのあるディレクトリを指定し、sys-type には OS の名前を指定します。例えば FreeBSD の場合は、次のように指定します‡。

```
make REAL_DAEMON_DIR=/usr/libexec freebsd
```

コンパイルが成功すると tcpd というバイナリが生成されます。tcp_wrapper のパッケージにはインストール用の手続きが無いので、tcpd を手動で適当なディレクトリにコピーします。例えば

```
intall -c -m 755 tcpd /usr/local/etc
```

とします§。マニュアル類は、ソースのあるディレクトリにある、数字を拡張子とするファイルです。roff 形式で書いてあるので nroff などを使って読みます。例えば、tcpd のマニュアルを見るには

```
nroff -man tcpd.8|more
```

とします。

次に/etc/inetd.conf を編集します。BSD 系 OS の場合、例えば図 1 のような記述があるはずです。行の最後の部分が実行時のコマンド、その前がプログラム名です。これを図 2 のように編集します。つまり、プログラム名の部分を tcpd と入れ換えます。編集が終ったらシステムを再起動、あるいは

```
kill -HUP pid
```

を実行します。pid は inetd のプロセス ID です。

tcp_wrapper からの出力は SYSLOG、FreeBSD の場合/var/log/maillog、に記録されます。指定の無い場合は mail.debug として記録が残されますが、tcp_wrapper の Makefile を編集することで、他の SYSLOG として記録することも出来ます。SYSLOG に記録される内容は図 3 のような形式で出力されます。

ネットワークサービス利用の記録を残せるようになったら、しばらく利用状況を調べましょう。単に SYSLOG を見るだけでも役立ちますが、適当なフィルターを使うと必要な情報を取り出してみ

*http://www.cert.org/ftp/tools/tcp_wrappers

†単に make と実行することで、パラメタの設定方法を表示することが出来ます。

‡FreeBSD の場合は、packages や ports からインストールすることも出来ます。

§こういう操作は OS ごとに異なります。

図 1: BSD 系 OS の場合の元のinted.conf の記述

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd     ftpd -l
```

図 2: BSD 系 OS の場合の tcp_wrapper を使ったinted.conf の記述。設定を容易に元に戻せるように、元の設定を“#” を使ってコメントアウトし、新たな設定を記述している。

```
#ftp     stream  tcp     nowait  root    /usr/libexec/ftpd     ftpd -l
ftp     stream  tcp     nowait  root    /usr/local/etc/tcpd     ftpd -l
```

ることが出来ます。例えば、perl のような言語を使って集計し、それをメールで定期的に自分宛に送るような仕組みを作ると良いでしょう。

6 tcp_wrapper を使ってネットワークサービスの利用制限をかけましょう

tcp_wrapper は、inetd を介した要求をモニタするだけでなく、その制限をすることも出来ます。制限は/etc/hosts.deny と/etc/hosts.allow で行います。その名の通り、/etc/hosts.deny は利用を拒否する内容を、/etc/hosts.allow は許可する内容を記述します。これらのファイルの記述を見るには、ソースのあるディレクトリで

```
nroff -man hosts_access.5|more
```

とします。

これらの記述の一般的方法は次のようになります。

サービスプログラム名:ドメイン名

サービスプログラム名には、telnetd などの daemon 名を記述します。ドメイン名には、ホスト名またはドメイン名を空白で区切って列挙します。ドメイン名の場合は “.” で始まります。例えば

```
telnetd:.mydomain.com
```

は.mydomain.com ドメインからのアクセスを拒否または許可します。予約語として All は全てのサービスプログラム名または、ドメイン名を表すものとして定義されています。

具体的な設定の例を示しましょう。一般的なコンピュータの場合、想定しているネットワークサービス以外は拒否するような設定が望ましいでしょう。そのような場合、/etc/hosts.deny には

```
All:All
```

と記述します。これで、原則として、全てのネットワークサービスに対して全てのドメインからの利用を拒否します。ネットワークサービス許可するドメインを/etc/hosts.allow に許可します。HTTP やメールなど、特別なサービス専用のコンピュータの場合には、管理用端末からだけの telnet などだけを記述します。それ以外ならば、そのコンピュータの利用目的を吟味して記述します。inetd.conf の変更と異なり、設定変更の度にプログラムの再起動を行う必要はありません。

7 情報処理センターの利用者としての対応

前節では、パーソナルコンピュータの使用者を含めて管理者の安全対策について書きました。次に、情報処理センターに利用資格を持つ利用者としての安全対策について書きます。

一般に、利用者といえども、安全対策に注意し

図 3: tcp_wrapper からの SYSLOG への出力形式

