# サーバ管理者のためのセキュリティ

総合情報処理センター
鶴　正人
tsuru@net.nagasaki-u.ac.jp

## 1　はじめに

学内で“サーバ”計算機を管理する人がネットワークセキュリティに関して最低限知っておくべき基本事項を述べる。“サーバ”は、他の計算機に何らかのサービスをネットワーク経由で提供するものであり、それゆえ、“不正アクセス／侵入”の対象となりやすい。歴史的に有名な事件(大規模な侵入)としては、“Internet Worm”事件と“Network Monitoring Attack”事件が知られている。

1988年に起きた“Internet Worm”は、“The Great Worm”とも呼ばれ、UNIXのセキュリティホールを突いて、ワームを潜り込ませた。そのワームはインターネット上の2000台以上の計算機に侵入し、(それ自身のバグで)意図せずして計算機の資源を食い潰して、それらの計算機を麻痺させた。これをきっかけに、CERT/CCなどのネットワークセキュリティ専門ボランティア組織が誕生した[†1]。

1994年に起きた“Network Monitoring Attack”は、UNIXのセキュリティホールを突いて、パスワードファイルを盗み、それを解読して利用者のパスワードを得て不正侵入(遠隔ログイン)し、そこから、様々な手口でroot権限を得た。root権限を使えば、自分の痕跡を消すことや、接続しているネットワークの盗聴が可能であり、そのネットワーク上を通過するパスワードが入手できるので、それを使ってさらに別の計算機に侵入した。このような手口を巧妙、大規模に繰り返し、全世界で10万人分以上のパスワードを盗んだと言われている[†2]。

さて、これらの事件やその後のインターネット上での侵入事件から得られた教訓によれば、

1. 個々の利用者にパスワードの管理を徹底させる。
2. 不正アクセスを監視／記録し、また予想されるものは未然に防げるような仕組を入れる。
3. ファイル／ディレクトリのアクセス権設定に十分注意する。
4. 個々のアプリケーションの運用において設定ミスやセキュリティホールに十分注意する。
5. ファイアウォールを用いて組織の境界で基本的な防御を行うことが有効である。

などが重要であるが、以下、1, 2, 4. について簡単に説明する。具体的な説明の大部分はUNIXを前提にしているが、原理的には他のOSでも対応する。UNIX以外のOS(例えば、WindowsNTのサーバが最近増えているが)の場合、インターネットでの利用の歴史が浅く、かつ情報が必ずしもオープンでない分、設定漏れや抜穴の危険性はむしろ高いと言えるので、十分な注意が必要である。

## 2　パスワード管理の徹底

rootアカウント(特権利用者)にパスワードを付けていないようなでたらめな管理者は事故があったとき必ず責任を問われるであろう。root権限でなくともパスワードなしの(または簡単なパスワードの)guestアカウントを作ることも論外である。

---

[†1] http://www.cert.org/research/JHThesis/Chapter3.html

[†2] http://www.cert.org/advisories/CA-94.01.ongoing.network.monitoring.attacks.html

各利用者につける初期パスワードは推測しにくいものにし、また、時々、パスワードを変更させるべきである (簡単なパスワードには変更できないように制限した上で)。

"推測しにくい" ためには、ログイン名に似ているなどは論外で、その他の個人情報とも関連を持たせない。銀行のキャッシュカードの暗証番号の場合も、生年月日や電話番号は使わないで下さい、と警告している。そういう安易な番号を使ったキャッシュカードを拾われて悪用された場合、法的にも不利になる可能性がある、という話も聞くが、それと同様である。また、辞書に載っている単語を使っていたりすると、世の中に出回っている "パスワードを推測するプログラム" による高速な繰り返し試行で、探り当てられてしまう。逆にこのプログラム[†3] にかけてみて安全かどうかを確かめることができる。

さらに根本的には、UNIX の "8 文字以内の文字列としてのパスワード" は、計算機の能力が上がった現在、機械的な全数探索によって解かれてしまう。つまり、もし、パスワードファイルが盗まれたら、(相手が本気なら) パスワードは root も含め全部解読されたと思った方がよい。その場合は、即座に計算機をネットワークから外し、OS から再インストールし、全パスワードを変更するしかない。

