Linux サーバ構築とセキュリティ

久保 元治 (Linux Business Initiative 代表、(株)サードウェア代表取締役)

1998 年 12 月 17 日

InternetWeek 98　国立京都国際会館

(社)日本ネットワークインフォメーションセンター編

目的

Linux サーバの構築について説明し、セキュリティ対策についての「身の丈」という提案をします。また、アクセス制御やシステム運用管理、実際のセキュリティ対策に必要な情報収集についても紹介します。最後にサポートなどのビジネスの現状についても簡単に触れます。

目次

1. はじめに
2. サーバ構築の概要
3. 「身の丈」セキュリティ対策
4. アクセス制御
5. システム運用管理
6. 情報収集の方法
7. 最近のセキュリティ動向から
8. 最近のビジネス状況
9. Q&A

1. はじめに

最近、Linux(リナックス、リニュックス、ライナックスいろいろな呼び方があります)が新聞などで取り上げられる機会が非常に増えてきております。日経の各紙、日経 BP の雑誌、当然ながら Software Design など UNIX 系の雑誌は昔からですけれども、最近ビジネス系の雑誌や新聞でもよく出ています。そこでかぶせられている形容詞、表現というものをいくつか取り出してみました。

まず「フリーPC UNIX である Linux は」という表現があります。「無料 OS」という表現ですが、これは姿を消してほしいと思っており、実際姿を消しつつあります。フリーを無料と訳しているわけです。無料ということが本質ではないので、これはうそというか、明らかに間違いだと思います。

「オープンソース」これはフリーという言葉を一種置きかえたわけですがその特徴はソースコードが公開されていて、だれでも自由に入手して改変できる権利というところを表している言葉として、私もこれを好んで使っています。

そして、10 月の頭に使われた表現として、「Linux のようなオープンソースのものは」―「OS は」というよりも「オープンソースのものは」ということですけれども、「公共財である」という見方があります。その上でいろいろなアプリケーションを動かしていく共通の基盤、インフラとして、そういう意味で OS など基本的なユーティリティ機能、こういうものは公共財であろうという見方があります。

そして、最近言われ出して、とうとう出たかと思ったのが、「次世代 OS」という表現です。「次世代 OS」ということで、そこまで新聞社さんが認知してくれたというか、勝手に認知したというか、そういう状況になっております。

私のタイトルに使われている、Linux Business Initiative について簡単に説明しておきます。今年２月に Linux をサポート、普及させていこうという会社が集まりまして、緩やかな共同の活動をやっていきましょうということで作った組織です。Linux の普及を促進するための情報提供や共同マーケティングなどを現時点での活動目標にしておりますが、私を含めて各社相当忙しいらしく、例えば web の情報更新なども実際にままならないという状況です。また、緩やかな集まりということで、現時点では例えば規則や独自の会計を持つなどには至っておりません。現在の会員数としては約 30 社前後だろうと考えております。

2. サーバ構築の概要

Linux のインストール自体は非常に簡単になってきました。ここではセキュリティ対策を意識したサーバ構築のポイントを紹介します。

「Linux の種類」について

・「Linux」とは

カーネルの名称(本来の意味)

カーネル、周辺コマンド、アプリケーションなどを統合した

「ディストリビューション」(広義)

代表的なディストリビューション

Red Hat、TurboLinux、Slackware、Debian、Caldera など

Linux については、ご存じのように Slackware とか Red Hat、Debian、その他いろいろなパッケージまたはディストリビューションというものがあります。ここしばらくは、それぞれ特徴のある新しいものが出てくるという発散のような状況が続くかと思います。この意味では一種の混乱といった時期にいったん入るのではないかと思っております。

Red Hat を取り上げる理由

・運用管理を重視したパッケージング

・バグ対応などのアップデートが迅速

・商用ベースのサポートも利用可能

・(私自身の)経験が最も豊富

・以下、おもに RedHat 5.2 ベースで説明

ここで取り扱いますのは、そして私どもが実際にビジネス上で使っておりますのは専ら「Red Hat」になっております。インストール周りについては、各パッケージの差異もあるのですが、実際に使う場合には、大体共通性が高くなっています。なぜ Red Hat かという理由ですが、これは他のパッケージがだめだというわけではありませんが、私どもが実際に約２年使ってきました経験から説明しておきます。

