# Linux サーバ構築とセキュリティ

荒木 靖宏

((株)インターネットイニシアティブ、日本 Linux 協会(JLA)

1999 年 12 月 14 日
Internet Week99 パシフィコ横浜

(社)日本ネットワークインフォメーションセンター編

# 目次

# 1 概要

この講演では、Linux サーバ構築とセキュリティについて、セキュリティと利便性とのトレードオフを意識した運用ポリシーをもとに、コストに見合った投資を行うという観点から説明します。

最初に、運用ポリシーを明確にするというテーマを説明し、インターネットサーバ構築の概要として、Linux の利点やディストリビューションおよびサーバ向けのパッケージ選択の指針について説明します。サーバとして運用する場合に必須となるカーネルのリコンパイルについてその注意点を説明し、インターネット上でのカーネル関連情報についても紹介します。

実際の運用時のアクセスとサービスの限定方法、また具体的な手段として使用される OTP(One Time Password)や SSH(Secure Shell)、また tcp-wrapper や ipchains などのセキュリティ強化ツールやパケットフィルタリングについても説明します。最後の部分では、内部ユーザおよびネットワークからの攻撃や物理的および電気的な破壊からの防護策や対応策、日常的なシステムの管理としての log 監視の重要性や方法を説明します。

# 2 運用ポリシー

サーバを運用する場合に考慮しなければならないことはたくさんありますが、特に重要なものは運用ポリシーです。マシンの管理者としては一般にスキルが高いと思われる人が選ばれることが多く、「彼にまかせておけば大丈夫」ということで明確なポリシーなしに運用しがちです。運用ポリシーとして、どのような Security Incident をどこまで対策するか、ということを考える必要がありますす。すべては運用ポリシー次第ですから、まず「何をどう守るのか」を明確にする必要があります。たとえばデフォルトの home ディレクトリの umask をいくらにするか、ということもその一つです。

次にサーバ設置の目的をできるだけ明確にして、「危険なサービスに対しては代替手段」を考えなるべくその提供を避けます。そして、「今必要なサービスだけを行う」ということが重要です。たとえば Linux のデフォルトでインストールされてしまう IMAP daemon サービスなどは必ずしも必要なものではありません。

しかしユーザからはそのようなサービスを要求されることもありますので、その場合には不必要なことが十分説明できなければなりません。知識武装をし、セキュリティと利便性のトレードオフを意識して運用ポリ

シーを考えることが必要になります。

# 3 Linux でのインターネットサーバ構築の概要

Linux でサーバを構築するときの最低限の留意点について説明します。

## 3.1 なぜ Linux か

最初に、なぜ Linux を選ぶのかという点について説明します。Linux には以下のような利点がありますが、これらが当てはまらない場合は Linux ではなくサポートのある OS を使う方が良いということになります。

- 管理者が慣れている OS
  Linux は古いノート PC でも動きますので、最近では自宅でカーネルのハッキングをしたりして Linux に詳しくなっているという例が多くあります。そのため管理者のスキルが向上して管理コストが小さくなるという利点があります。つまりシステムに対して適正なコストで運用できるというという点が最も重要です。

- 多くのアーキテクチャで動く: Intel PC、SPARC、PowerPC など
  もちろん Intel PC が主なものですが、たとえば SPARC などでもライセンス切れなどの理由によりサポート patch が手に入らないような場合、OS を Linux にしてしまうということが考えられます。

- 大規模なマシンから小規模なものまで
  現在では aquarium や Cobalt Cube などのミニサーバ専用機も多く出ていますし、NEC や SGI などもサーバ PC で Linux をサポートしています。ホットスワップ可能なディスクなどの新しいハードウェアがサポートされている場合もあります。そのようなハードウェアを利用したい場合には Linux が良い選択肢になります。また古いノート PC でも Linux で比較的小規模な Web サーバを動かすには十分な場合があります。

## 3.2 Linux とディストリビューション

一言で Linux といいますが、実際は次のようなものです

- 狭義の定義： カーネルそのもの
- 広義の定義： ディストリビューション
  (カーネル+さまざまな基本コマンド+アプリケーション)

Linux のディストリビューションには、いろいろなものがありますので、何を選ぶべきかということが問題となります。まず一般的な見方として

は、パッケージが豊富であること、運用を含めた使い易さという点が重要です。つまり自分が慣れているということですが、運用コストの低減につながります。

サーバを構築する場合は、バグ、セキュリティホールに対するアップデートの速度が重要です。かつて Linux はセキュリティホールの荒探しの状態で、一時期の NT のような状況にありました。当時は日本国内のディストリビューションベンダの対応も鈍かったと思います。