```
日付 時間 マシン名 サービス名 [プロセス ID]: connect from 要求元
```

なくてはなりません。その基本はパスワードの管理です。UNIX システムの場合、利用者名とパスワードによって、利用資格の有無を確認します。パスワードは、利用者一人一人の利用の安全を保護すると共に、利用資格を有しない人による不正攻撃からシステムの安全を守る重要な要素です。

不正攻撃者のパスワード破りの手法は日々進歩しています。利用者名と同じパスワードはもってのほかです。もちろん、パスワードは、暗号化されていて、それを直接復号化して解読することは非常に困難です。そこで、不正攻撃者は、候補となる文字列を暗号化して、暗号化されたパスワードと比較して、パスワードを破ります。候補となる文字列は、辞書やコンピュータ内のファイルから拾われます。つまり、通常の英単語、コンピュータ関係の略語、有名人の名前を使ったパスワードは簡単に破られます。パスワードをコンピュータ内に書いておけば、それから簡単に破られます。

パスワードは、英小文字、英大文字、数字、記号を混ぜたものにしてください。このような文字列は、辞書などから選択することが出来ず、パスワードを破ることが困難になります。また、定期的に変更してください。パスワードが何かの機会に漏れているかも知れません。一方の計算機のパスワードが漏れることによる連鎖的パスワード漏れを防ぐために、異なる計算機には異なるパスワードを設定してください。

こういうパスワードの設定と管理は面倒な事柄です。しかし、簡単なパスワードの設定やいい加減な管理によって、利用者本人だけではなく、同じシステム、同じネットワークを利用する人に被害が広がる可能性があることを留意してください。

また、当然のことながら、パスワードを他人に教えたりしてはいけません。ましては、メールを使ってパスワードを教えることは、不正攻撃を企む人にパスワードを教えているのと同じです。通常のメールの内容を見ることは、不正攻撃者にとっては簡単なことです。

telnet などを使って遠隔利用する際にパスワードを送ることも非常に危険です。送られる文字列は暗号化されずに送られています。パーソナルコンピュータからメールを読む際に利用している POP もパスワードを暗号化せずに送っています。遠隔地の X クライアントを利用している場合には、一つ一つのキー入力がネットワークに流れています。ssh などの暗号化可能な方法を使って遠隔利用を行うべきです。

8 Secure Shell の利用

前節で述べたように、ネットワークを流れる情報は、もともとは暗号化されていませんでした。しかし、ネットワークの利用が社会的に浸透すると同時に、盗聴などの不正行為が起きるようになったため、様々な暗号技術が必要となってきました。通信の暗号化技術の一つが Secure Shell (ssh) です [7]。

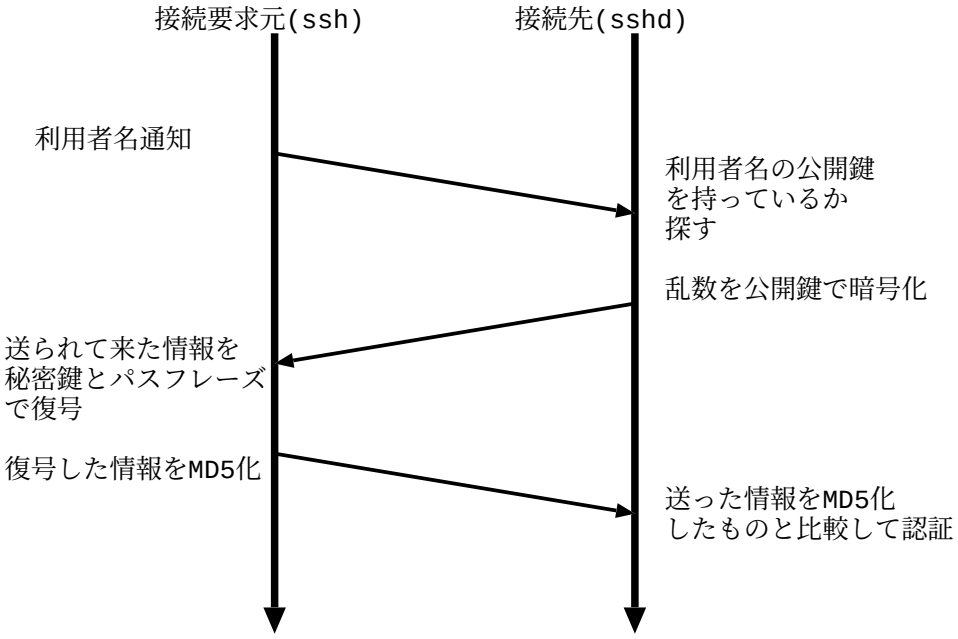
Secure Shell は UNIX が持っている `rlogin` などのリモートコマンドの上位互換のコマンドを提供します。ssh は RSA* と呼ばれる暗号化技術 [8] を使って、利用者認証、ホスト認証、更に通信の暗号化を行う為、通信内容の盗聴を防ぐとともに、アドレスなどの偽装による不正利用を防ぐことが出来ます (図 4)。

ssh のデーモンプロセス (sshd) をインストールすると†、マシン固有の暗号鍵が生成されます。利用者は、ログインする元のマシンで `ssh-keygen`

*RSA はアメリカ合州国からの持ち出しが禁止されています。しかし、ssh は、アメリカ合州国外で開発されたもののため、合州国以外のサイトから入手することは可能です。

†早くセンターにもインストールして下さい。

図 4: ssh を使った認証の概略



によって利用者ごとの暗号鍵を生成します。この時、パスフレーズ、つまり文章のように長い暗証文字列を指定します。この時生成される暗号鍵 (UNIX の場合、`$HOME/.ssh/identity.pub` というファイルに書き込まれる) を sshd の動いているログイン先のマシンの適切な場所 (UNIX の場合、`$HOME/.ssh/authorized_keys` となる) に記入しておくと、