一方、最終的にパスワードを管理するのは個々の利用者なので、利用者の意識向上 (教育) も重要である。

## 3 不正アクセスの監視及び予防

不正アクセス (不正な遠隔ログイン／データアクセス／運用妨害) を監視したり、アクセス制限をかけたりするには、(a) 各アプリケーションの持つ機能を使う場合、(b) 各アプリケーションを起動する共通親プログラムの機能を使う場合、(c) ネットワークレベルの機能を使う場合、が考えられる。ここでは、(a) の機能が不十分な場合や、そうでなくても、(a) の保険としても必要となる (b) のプログラムとして、**xinetd** を紹介する[†4] 。

**xinetd** は、UNIX 標準のスーパーデーモン (ネットワークアプリケーションを起動する共通親プログラム) である **inetd** を置き換えて使う。基本的に、inetd は、ネットワークからサービス要求 (tcp や udp のポート番号で指定される) が来たら、対応するサーバプログラムを起動するものだが、xinetd は、それにアクセス制限とアクセスログ (記録) の機能が付け加わっている。

サービス要求と起動すべきプログラムの対応や、制限やログの詳細条件などは、**/etc/xinetd.conf** というファイルに記述しておく。このファイルは、inetd が使う **/etc/inetd.conf** とは形式が違うので、新規に inetd から xinetd に切り替る場合は、まずのこファイルを用意する必要がある。

**xinetd.conf** の基本的な形式は[†5] 、最初に defaults という名前で、全体の標準的な値を { } で括って、数行にわたって記述する。例えば、

```
defaults
{
        instances       = 8
        log_type        = SYSLOG auth notice
        log_on_success  = HOST PID
        log_on_failure  = HOST
        only_from       = 127.0.0.1
        disabled        = tftp whois rquotad rstatd rusersd
        disabled        = sprayd walld shell login talk exec
}
```

[†3] Cracklib や password+ などのソフトがある。

[†4] 同様の機能を持つものに、tcp_wrapper というプログラムもある。

[†5] 詳しくは、xinetd.conf の man を参照。

instancesは、そのサービスに関して同時起動数を制限する。log_typeは、ログの方法で、上のSYSLOG auth notice の場合、OS の syslog 機能を使って、auth.noticeという種別で出力する。disabledは、指定したサービスを禁止する。不要なサービスは禁止することが重要である。

その下から、service(アプリケーションプロトコル) 毎に { } で括って、標準と違う項目だけ、数行にわたって記述する。例えば (telnet サービスの記述例)、

```
service telnet
{
        instances       = 16
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/etc/in.telnetd
        log_on_success  = HOST PID USERID DURATION
        only_from       = 1.2.3.4 1.2.3.5
        only_from       += 127.0.0.1
        flags           = IDONLY
}
```

serverは、起動すべきプログラムのファイル名である。上のonly_fromは、1.2.3.4 と 1.2.3.5 と 127.0.0.1 からのアクセスを許可する。only_fromは、サブネット単位でも指定できる。1.2.3.0 と書けば、1.2.3.0 ～ 1.2.3.255 からのアクセスを許可し、0.0.0.0 と書けば、世界中からのアクセスを許可する。

計算機の起動時には、xinetdを自動的に立ち上がるように設定しておく。inetd の代替なので、inetdを起動している場所を探し、そこを xinetdに置き換えればよい。

運用中に設定ファイル (xinetd.conf) の中身を変更した場合は、root 権限で、

```
kill -USR1 ????
```

を実行して、xinetdに通知する (????は動いているxinetdのプロセス番号)。

## 4 個々のネットワークアプリケーションの運用上の注意点

基本は、自分が運用しているアプリケーションの危険なバグ (セキュリティホール) の情報に常に注意し、発見されたらすぐに最新版に入れ替えるなどの対応を取ることである。以下の WWW ページが参考になる。