また商用プログラムを使用する場合は、対応しているディストリビューションを選ぶことになります。たとえば Oracle には RedHat が、ApplixWare には TurboLinux が添付されています。しかし、サーバを構築するために Linux で使用されるアプリケーションはごく限定されています。あるとすれば、データベースぐらいですが、最近ではファイアウォールや Virus スキャナの製品も出ています。

重要なことは、自分が把握、理解している最小限の範囲のものを使うということです。このような観点から、ほとんどベースパッケージのみで構成されている Trinux のような最小限のディストリビューションを選ぶのも一つの方法です。最近絶賛されているような CorelLINUX のインストールした直後の状態は必ずしも良くありません。

## 3.3 インストール

(1) インストール時の指針

インストール時の指針としては、自分が把握している必要最小限のパッケージのみを選ぶことになりますが、パッケージは最新のものを使用して下さい。これは patch のコンフリクトを避けるためにも必要ですが、バグフィックスなどは最新のパッケージをベースに行われています。そのため、雑誌の付録の CD というのは良くありません。インストール時に最小インストールを選んだ方が、後から使用しないようにするよりも確実です。さらに、UNIX はプロセスの起動が比較的重く、SMP の効果はまだあまり出ていませんので、メモリやプロセステーブルの節約にもつながります。

(2) インストールの必要のないサービス

パッケージの中にはサーバとして必要のないサービスも多くありますので、以下にいくつか説明します。

- X Window: X そのものに問題がある場合も
  TCP/IP の port/6000 を使用すると侵入されるという問題、Xauth の認証や XFree86-3.3.2 での Xserver setuid のセキュリティホールなどがありました。

- **コンパイラ**
  **侵入の手段(踏み台にされる)となるツールがコンパイルこともありますので、なるべく他の同構成のマシンを別途用意し、そこでコンパイルしたバイナリをコピーするようにします。他のマシンを用意できない場合は、ソースからコンパイルをせずディストリビューションを信じてそのまま使うということも可能です。**

- **かな漢字変換**(canna、wnn **など**)
  **これは必要としません。一般にクライアント・サーバで使うものはセキュリティホールとなる可能性があります。かな漢字変換がどうしても必要な場合は、**SKK **をローカルで使用するか、**telnet **経由で** EUC **コードを送り込む、ということで十分です。**

  **注意すべきことは、日本で使うことを意識して作られたディストリビューションで、インストール時にかな漢字変換が最初に起動されるものがあるということです。多分もう修正されていると思いますが、**VINE **では** libcanna **とダイナミックリンクされており** canna **を削除すると** vi **すら動かなくなる、ということがありました。**

- **グラフィックライブラリ：**svgalib、fb、gtk、qt **など**
  **サーバではまず不要ですし、これらはハードウェアを直接アクセスするものが多いのでシステムのダウンにつながる可能性があります。**gtk **はメモリリークが多く、現状ではサーバにはまだ使わない方が良いと思います。**

- NFS、RPC、NIS
  **通常は不要のはずですので、**portmapper(/sbin/portmap)**を含めてサービスを停止します。**

- **その他のサービス**
  inetd **から起動するサービス**(finger, talkd **など**)**を確認して停止して下さい。また** inetd **経由でなく個別に起動されるものは** ps **コマンドで確認して下さい。**

**次に、インストールに関連した、いくつかの問題を紹介します。**

- **シェル問題**
  Linux **では** bash **が起動シェルになっている場合が多いのですが、ディストリビューションによっては** static link **になっていない場合があります。そのため、マシンのダウン時に修復用の** shell **が動かないというトラブルの元になることがあります。このような場合は、**sash (static A shell) **を使用して下さい。これは** static link **された最低限の機能を持つシェルで、例えば、**foreach **などはありませんが** /etc/init/* **にあるスクリプトぐらいは動作させることはできます。**Debian **では** bash **ではなく** sash **を使っています。**

- **ファイルシステムの問題**
  **現在のところ、**history **が記録できるファイルシステムは** Linux **には**

ありませんので、その対応はアプリケーション側で行うことになります。これについては現在の Linux の ext2 ファイルシステムと互換のある ext3 や afs などいろいろ開発されてはいますが、SGI が開発している XFS が kernel-2.4 位からサポートされるそうです。このため、現在のところは XFS 待ちといった状況です。

- NFS 問題
  これは kernel-2.2 の途中で nfsd の実装がユーザ空間のコードからカーネルコードへと変更されたために、ロック形式の違いなどで NFS が動作しないことがあった、という問題です。現在では knfsd (kernel nfsd) の登場で解決しつつあります。