まず運用管理重視というパッケージングになっています。例えば、システムの設定ファイルは/etc の下に基本的に全部集められており、設定内容のバックアップする場合はそこを取っておけばいいという構造になっています。これは Slackware などでは、デフォルトでは、/usr/local/etc などに分散しています。ただし、これは Red Hat が供給している RPM という形式のファイルでインストールした場合ということで、ソースから展開した場合は当然、開発者のデフォルトの設定になります。

それから２番目に、これがある意味で一番楽できるところですが、バグ対応やセキュリティホールなどに対してのアップデートが非常に迅速であるという点があげられます。いくつかのメーリングリストなどで、セキュリティホールなどの議論が行われ、そのパッチなどがポスト、テストされて OK になると、翌日、２、３日後には Red Hat のアップデート

のディレクトリに RPM 形式でアップデート版が公開されます。ですから Red Hat ベースでは、そういう動向をつかんでいれば、出たらとりあえずその RPM で入れかえて、経過を観察して OK ならば、その件に関してのアップデートという対策は終わりというアプローチが可能です。
それから３番目は、商用ベースのサポートも利用可能という点です。現時点ではまだまだこれからだとは思いますが、基本的にまず米国では Red Hat 社自身がいくつかのサポートの有料区分も設けたものをやっていますし、その他の会社でもやっています。日本でも、多分来年２月、３月ぐらいをめどにしたいと考えておりますけれども、私の方でも始めていこうと思っております。
それから私自身、これが一番経験が豊富であるということです。ただし私自身はこれまで専ら Red Hat 4.2 でやっておりますので、5.2 は今日が初挑戦ということになります。

インストール

- ほとんどの PC (AT 互換機)にインストール可能
- バージョンごとにインストールがより容易に
  (Red Hat 5.2 ではハードウェアを自動検出)
- 難しいのはパッケージ選択
- Server インストールは選ばない方がよい(Custom を選ぼう)

現状で大半の PC でインストールして動くといえます。X などになると少し苦労することがあるかもわかりませんが、私の提案はサーバに関しては X は入れないということですので、大半のマシンで大丈夫ということになります。
Red Hat に即して申し上げますが、まずハードウェアの自動検出が非常に進みました。例えば、SCSI カードなどのアダプタカードは自動検出されます。X の場合でもビデオカードなどは自動検出されて、多分パラメータで入れるのはモニタの水平・垂直の周波数などのスペック部分だけです。マウスも自動検出ですので、少なくとも一般的なハードのチップの場合は、大半のいわゆる変動要素は自動検出されると思います。

サーバ用のパッケージ選択

- 鉄則: 必要最小限かつ最新のパッケージをインストールする
  最新のディストリビューションを使う
  書籍添付の古い CD-ROM は論外
  FTP の updates ディレクトリで最新パッケージにアップデートするのが望ましい
- よくわからないものは、とりあえずインストールしない

しかしパッケージ選択というのは、現実に非常に難しくなっています。これは 4.2 のときからも感じていましたが、Red Hat が設定しているカテゴリー分けは必ずしもよくないようです。5.2(5.1 でもそうかもしれません)では Server インストールというものが Work

Station、Server、Custom という３つのカテゴリーで最初の方に出てきます。Server インストールをやってみましたらハードディスクを全部消されてパーティションも全部作り変えられてしまいました。これは、何十というサーバをインストールするための、いわゆる NT 対抗のためのわかりやすさを強調したパッケージングだと思うのですが、このような構造になっています。ですので Server インストールを選んではいけないと考えます。