- CERT/CC(Computer Emergency Response Team/Coordination Center, CMU.)[†6]
- CIAC(Computer Incident Advisory Capability, U.S. Department of Energy)[†7]
- L0pht Heavy Industries Security Advisories[†8]
- JPCERT/CC(Japan Computer Emergency Response Team/Coordination Center)[†9]

各メーカーの提供する OS(Solaris, SGI, FreeBSD, Linux などの UNIX や、WIndowsNT) やそこに含まれるサーバプログラムに関するセキュリティホールの修正版の入手に関しては、学内 LAN のホームページから辿れる“インターネット利用について” のページ[†10] の後半にリンクが集めてある。

[†6] http://www.cert.org/

[†7] http://ciac.llnl.gov/

[†8] http://www.l0pht.com/advisories.html

[†9] http://www.jpcert.or.jp/anm/index.html

[†10] http://www.nagasaki-u.ac.jp/internet/internet-jis.html

また、学内のサーバ管理者間の情報交換のためのメーリングリスト[†11] があるので是非参加して欲しい。

**電子メールの中継**サーバは、SMTPという方式（プロトコル）で実現され、**sendmail**というフリーのプログラムまたはそれにメーカーが手を入れたものがよく使われている。インターネットで最もよく使われるアプリケーションの一つであるが、メールを受取った後、利用者のメールボックスに書込んだり、（メーリングリストがそうであるように）あるプログラムを起動したりする強力な機能を持つので、クラッカの執拗な攻撃に遭い、セキュリティホールに関するイタチごっこが続いている。元々、古き良き時代に設計されたプロトコル／プログラムなので、運用妨害や不正メールの転送の踏台（不正中継）にも遭い易い。しかも、安全な設定は初心者には難しいので、JPCERT/CCから提供されている“技術メモ - sendmail バージョンアップマニュアル -” [†12] を参照して欲しい。また、細かい設定を比較的簡単な構文で書けるツールとして、CF[†13] の使用を勧める。

**電子メールの読み出し**サーバは、POP3とIMAP4という2種類のプロトコルが主流で、フリーのプログラム (**qpopper** と **imap**) やそれらにメーカーが手を入れたものがよく使われているが、セキュリティホールもしばしば発見されている[†14] 。パスワードが裸のままネットワークを流れる問題に対処するには、APOPを使う、SSHとの組み合わせで使う等の工夫が必要である。

**WWW**サーバは、HTTPというプロトコルで実現される。源流は、**NCSA httpd** と **CERN httpd** というフリーのプログラムであったが、現時点では多くの種類がある[†15] 。攻撃対象となるのは、基本的には、CGIと呼ばれる遠隔実行機能と、アクセス制限機能であるが、もう一点重要なのは、サーバのコンテンツ (ページの中身) を多数の人が更新する必要がある場合の、遠隔ログイン／ファイル転送機能の安全性である。HTTPとは直接は関係ないそれらの不備がしばしば危険を招いている[†16] 。

WWWサーバに関する別の問題は、パソコン利用者でも簡単に立ち上げることができるようになった（そういうお手軽なサーバソフトがアッと言う間に普及した）点にもある。一般に急速に広い層に普及すればセキュリティへの認識が低い人も増加することは避けられない。そういう意味で、啓蒙のためのページも多数提供されている[†17] 。

**FTPファイル転送**サーバや**TELNET遠隔ログイン**サーバは、これも古き良き昔の仕組みで、インターネット経由のアクセスには向かなくなってきてる。しかし、UNIX計算機の場合、標準状態で起動すると、これらのサーバが立ち上がってしまい、ましてや、そこにパスワードなしのguestアカウントなどがあったら知らない間に侵入されてしまうので、

- 使う必要がなければ起動しないような設定にする。できるだけ、SSHなどで代用する。
- 使うなら最低限、先に述べた **xinetd** などを使って、IPアドレスでアクセス範囲を限定する。

なお、ftpサーバを使って、anoymous-ftp(匿名 ftp) というサービスを提供することがある。これは依然WWWサーバでは代用できない場合もあるが、運用には十分な注意が必要である。例えば、アップロード系操作 (ファイルの書込み、アクセス権変更、ディレクトリの生成など) は禁止すべきである。クラッカたちに、不法コピーしたプログラムの交換の場などに使われてしまう。また、最近