- 普段からの情報収集
  普段からの情報収集が重要ですので、いくつかの URL を紹介します。Vine、Debian、Kondra の IRC で聞いてみるのもよいと思います。

| | |
|---|---|
| Vine Linux | *http://vine.flatout.org/errata.html* |
| Debian GNU/Linux | *http://www.debian.org/security/* |
| Kondara | *http://www.kondara.org/* |
| Plamo Linux | *http://www.linet.gr.jp/~kojima/Plamo/* |
| redhat | *http://www.redhat.com/corp/support/errata/* |
| Securityfocus | *http://www.securityfocus.org/* |

# 4 カーネル

Linux のカーネルは自分が利用するために(自分の責任で)自由に変更ができますので、これを生かすような運用を行います。

## 4.1 インストール時のカーネル

インストール時のカーネルは、インストールと、あらゆるサービスができることを主眼においた非常に generic なものです。40 種にものぼる SCSI ドライバや NFS や SMB、NCP など不必要なプロトコル、また CDROM や ISDN などまでがサポートされています。また、比較的古いバージョンのカーネルが使用されています。

たとえば kernel-2.2.13 でも問題が見つかっていますので、最新版を使用すべきですし、セキュリティやパフォーマンスでの問題もありますのでインストール時のカーネルはそのまま使わないようにします。あるいは、kernel-2.0 系で security fix patch を使うという方法もあります。

## 4.2 カーネルのリコンパイル

- 余計なドライバを削る： 多数の SCSI ドライバ、CDROM、NIC
  一部の NIC のドライバには Linux でサポートされているが、その NIC で動作しないプロトコル(multi cast、IPv6 など)があります。

- 必要のない機能を削る： a.out バイナリ、マウス、サウンド
  a.out バイナリは、現在主流の実行形式 elf 上で古い実行形式である a.out をサポートするための機能ですが、サーバには必要ありません。

- ネットワークプロトコル
  NFS、SMB、NCP、appletalk などを削り、CONFIG_FIREWALL などが必要なら加えます。CONFIG_FIREWALL は、IP change によるアクセス制限や IP accounting などの機能があるため非常に便利です。また syn-flood 攻撃の対策である SYN_COOKIES=y も必要と思われます。

## 4.3 カーネルのリソース制限

Linux ではカーネルをリコンパイルしなくても、/proc 以下にあるファイルのパラメータを変更することで、カーネルによるリソース制限を設定できます（たとえば、IPv6 のサポートなど）。ほとんどの場合は必要あ

りませんがチューニングを行う場合には非常に便利です。カーネルのドキュメントに説明されていますので必要な場合は参照して下さい。

```
/proc/sys/kernel/file-max
/proc/sys/kernel/inode-max
/proc/sys/vm/freepages
```

- フェイルセーフ patch： メモリの使用を厳しく制限する patch で、root の使った共有メモリをユーザが使用できないようにします。

## 4.4 TCP syn flood 攻撃への対応

TCP の hand shake を悪用した攻撃ですが、kernel-2.0.3x/2.2 系では既に対応されています。

カーネルコンパイル時に CONFIG_SYN_COOKIES=y として、/proc/sys/net/ipv4/tcp_syncookies を 1 にします。ほとんどのフリーUNIXでは、このあたりには同じコードが使用されています。

## 4.5 カーネル関連情報

カーネル関連情報については、次の Web ページが参考になります。

- KernelNotes.org — *http://www.kernelnotes.org/*
  一番良く見られています
- kernel Traffic — *http://kt.linuxcare.com/*
  サポート会社が運営する web ページ
- LinuxHardware.net — *http://www.linuxhardware.net/*
  大規模マシン向け
- Alan Cox's Diary — *http://www.linux.org.uk/diary/*
  カルト的、Alan Cox が今日どのあたりをハックしたかを見ることができます

最近は、*http://www.linux.or.jp*のトップページにもカーネル関連情報が掲載されていることがあります。

# 5 アクセスとサービスの限定

セキュリティ対策の基本はサービスとその対象を限定することです。

## 5.1 安全のためには?

- 利用できるサービスを制限する
  たとえば IMAP をサービスする必要がある場合は、SSL で使うといったようなことです。

- 安全な代替サービスを検討する
  このためには mailing-list で聞いてみることもできますが、コンサルティング対応のある SI ベンダを確保することが必要です。

- サービスを行うユーザ権限やファイルなどを制限する
  たとえば apache は nobody で動かす、一般ユーザは 1024 以下の port は使えないようにする、というような対応があります。

- サービス構成を理解した運用: 使用するディレクトリを知る
  重要なことは、たとえば sendmail が参照したり、使用したりしているファイルは何かという質問に答えられるか、ということです。

- 安全な実装の利用
  どんなソフトにもセキュリティホールは存在します。安全ということばには、「実際に攻撃が不可能だから安全」という場合や、「攻撃されたとしてもそれを防御できるから安全」という場合のように、二つの意味があります。