サーバ用のパッケージ選択ガイドライン

・最小インストールし、rpm -i で追加インストールするのも有力
・X ウィンドウは不要
　　コマンドベースで十分管理できる
　　システムリソースが少なくてすむ
　　X 自体がセキュリティホールになることも
・コンパイラ類も極力インストールしない

ではガイドライン的にどうするかというと、特にインターネットに露出させることを考えると、外部からの不正アクセスが必ず来ます。そういう意味から、できるだけ穴になるような機能を最初から持たせないということになります。ですから必要最小限かつ最新のパッケージのインストールが鉄則になります。
最新のという方も当然、アップデートの問題があります。まず雑誌とか書籍についてくる半年前、１年前のパッケージが手元にあるからといっても使ってはいけないということです。最新のものを使った上で、FTP サイト(Red Hat やミラーサイト)の updates ディレクトリからの最新のもので上書きということまですべきだと思います。
また、よくわからないものはとりあえずインストールせず、後でどうしても必要だというのがわかった時点で入れるというのも重要です。パッケージ選択ではカテゴリーごとに選ぶのではなく、個別にパッケージ名(RPM のファイル名)を選択するのが本来は一番よいのでないかと考えています。
現実的には、まず最小インストールを行うのがよいと思います。最小インストールというのは、パッケージ選択を一切せずにインストールしてしまうということで、システムブートなどに必要な最小限のものだけがインストールされます。これでも余分なものが入ったりしますが、後で１個ずつ必要なパッケージを追加する、例えばメールサーバならば sendmail、というようなアプローチがあります。
X ウィンドウは、リソースも食いますし不要だろうと考えています。ユーザ管理などの X ベースの便利なユーティリティがあるのも事実ですが入れるべきではないと思います。
それから、実際に外に露出するサーバならばコンパイラ類もインストールしません。コンパイラなどが入っていると、当然何かがあったときに万一悪用されるということがあります。どうしても自分自身でコンパイルが必要な場合は、別のマシンでコンパイルして持ち

込むというようにすべきです。

インターネットに接続する前に

・セキュリティポリシーを立てよう

・不要なサーバはホールになりかねない

ps ax **で起動されているデーモンを確認**

・最小限のアクセス制御

/etc/inetd.conf **の設定**

/etc/hosts.allow **の設定**

・パスワードのシャドウ化

4.2 **では** pwconv5**、**5.x **では** pwconv **コマンド**

最近いろいろなところからアタックが来まして、そのアタックが来た元を調べてみたら、telnet **が開いていて** Red Hat **などのプロンプトが出てくるというのがよくあります。つまり、大半のところではインストールが終わったら息切れして、「さあ、できた。つないでしまおう」ということではないかと推察しています。**

インターネットに接続する前には、何をしたいのか、何を許すのか、何を禁止するのかなど、最低そういう側面だけでもポリシーをきちんと立てる必要があります。

極めて現実的にいきますと、不要なサーバの機能はアクセスされてねらわれるということになりますので、ps **コマンドなどで起動されているデーモンを確認して、分からないものがあったらオンラインマニュアルで調べて、自分が理解していないデーモンが動いているということは避けましょう。また、つなぐ前に最小限のアクセス制御として、**TCP wrapper**、パスワードのシャドウ化など、せめてこれだけはやってほしいなと考えております。**

3. 「身の丈」セキュリティ対策

コストや人手に制約がある場合でも、セキュリティ対策は欠かせません。ここでは現実的なセキュリティ対策について説明します。

「身の丈」とは

- セキュリティ対策は、本来手間がかかり、専門的知識が必要
- 大規模サイトだったら、可能だし必要
- 小規模サイトだったら….
  - かけられるコストと手間は限られている
    - 利用目的を絞り、アタックされにくくする
    - そのための重要ポイントは何か
    - 継続的な対応のための有用な情報源