---

[†11] net-admin@ml.nagasaki-u.ac.jp

[†12] http://www.jpcert.or.jp/tech/98-0001/

[†13] ftp://ftp.kyoto.wide.ad.jp/pub/mail/CF/

[†14] http://www.cert.org/advisories/下の、CA-98.09.imapd.html や CA-98.08.qpopper_vul.html など。

[†15] http://webcompare.internet.com/

[†16] 某国や某社の公式ページが反対分子にクラックされページの中身を書換えられた、という記事をよく目にする。

[†17] 基本は、http://www.w3.org/Security/faq/www-security-faq.html

見つかったセキュリティホール[†18] もこのアップロード系操作によって危険度が増す。

**NFS** によるファイル共有、**NIS** によるパスワードやホスト情報の共有、**rsh/rlogin** などに基づく遠隔処理は、基本的に、信用できる閉じた世界で使うことを前提とした仕組みであり、それゆえ、逆にその弱点を狙ったクラッカの標的でもある。これらの仕組みを使う場合は、その世界 (研究室や学科等) の外からはアクセスできないような制限をかける必要がある。また、信用できる範囲内の計算機が侵入されたら他も全滅する可能性が高いというリスクを認識しておく必要がある。使わないで済むなら使わない[†19] 。

**パソコン系のファイル共有やリモート操作**に関しても同様のことが言える。Windows や Macintosh 標準のもの以外にも、CAP, samba, VCN, pcANYWHERE 等々、いろいろなプロトコルやプログラムが出回っており、特にパソコン／パソコン間のリモート操作やファイル共有は、非常に簡単に設定できるらしい (つまり、全員がサーバ管理者になってしまった !) が、パスワードやアクセス制限の設定を本当に全員がきちんとできているのだろうか? セキュリティホール (バグ) のリスクは認識してるだろうか?

**ダイアルアップリモートアクセス**サーバは、電話回線経由で自宅のパソコンや出張先のノートパソコンを、IP/PPP などの方式で大学のネットワークやインターネットに接続する。原則的には、遠隔ログインと同様に、なんらかの利用者認証が働くが、パスワードが弱いと破られる。歴史的には電話回線経由での不正侵入は非常に多い[†20] 。

特に問題なのは、そのサーバを経由して学内ネットワークにアクセスされると、それは、一般の (正しい) 利用者が自宅からアクセスしているのと、全く区別が付かない点である。ファイアウォールも無力である。そういう意味で、潜在的に非常な危険を孕んだサービスである。

さて、指定したサーバ計算機に対して、このようなセキュリティ上の問題をネットワーク経由で検出／テストするプログラム (ツール) というのも出回っている。設定ミスや漏れはつきものなので、自分の管理しているサーバ計算機をプログラムでチェックすることは有効であるが、一方、これは素人でも簡単に他人の計算機を攻撃できるツールにもなる。しかも最近は、日本語の書物で、この手のネットワークセキュリティ問題検出ツール (= クラッキングツール) が付録の CDROM に入ったものが、良く売れているそうである。

もはや、少くともこの手の本に載る著名な問題点は、すべてクリアしておかないと、クラックされるのは時間の問題ということもできよう。学内でサーバを管理している人には、是非とも自分の計算機の管理状態の再確認をお願いしたいと思う。

また、直接はネットワークセキュリティの問題ではないが、サーバプログラムの多くは明示的に日付を操作するので、西暦 2000 年問題を含む場合が多い。それらの対応版はほとんどのサーバで出揃っているが、それを入手して入れ替えるのはサーバ管理者が行うしかない[†21] 。

---

[†18] `http://www.cert.org/advisories/CA-99-03-FTP-Buffer-Overflows.html`

[†19] `/etc/hosts.equiv`, `/etc/exports`, `/.rhosts` 等は中身を完全に理解できている場合以外は消す。

[†20] 電話番号を部外者に公開しないことが第一歩であるが、範囲を予想して順に番号を試すツールも出回っている。

[†21] お金があれば外注することもできるかも知れないが、、、