- サービスを絞る
  提供するサービスを少なくするために使わないものは削ります。しかし port 113 の ident などは、削ると相手のサーバがサービスを拒絶して、ftp や Web のサービスを受けられなくなる場合がありますので注意が必要です。これは適当なシステム名を返すだけで十分なので、そのような ident-client を入手して動かすだけでも回避できます。

## 5.2 login とパスワード

以下の事柄に注意する必要があります。

- login
  このサービスやシェルの提供については、特に考慮する必要があります。なぜなら、「ユーザ名」と「パスワード」のみという脆弱な最低

限の認証方法しかとっていませんし、リモートからほとんどの資源が利用できるということを意味しています。いったん login されてしまうと普通のユーザと一切区別が無いため、可能なことは何でもできるということで、「乗っ取り」や「他攻撃の踏み台」などにされるということも起こりかねません。

- パスワードファイル(/etc/passwd)
  パスワードファイルは、ユーザー情報が詰まっている基本のファイルですが、シャドウ化を行うとパスワードは/etc/shadow に保存されます。パスワードファイルの暗号化にはほとんどの場合 DES が使用されていますが、PAM(Password Authentication Module)を使用すると認証方式の変更が可能となり、より強力な MD5、SHA1 などを使用することができます。特に昔からのクラッキングツールには DES に特化したものが多いので、認証方式を変更するだけでも安全性は向上します。パスワードファイルはあらゆるプログラムが利用しますが、これはあらゆるユーザが読めるということと同じですのでシャドウ化は必須です。Debian ではシャドウパスワードが選択可能になっています。

- パスワードなどの盗聴
  さまざまなパケットの盗聴プログラムが存在しますが、クラッカがこれを基幹(バックボーン)にしかけると被害は甚大です。特に promisc mode で NIC が動作しているときは注意を要しますので、コマンド "ifconfig -a" により確認して下さい。もっとも侵入者が ifconfig コマンドを入れ換えるのは常套手段ですので、別途 lsop などのプログラムも利用すべきです。

  根本的な対策としては、スイッチやルータによりネットワークを分割し原理的に見えなくするという方法がありますが、ルータではネットワークの速度が遅くなりますので、現在は安価になったスイッチの方がベターです。また OTP (One Time Password)の利用や通信路の暗号化も有効な方法です。

- パスワード認証を必要とするサービス
  ユーザの特定や書き込みを行うサービスのほとんどは認証が必要になります。たとえば、telnet や ftp がそうです。Windows のクライアントにはメールの確認をするために 10 分に 1 回 POP や IMAP へのポーリングを行うものがあります。これだけの頻度でパスワードがネットワークに流れるとパスワードが盗聴される危険性が非常に高くなります。誰もが利用するサービスということもあり、POP や IMAP はかなり危険と言えます。

## 5.3 telnet、r 系サービス

リモート端末から login できるようにするサービスですが、login と同時にシェルを起動しますので、その後はすべてができるようになります。またユーザ名、パスワード、通信の内容などは一切暗号化されません。

.rhosts の設定でパスワード認証を行わないような設定での運用はより危険ですが、外部からのネットワークへのアクセスがない閉じた環境であれば問題はありません。

## 5.4 限定した権限および環境でのサービス提供

サービスの付随する危険性を軽減するために、そのサービスを特定のユーザ権限や環境に限定して提供することができます。ある種のサービスには限定したユーザアカウントで運用されるものがあり、/etc/inetd.conf あるいはそのサービスのプログラム内で指定されたりしています。たとえば、nobody アカウントは多くの WWW サーバや skk などで、サービスを提供するために使用されます。IMAP を mail ユーザの権限で動かすのも同様で、読み出しだけですむサービスなどはこのような形で運用することができます。

例： noboddy 権限で起動される skk サーバの inted.conf での設定
skkserv stream tcp nowait nobody /usr/sbin/tcpd /usr/sbin/skkd-cdb

サーバプログラムを動作させる場合に chroot()により、特定のディレクトリをルートディレクトリとして動作させることがあります。たとえば、/var/local_host1、/var/local_host2 などに chroot して仮想マシン的に使う場合などがあります。wu-ftpd など、サービス提供に危険性が伴うようなプログラムの場合、標準で利用するものもあります。この場合ライブラリ、その他の必要ファイルも忘れずに作成しておく必要があります。そうしないと、ftp で ls が動かないなどのトラブルの原因になります。

## 5.5 OTP と SSH

パスワードや暗号化には次のようなものがあります。

- OTP (One Time Password)
  パスワードの再利用攻撃を防ぐために平文パスワードのかわりに使う使い捨てパスワードです。サーバへの実装として OPIE などがあります。telnet および ftp の OTP 化したものでは/etc/opiekeys に MD5 化されたパスワードが保存されています。