```
ssh ログイン先
```

でログインすることが出来ます。

ssh でログインすると、ログイン先からログイン元の X サーバへの接続も ssh を使って暗号化され、キー入力などを盗聴されることを防ぐことも出来ます。

最近、Windows でも ssh を利用できるようになりつつあります。`Tera Term` と呼ばれる端末ソフトが ssh に対応しているそうです*。

*`http://hp.vector.co.jp/authors/VA002416`

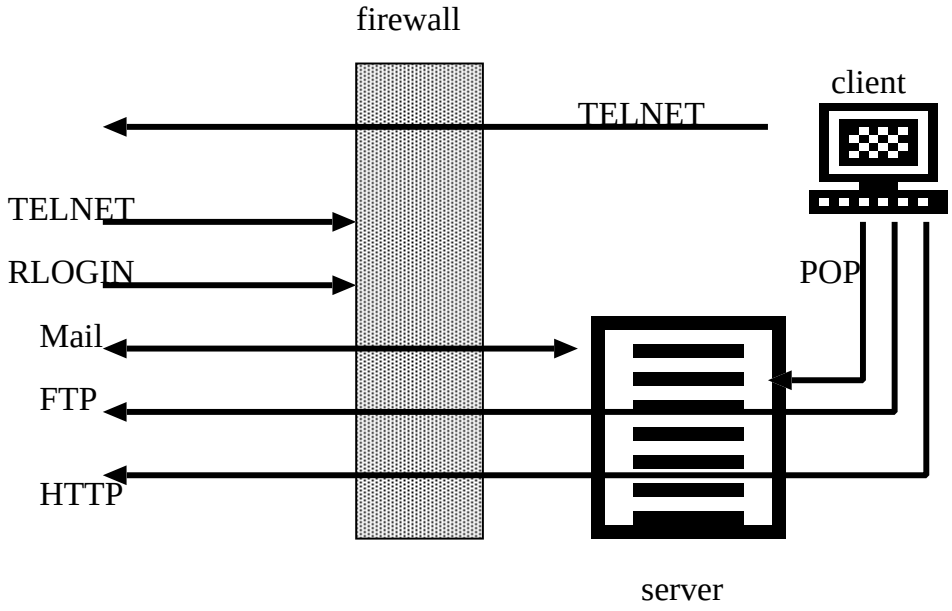
9 ネットワーク組織としての対応

tcp_wrapper での利用制限は、個別のコンピュータごとの安全確保の手法です。では、OS がそういう安全技術に対応出来ない、あるいはコンピュータの管理責任者が安全対策について技術的に対応できない場合はどうしたら良いでしょうか。その対策としては、インターネットから切り離すのが基本的な考え方です。

それでも、なんとかインターネットの資源を利用した場合には、ネットワーク組織として、利用制限を行う方法があります。つまり、外部ネットワークとの中継点での利用制限です。中継点での利用制限の方法として、ファイアウォールや NAT などの方法があります。

組織のネットワークを外部と必要に応じて切り離す仕組みを、防火壁 (ファイアウォール) と呼んでいます。ネットワークを流れる通信は、パケットと呼ばれる小さな単位で行われます。パケットには、それぞれ発信元、受信先、ネットワークサービ

図 5: ファイアウォールでのパケットフィルタリングの例。パーソナルコンピュータ等のクライアントは、外部への telnet は行えるが、外部から telnet や rlogin をすることは出来ない。メールはサーバが受ける。FTP や HTTP はサーバが代表して受信する。このようなネットワークの場合、個々のクライアントはインターネットアドレスを持つ必要はない。



スを区別するポート名と呼ばれる符丁がついています。この部分を解読して、ネットワークサービスの要求元、要求先さらにサービス名ごとに通過を許可あるいは拒否する仕組みを、パケットフィルタリング或は単にファイアウォールと呼びます (図 5)。

一つ一つのコンピュータごとにパケット通過の許可指定を行っていくことは、例え小さな組織でも非常に大きな管理コストを要します。佐賀大学の場合には、サブドメインごとの対応になりますが、通常は、ほぼ全てのコンピュータについて同様な利用制限を掛け、特別な用途のコンピュータに対して個別の詳細指定を行うことが現実的です。どのような運用体制を取ることになっても、コストに関する十分な議論が必要です。同僚の誰かに任せれば良いという安易な考えは禁物です。任された人にとっては、他のことが出来ない程の重荷になります。

佐賀大学の多くの利用者は、センターから POP でメールを読み書きし、WWW を見ているだけではないでしょうか。そのような利用者にとって、個々のコンピュータが固有のインターネットアドレスを有していることは意味がありません。固有のインターネットアドレスを有しているが故に外部から不正利用を招くことがあります。そこで必要な時に、一時的なインターネットアドレスを付与する仕組みが NAT や DHCP などの方法です。DHCP はコンピュータの起動時に一時的アドレスを管理組織から付与する仕組みです。それに対して、NAT は中継点から外部にでる時に一時的アドレスを付与する仕組みです。いずれも、恒久的なインターネットアドレスが付与されないために、外部からの攻撃が困難になります。この場合もアドレス管理のコストに関する議論が必要です。