セキュリティ対策というのは、本来そんなに複雑怪奇で難しいというわけではないのですが、ある程度の手間と専門的な知識がやはり必要になります。大規模なサイトでしたら、人手や予算を確保できますが、小規模サイトではそのコストや人材、手間などはどうしても限られてしまいます。しかし、インターネットにつないでメールや WWW ブラウジング、また WWW サーバ公開などのメリットを享受したいわけです。そうすると、そこに割り切りが必要になるだろうというのが、この「身の丈」という考え方です。
まず利用目的をしっかりと絞り込み、そのために何をすればいいのかはっきりさせることです。もうひとつは、継続的な運用が不可欠になりますので、どこを見ればよいかという情報源をと確保しておくということになると思います。

セキュリティ対策の必要性

- UNIX 系はリモートからシェルを使える
  - 自滅ではすまない
- 不正侵入されたら….
  - システム資源の悪用
  - 業務データ、プライバシーの流出
  - 「踏み台」として使われる
    - ・他サイトをアタックするための出先基地
    - ・社会的信用の喪失につながる

セキュリティ対策に関して、Linux も UNIX 系ですので UNIX 系ということで考えると、まず万一侵入された場合、シェルが使えるということになります。これは自滅することもあれば、自滅では済まずに他のサイトに被害を与える。それが現在、世界中毎日のように来ているポートスキャンなどとなって現れるわけです。

引きかえに、NT というのは今のところはどちらかというと自滅型です。サービス妨害やサービス不能など、落とされたり、それから OS 自体がハングアップするようなコードを実行されたりという、どちらかというと自滅型主体ですので、他のサイトに被害を及ぼすことを最小限にしたいというならばNTで組むというのも一つの選択肢なのかもしれません。
それから WWW サーバですが、CGI などが要らない、いわゆる静的なページだけでいいというときには、Mac がよいのではないかという説も最近非常によく聞きます。
それでは、不正侵入された場合に何が起こるかといいますと、システム資源の悪用、業務データ、プライバシーの絡むデータなどの流出、それから現在世界的にはやっていますポートスキャンなどの「踏み台」として使われる、ということがあげられます。
踏み台として使われることの大きな問題点は、自分のサイトにとってもそうですが、他のサイトに迷惑をかけるということです。「どこから来たんだ」「何とか co.jp」ということで、やはり社会的信用を重視する必要があるのではないかと思います。

まさか私のマシンは…

・これは通用しない!
　自動化したアタックツールにより全部のインターネットサーバが標的になっている
　インターネットに接続したら、たった数日後に不正アクセスが来ることも

つい２、３日前も CERT からの Advisory が出まして、そこでもやはり自動化ツールを使ったアタックが指摘されていたと思います。
まずインターネットにつないだ時点で、トップレベルのドメイン、JPNIC とか NIC とか、そういうところに DNS 登録をします。そうすると、私のところでも経験がありますけれども、つないだら２、３日後にもう自動化されたアタックツールで、ポートスキャンその他のアタックが来ます。これは特に JP ドメインをサンプルとして引用したマニュアルがついているような自動化したアタックツールがあります。
これは「来ないだろう」「ひょっとしたら運が悪かったら来るか」ではなくて、確実に来るという前提で考えなければいけません。

不正侵入までのステップ

・利用可能な「入り口」のスキャン
　スキャンを自動化するツールが出回っている
　入り口が見つからなかったら別のサイトに回る
・とりあえず侵入
　パスワード推測、セキュリティホールなどを利用
・「バックドア」の設置
　次回以降の侵入を容易にするため
　プログラムの置き換え、設定の変更など