- APOP
  OTP を使って認証するようにした POP(内容の保護は行わない)です。対応クライアントは増えてはきましたが、十分ではありません。例えば、Outlook-Express や Netscape/Messenger では APOP 対応していません。APOP の実装ではユーザ名と時間情報から OTP を生成していますのでかなり強力です。

- SSH(Secure Shell)
  通信路も含めて暗号化を行います。SSH の実装が良くできているのは local-secure を前提として鍵管理を簡素化しているという点で、

SSL などのように複雑な鍵管理の階層などは必要としません。インストールした直後の 1 回目の認証は弱いですので、安全な通信路を使う必要があります。SSH では IP アドレスは信用していません。例えば、最初の利用以後はホスト鍵に変化があると警告を行います。またユーザ認証手段に RSA や Diffie-Hellman の選択も可能になっています。SSH は F-Secure 社が商用ライセンスを持つものですので、ライセンス条項に注意する必要があります。

Laser5 のディストリビューションには SSH の商用ライセンスが添付されているようでが、この 10 月の OpenSSH (*http://www.openssh.com/*)の登場でこの点もクリアになりました。OpenBSD にはこのフリーな実装のものが同梱されています。

OTP と SSH の長所と短所は次のとおりです。

- OTP の長所
  あらゆる環境で利用可能です。OTP がわかっていれば OTP の計算を行うローカルな計算機を必要としません。OTP 対応のクライアントには、ftp では Windows の ffftp、Mac の fetch、telnet では TeraTerm などがあります。

- SSH の長所
  - 移行しやすい
  - r 系サービスのほとんどを置き換え可能
    たとえば rsh に相当するコマンドは ssh になっており、r 系のコマンドを置き換えるインストールも可能なので、ユーザが意識せずに利用可能です。
  - より強固な認証手段の選択が可能
    通信路も暗号化され、ホストとユーザの認証手段をそれぞれ別のもの(たとえば RSA と Diffie-Hellman など)にすることが可能
  - 他プロトコルをカプセル化(tunneling)可能
    例えば、APOP が使えない場合でも、POP を SSH でカプセル化して安全に使用することができます

- OTP と SSH の短所
  OTP は通信内容が保護されないという点をユーザにも意識してもらう必要があります。

  また SSH ではどうしても通信路暗号化のコストがかかります。最近の高速マシンではあまり問題とはなりませんが 100Mbps などの高速な LAN では、さすがに負担となります。たとえば、Pentium-II の CPU でも 100Mbps のネットワークでは ftp の速度が 1/3 程度になってしまいます。

# 6 セキュリティ強化ツールの活用

tcp-wrapper(libwrap)などのセキュリティ強化ツールを使用して、アクセス元、アクセスのアドレス、サービスの種類、利用者、時間帯などによるアクセス制御が行われます。たとえば、guest のログインは 9 時から 17 時まで、root は 24 時間可能などの制御可能です。Linux 独自(カーネル依存)のものですが、ipchains も利用可能です。

## 6.1 tcp-wrapper

tcp-wrapper は多くの場合 /usr/sbin/tcpd としてインストールされます。制御ファイルには/etc/hosts.allow と/etc/hosts.deny がありますが、tcp-wrapper-7.2 からは hosts.deny を使用しなくなり hosts.allow のみで制御されます。また、libwrap を用いるようなプログラム（configure 時の --with-libwrap オプションのあるもの）も同じファイルを使用しています。詳細については man 5 hosts_access を参照して下さい。

/etc/hosts.allow の設定例は次のようなものです。

- すべてのアクセス許可
  ALL: ALL: ALLOW
- 特定のネットワークからのアクセスのみすべて許可
  ALL: 10.0.0.0/255.0.0.0: ALLOW
  ALL: ALL: DENY
- サービス毎に制御
  imapd: 10.1.1.0/255.255.255.0: ALLOW
  in.ftpd: ALL: ALLOW
  ALL: ALL: DENY

## 6.2 パケットフィルタリングの利用

セキュリティ強化ツールとして以下のパケットフィルタリングが使用できます。

■ ルータの使用

ルータなどの他の資源を利用したセキュリティ対策も考慮すべきです。SOHO 用のダイヤルアップルータでもパケットフィルタリングは可能となっています。ルータはアクセス制御に有効ですが、その限界としてパケットレベルのフィルタリングにすぎないため、tcp-wrapper のようなユーザによる制御はできません。また、サービスに必要なもの以外は塞ぐべきです。通常外部からのアクセスを禁止すべきサービスには tftp、finger、portmapper、NBT、 snmp、exec、login、shell、X などがあります。

■ ipchains

これは Linux カーネルに依存した(Linux のみで利用可能な)ツールで、Cisco のルータとほぼ同じ形式のルールセットを使用できるフィルタリ

ングツールです。

- input、output、forward の評価をする
- デフォルトではルールなし(全パケットを通す)
- source、dist、protocol、interface ごとにふるまいを指定できる