10　おわりに

ネットワークの不正利用は、残念ながら、日常的な現象になってしまいました。不正攻撃者は、一つのコンピュータを乗っ取ると、そこを足場に次のコンピュータを狙います。不正攻撃者の仲間内には、OS やソフトウェアの弱点に関する情報や、管理の甘いネットワークやコンピュータに関する情報が共有されています。佐賀大学では、幾つかの不正攻撃の成功があったため、佐賀大学が攻撃目標として登録されているかも知れません。

こうした不正攻撃者からの攻撃から身を守るには、一人一人の利用者が、個々のコンピュータ及びネットワーク組織の安全性について、関心を持ち、対策を取ることが必要です。全ての皆さんが、まず自分のコンピュータがどのようなサービスをしているかを確認してください。特にパーソナルコンピュータを使っている人は、ファイル共有など余計なサービスをしていないかを調べて下さい。

また、ネットワーク利用を便利さの点だけで考えることは非常に危険です。便利さと危険が隣合わせであることを知って下さい。例えば、利用者認証無しに使えるということは、不正攻撃者も同様に入ることが出来るということです。多くのネットワークプロトコルは、通信内容をそのまま送っています。そういう通信内容を傍受することは、それほど難しくありません。様々なプロトコルが暗号化通信を採り入れていますが、まだ完全ではありません。

安全管理に関する組織としての議論も必要です。各部局ごとに、そして佐賀大学全体として、安全管理と責任体制を早急につくるべきです。ボランティアを中心とするネットワーク管理者に、こうした問題を押しつけたままにした場合、外部から責任を問われた場合に、対応が出来ません。また、直接管理に携わる人々だけで管理方針を議論すれば、原則利用禁止という方向になる恐れもあります。全学的な議論を期待します。

参考文献

[1] 石田晴久「コンピュータ・ネットワーク」(岩波新書、1991).

[2] 様々なネットワーク攻撃の情報と対策の助言を、「コンピュータ緊急対応センター」(http://www.jpcert.or.jp) から得ることが出来ます。

[3] 「情報処理振興事業協会 (IPA) セキュリティーセンター」
http://www.ipa.go.jp/SECURITY/index-j.html

[4] 宝木和夫、小泉稔、寺田真敏、萱島信「ファイアウォール」(昭晃堂, 1998).

[5] A. E. Hutt, S. Bosworth and D. B. Hoyt ed. *Computer Security Handbook 3rd ed.*, (John Wiley & Sons, 1995).

[6] D. Russell and G. T. Gangemi Sr. *Computer Security Basics* (O'Reilly & Associates, 1994).

[7] Secure Shell とその利用に関して、次の解説がある。島慶一「Secure Shell」UNIX マガジン 1998 年 6 月号, 41; 1998 年 7 月号, 62; 1998 年 8 月号, 47.

[8] R. L. Rivest, A. Shamir and L. Adelman, Comm. ACM **21** (1978) 120.