・侵入を検出しにくくする
　侵入者を見せないプログラムに置き換え
　ログの改竄など

不正侵入までのステップとして最近は、まず利用可能な「入り口」であるポートが開いているかどうかスキャンするということから入るようです。
そのスキャンを自動化するツールが出回っており、具体的には telnet や pop などのいわゆる well-known のポートが開いているかどうかを調べます。そして、もし何かできそうだということになると、とりあえず侵入を試みるということが次に続くようです。
そこで使われるのが、例えばパスワード推測で正規に入り込むためにパスワードファイルを盗もうとか、それから最近では、DNS、bind のバージョンを調べて、bind のセキュリティホールを使って入り込もうとか、そういうことをしていきます。
では侵入に成功したとしますと、当然ながら次回以降の侵入を容易にするために「バックドア」と呼ばれるプログラムを仕掛けたりします。また入ったことがすぐに検出されたら対策をされてしまうので、侵入したこと自体をわかりにくくするために侵入して動かしているプログラムなどを見せないように、プログラムをすりかえるということもやります。例えば ps コマンド自体をすりかえて、アタッカーが動かすプログラムが ps や ls で表示されないようにするなどそういうことも含めたプログラムの置きかえを行います。またログを消してしまう、改竄してしまう、ということを行ったりします。
その後で自分がやりたいことを正々堂々とやるというようになるわけです。

セキュリティポリシー
このような不正侵入に対して何を考えるべきかといいますと、まずセキュリティポリシーです。サーバをどのように運用していくのかという方針をしっかり立てるべきだろうと思います。
サーバ設置の目的は?
・できるだけ明確に絞り込む
・危険なサービスは実施しないか、代替手段を使う
　　telnet、FTP などは極力使わない
・当面利用しないものは動かさない
　　POP のみなら IMAP はインストールしない
・利便性とセキュリティはトレードオフ
目的は、できるだけ明確に絞り込む必要があります。例えばメールサーバとしてメールのやり取りをやりたいのか、FTP サーバは必要かなどです。
FTP サーバは運用管理の点でできれば避けたいものですが、WWW サーバの HTTP でちょっとしたファイルならばダウンロードさせることも可能ですので、そちらに一本化するこ

ともできます。具体的にできるだけ明確に、必要なものや目的を絞り込むべきだと思います。その中で特に危険な、クラッカーにとって利用価値が高い、telnet や FTP などは、目的を絞る段階で極力使わずに済むように考えるべきと思います。
また、いずれ使うかもしれないが当面は要らないものは動かさない、または最初からインストールしない、ということです。Windows クライアントなどからメールを取るために POP や IMAP を使いますが、例えば、今のところクライアントは全部 POP ならば IMAP は入れないようにします。
ユーザがたくさんいると、いろいろな要求が出てきます。例えば「自分のプロバイダ経由で会社のマシンに telnet で入りたいんだけど」などです。このような利便性、サービスの使いやすさとセキュリティはトレードオフですので、後はどれだけの手間と費用をかけられるのかという点でバランスをとっていくしかないかと思います。

何を保護するのか? その理由は?
・データ: インターネットサーバに内部データは置かない
・社会的信用: 踏み台、メールの不正中継対策がとくに重要
・ハード資源やソフトウェア自体: 再インストールは手間と時間のムダ
セキュリティポリシーとしてもう一つ考えなければいけないのは、何を保護したいのか、その理由はということです。すべてのものを保護したい、という考えもありますが、やはりその対策を考えなければいけません。
例えばデータに関しては、インターネットに接続して露出している、また非武装ゾーンに置いているサーバ上には、内部データは置かない方がよいでしょう。WWW と連携するような場合でも、最小限のデータだけ公開している WWW サーバ上に持っていく、コピーの方法自体も社内のデータベースマシンと TCP/IP Socket などで通信をしてリアルタイムにアクセスするというようにして、サーバ上に内部データを置くことは極力避けます。
また、社会的信用、体面ということを非常に気にしなければいけないサイトも当然あります。そういう場合は踏み台に使われるようなことは極力避ける必要があります。それから、最近無視できいのがメールの不正中継です。
また、ハード資源やソフトウェア自体というのもあげておきましたが、やられたら再インストールなどにいくら短くても半日、１日とか食われます。データが持ち出されたということになると、その経路やインパクトを調査をしたりなど、もっと手間や時間のかかることになります。

どういう手段で保護するか?
・外部からのログインは不可欠?
・パスワード管理は?
・WWW サーバで CGI は必要?

・日常の監視内容と方法は?