これらは IP マスカレードで使われることが多いので、意識せず使用していることもありえます。

Linux をルータとして使用する場合、パケットフィルタリング機能を付加することができます。またノート PC ではネットワークデバイス ppp0 と eth0 との設定を変えて ppp では外部から内部へアクセスできないようにルールセットを設定すると安全になります。Linux カーネル内での処理を行う分ルータより遅くなりますが、 1.5Mbps ぐらいの低速リンクならば 486CPU でも十分実用になります。

- ipchains の適用範囲
  アクセス制御という点では tcp_wrapper のような細かい制御は不可能で、特定マシンの IMAP を制御するといった程度です。フィルタリング機能の高速性という点では、ルータ、ipchains、tcp-wrapper の順番で遅くなります。また NAPT(IP マスカレード)でも使用されています。応用例としては、NAPT+cipe と組み合わせて VPN を構成するというものがあります。

- ipchains の動作
  ipchains はカーネルで IP を再構成した後でその評価を行い、単純にヘッダを見て(必要ならば変更して)出力側の NIC に流します。たとえば NAPT の場合はプロトコル番号を変更します。tcpd のようにパケット内容についての評価はできませんので、必要ならば、アプリケーションレイヤの透過プログラム(たとえば squid のような)を別途用意する必要があります。

ipchains の応用例ですが、IP マスカレードと組み合わせて http プロトコルを強制的に特定の Squid/DeleGate を通るように設定するというものがあります(透過型プロキシ)。こうすると、ノート PC を持ち歩くような場合 proxy を再設定する必要がなくなります。

# 7 攻撃に対する防護策

運用を行う面でのいくつかの攻撃に対する防護策について説明します。

## 7.1 内部ユーザから防護: permission、resource、sudo、PAM

攻撃者はほとんどの場合、まず一般ユーザの権限を得てから root 権限を手に入れます。誰でも root になれるような設定は論外ですが、内部ユーザから防護することは攻撃者が root になる壁にもなります。root 権限を奪われないようにするには、まずファイルパーミッションを確認して下さい。

- world writable になっていないか?
  最近のプログラムでは、たとえば sendmail-8.8 以降のようにスプールが world-writable だと起動しない、などになっているものがあります。
- umask を厳しく設定する
- immutable ビットの利用： write されると syslog へ記録
  (Linux の ext2 ファイルシステムのみで利用可能)
- ファイルシステムのマウント： nodev, nosuid, noexec などのオプションを活用
  nodev：ファイルシステムをデバイスとしてのアクセスを禁止
  noexec： 実行パーミッションを無効にする
- プログラムの setuid： ほんとうに必要な setuid なのかを再考する
  たとえば/bin/mail は setgid mail されています。
- 資源占有を防ぐ： shell でのリソース制限: limit/ulimit
  サーバで使う場合は、ユーザを限定することをまず考えるべきで、普通はサーバには一般ユーザの login アカウントは作成しません。
- sudo
  これはコマンドを特定のユーザ権限、たとえば root で動作させるもので、su のコマンド実行限定版のようなものです。たとえば web 作成者に apache の restart をさせる場合など、運用上あるユーザに root 権限を与える必要が生じる場合があります。このような時に root 権限を与えて apachectl restart を直接無条件に使用させるのではなく、実行できるコマンドを限定するために shell script を提供して、その中で root 権限を必要とするコマンドは sudo 経由で起動します。su とは異なり sudo では、あらゆるコマンドが実行時刻およびユーザ名と共に記録されます。問題点としては、sudo から起動されたプログラムだけでなく、他のプログラムも起動できる可能性があるということに注意して下さい。

- PAM
  PAM(Password Authentication Module)は認証方法を交換可能にするモジュールです。パスワードの暗号化方式を DES 以外のたとえば

SHA1 にすることや、メモリ使用の制限などユーザリソースの細かな制限、さらに tcp-wrapper と同様なユーザの login 制御などが可能となります。

設定ファイルは、/etc/pam.d/以下に個別のコマンド名のファイルとして置かれています。

## 7.2 ネットワークからの防護

ネットワークからの攻撃の対象としては、カーネルに対するものとアプリケーション層に対するものの両方があります。サービスについては不要なものは起動しないということが重要です。

DoS 攻撃は、完全に防ぐ方法がありませんので対応が難しい問題ですが、まず第一に普段のリソース監視が重要となります。たとえば異常なほど大量な web サーバへのアクセスがあったとしても、普段の状況を把握していなければ、見掛け上は web のヒットなのか攻撃なのかは区別ができません。

## 7.3 物理的攻撃および電気的攻撃からの防護

- 物理的攻撃からの防護
  Linux だからといって特別なことは何もありません。運用次第ですが設置場所もあまり考慮されていないことが多いと思われます。

- 電気的攻撃からの防護
  電気的攻撃からの防護としてはサージと停電に対するものが必要です。通常 UPS を使用しますが、商用電源、イーサネット、電話線の 3 つのラインに対するトラブルから防護できるようなものが必要です。Linux で使える UPS にもいろいろあり、最大手ベンダの APC は apcupsd などをフリーで提供しており廉価版(2 万円)のものも発売しています。ただこの廉価版の UPS では shutdown まではできません。その他三菱からも UPS が発売されており、FREQUPS に附属したプログラムが Linux から利用可能です。

## 7.4 破壊にいたらない場合の対策

破壊にいたらない場合というのは、ファイルやシステム領域が破壊されてブートできなくなるというような直接的な被害ではなく、UCE メールの踏台にされて他のサイトから苦情が来たり、DoS 攻撃により本来のサービスの提供に支障をきたしたりするというような場合です。UCE メールの踏台などになって加担すると、インターネット社会から締め出されてメールが送れなくなる恐れがあります。

Linux は Mail 管理が甘いというのが一般の認識であり、特に日本国内の

サーバがデフォルトのスキャニング対象になっているツールもありますので十分注意する必要があります。

DoS 攻撃のためのポートスキャンは日常的ですので、注意が必要です。DoS 攻撃によりサーバが十分なサービスがを提供できなくなるのも問題ですが、ネットワーク自身がトラフィックに埋まってしまうと、すべてのマシンのネットワーク上での活動が困難になってしまいます。そのため、これもセキュリティ上の問題と同じように扱う必要があります。

## 7.5 問題が発生した場合?

問題が起きてしまった場合には、できることは対策窓口を作ることぐらいです。いわゆる「お客様相談センター」的なものを設置して姿勢を示すことが必要です。

RFC2142 にはこのような場合に必要なサービスが記述されており、たとえば abuse、postmaster、 hostmaster などの e-mail エントリは最低限作成しておくべきです。その他、JPCERT/CC や契約 ISP などへ報告することも必要です。

# 8 システムの運用と管理

サービスを安定して稼働させ続けるためには、以下のことが必要になります。

## 8.1 日常の管理項目

- 全体のリソース: ディスク、メモリ、プロセス
  ほとんどの問題は可動部分に関連して発生しますので、まずディスクについて使用状況などを確認しておくことが重要です。
- ユーザ管理: ユーザの更新(削除)、パスワード
- データ管理: バックアップ
  データの保全のためにはバックアップ以外には方法はありませんので、バックアップを取ることが重要です。
- その他: マシンの物理的な環境(空調など)

## 8.2 全体のリソースの監視

まず、自分で提供しているサービスを使ってみることが重要です。自分で使用していないと、そのサービスを提供していることを意外と忘れがちになります。そのため、セキュリティ patch の適用を忘れたりする場合もあります。また、さまざまな不具合や動作上のボトルネックなど、

使ってみて初めて気がつくこともあります。全体のリソースの監視にはまず、ps、top、df などを実行してみて下さい。稼働しているプログラムから管理者宛てに送られるメールには最低限の情報が含まれていますので、最低でもこれぐらいはチェックしておく必要があります。

また「mon」という、以下のようなことをまとめて監視できるツールもあります。(*http://ftp.kernel.org/software/mon/* )

- 一般的なネットワークサービスをほぼ網羅： DNS、www、ping、ntp など
- ホスト情報： ディスクスペースなど
- SNMP と連携： uptime、プロセスなど snmpd がサポートしている情報が得られる
- Web ブラウザからのコントロールも可能

## 8.3 log によるシステムの監視と記録

ホストおよびネットワークの通常の状態を知ることが重要です。何か起きた場合にそれが果して攻撃なのかどうかを判断することができます。またサーバはサービスを行うことが第一の目的なので、サービスに対するリソース不足を理由にして設備要求を行うための材料にもなります。

- ホストの監視： syslog
  ホストの監視としては、syslog が重要です。 (/etc/syslog.conf を確認し下さい)

  留意点として、syslog は udp を使用していますので、しばしば記録すべき情報を取りこぼすことがあります。また syslog 自身が攻撃される可能性があり、/var/log/syslog ファイルがあふれたりすることがあります。可能であれば外部からのアクセスに対しては、syslog のポートをルータで塞ぐなどして下さい。パフォーマンスが必要な場合は、cyclog、Syslog-ng などの同等のツールが使用可能です。

- アプリケーション独自の log
  サーバ用のアプリケーションにはそのツール独自の log を出力するものが多く、これらはそのサービスの特徴が出ており利用価値も高いものですので慣れるだけの価値があります。web や ftp も独自の log を出力しますが、これらには専用の log 処理ツールが多数あります。また、iplogger というツールがありますが、これはアプリケーションというよりは、TCP、UDP、ICMP などの接続そのものの監視を行うものです。負荷が高いのが難点ですが、他のコネクションとの関係がわかるので非常に便利です。

- 異常の発見
  異常の発見はまず、普段と違う挙動を発見することから始まります。たとえば、同じ行為が異常に繰り返して行われる場合には、総あたり的な telnet コネクションのリトライなどによるものがあります。またサービス外の port へのアクセスによりポートスキャンが発見さ

れます。クラッカの典型的な手法を知っておくのは重要なことですが、ニュースグループ news:alt.2600 などによく流れている URL をチェックするといろいろな情報が流れているのがわかります。

- log の監視を楽にするツール
  log の監視という作業は管理者にとって非常に負担となるため、監視作業を継続するためにも log の監視を楽にすることが必要です。次のようなツールがあります。
  - syslog-summary： 同じメッセージがあればカウントして表示します。(繰り返し攻撃を受けている場合の検出が容易)
  - Xlogmaster　　http://www.gnu.org/software/xlogmaster/
    GNU tool、GTK+ が必要なのでサーバでは動かさずに別途用意したマシンで使用します。/proc や syslog のリアルタイムモニタを行うツールで、赤色文字やブリンクによる表示、ビープ音や pager への通知などが可能です。
  - swatch：パターンを検出したらリアルタイムで通知するツールです。メール、ビープ音、端末出力などなどで通知できます。

- 特定アプリケーション用 log 解析ツール
  専用ツールですので良くできているものがたくさんあります。
  例：
  - MTA
    sendmail 用　　syslog_stats.pl
    postfix 用　　pflogsumm.pl
    qmail 用　　qmailanalog
  - www 用
    analog
    weblaizer
  - ftpd 用
    xferstats

- ファイルの改竄、変更監視
  クラッカは都合のよいようにファイルを変更することが多いので、そのような場合に対応するツールです。

  tripwire は、有名なツールで、変更頻度の少ないファイルに対してファイルのハッシュ値 DB を作成し、チェックして変更を検知します。現在では、商用になっていますのでサポートを受けることもできます。

- ネットワークの監視
  promisc モード: ネットワークに流れるすべてのパケットを NIC から取り込むモードで、見てはいけないものも含めて何でも見ることができてしまいます。サーバマシンと独立に promisc モードで監視するマシンを設置することで、サーバへ負荷をかけずに運用可能となります。簡単なパケットモニタとしては iptraf や特に有名な tcpdump があり、その他 libpcap を使用して SFC の学生が作成しているものもあります。

NFR(Network Fright Recorder): これは商用の侵入者検出プログラムで非商用利用はフリー(ソースも提供)ですが、製品としての価格は高価であったと思います。収集したパケットを解析し、その結果はJAVA 対応ブラウザでの参照が可能です。

## 8.4 バックアップ

バックアップに要求されることとしては、ある時点に戻せるということが重要です。メディアには上書きできないものの方が良く、ディスク上の ISO イメージを直接見ることのできるツールもありますので安価になった CD-R が今ならば最適かもしれません。HDD をバックアップメディアとして使用することもあると思います。その場合バックアップ後もホットの状態でスタンバイさせておくという運用方法もありますが、停電などによるディスクトラブルが考えられますのでバックアップ処理後は外して安全なところへ移動させておくようにすべきです。

バックアップの対象

- インストール直後およびサービス直前のホスト全体
- 定期バックアップ： /etc、/home
- 不定期バックアップ： ホスト全体

# 9 まとめ

- **安全はあたりまえのことの積み重ねから**

  **こまめな情報収集や自分でサービスを使用することが重要です。Linux で標準的に使用できるツールは、F-Secure 社の Virus スキャナや透過型のコマンドスキャナなどの商用プロダクトを導入した場合でも重要です。しかし複雑なシステムを導入して危険を覆うより、危険の除去を考える必要があります。Linux はソースも入手できるので、コストに見合った投資のできる(コストと効果のトレードオフ点を選択して運用できる)OS であるということができます。ミスの減少を計るために log の監視などは自動化し、人はコンサルティングなどの人間しかできないところに注力すべきです。**

- **最後は人間**

  **最後は人間が問題となります。いろいろな環境を作ってとにかく慣れることで、Vine、Debian、Slackware など他のディストリビューションも使ってみてその違いを楽しんでみるぐらいのことは必要です。そのことにより各ディストリビューション依存部分や、今後すたれていく部分を見極めることができるようになります。**

- **攻撃してくるのは人間**

  **オープンソースの思想とは反しますが、自システムの状況や弱点をインターネット上で不用意に公言するのはやはり避けるべきです。**