スキル、かけられる時間によって、保護方法と監視方法を決める

どのようにして保護するかということに関して、もう少し具体的にいきますと、まず、外部からのログインは不可欠か、ということが問題になります。これが多分インターネットにつなぐ大前提になると思います。通常は大半のサイトでは、一般ユーザに対して特にインターネット経由で通常の telnet でのアクセスを認める必然性はないと思います。別の方法、例えば電話回線でのリモートアクセス(こちらも危ない面はありますが telnet よりは安全)や暗号化した SSH などを使う、などという対策がとれるはずです。ですから、裸の telnet のポートをインターネットに向けて露出する、開けるということは、多分私のイメージではほぼ 90%以上のサイトで実は不要ではないかと、真っ先に封じましょうということがいえると思います。

それから、パスワード管理です。telnet でも POP でも他の何でもそうですが、パスワード管理は極めて重要です。これをどうするのかという問題があります。

また WWW サーバで CGI を実際に動かす必要があるかということも問題です。今日の流れでは大半のところで必要になってくると思いますが、CGI はセキュリティ的に非常に怖い要素をいろいろ持っていますので、必要でしたら必要なりにかなり突っ込んだ検討が必要です。

それから、日常の監視方法・内容についても考えておく必要があると思います。やはりこれは運用する側のスキルや、現実的に運用監視に割ける時間によっても、現実的な割り切り、対応が必要になると思います。

最新号か前の号の「日経コミュニケーション」か何かに、このあたりの調査レポートが出ていましたが、日常のログ監視をしっかりやっているところほど不正アクセスを受けているという結果が出ています。これはその中でも考察されていますが、要するにログを見ていないところは不正アクセスすら気がついていないということの裏返しだろうということです。ログを監視することにより、初めて不正アクセスなどの兆候がつかまえられ、最悪の場合でも早く検出できる、ということで、監視方法はやはり考えておく必要があります。

問題が生じたらどう対応するか?

・緊急時にパニックにならないために

・現象の把握とネットワークからの遮断

・原因の追求

復旧方法

・連絡先、アドバイザーを明らかにしておく

それから、問題が生じたらどのように対応するのかということもポリシーとしてあらかじめ考えておくべきだろうと思います。これをやっておきませんと緊急時にパニックに陥ります。やられたということでパニックになってしまい、よく考えずに再インストールして

しまうということになると、再発対策を講じることができなくなります。
多分一般論としては、ネットワークから即時いったん遮断する、ケーブルを外すということが最初の対応になるとは思いますが、相手が何をしているか観察する必要がある場合もあります。特に入り込まれて何かやられている最中でしたら、何をやっているのかを観察するというフェーズをとってから切るなどということが必要な場合もあります。
それでまず現象を把握します。そして原因の追及と被害の程度を調べるなどということを行い、最終的に復旧の方向に向けて動くということになると思います。こういう緊急時の対応も考えておくべきだと思います。
緊急時の連絡先とかアドバイザ、外部コンサルタントを明らかにしておくことも重要だろうと思います。JPCERT などにも連絡すれば、割と一般的ですけれども、役立つ情報を得られたりもします。

オープンソースの対策

・Linux カーネル自体のファイアウォール機能
・有用なアクセス制御ツール
　　tcp wrapper
・不正アクセス監視ツール
　　Tripwire、swatch など
・こまめな情報収集とアップデート

オープンソース、いわゆる Linux のようなソフトを使った場合でも、こういうことがここまでできるということで項目をあげておきます。
まずカーネル自体がファイアウォール機能を持っております。パケットフィルタリング、それから有名な IP マスカレード、このあたりが絡んできているわけですが、ファイアウォール機能が使えます。
それから、Linux の周りのネットワークのツールして TCP wrapper というアクセス制御のツールがあります。
不正アクセス自体を監視するツールとしては最近、リアルタイムでパケット自体の中身までチェックするというような製品が出始めており、オープンソースでのインプリメントもプロジェクトがありますが、そこまでいかなくても不正アクセスを監視、検出するツールとして、Tripwire、swatch、(私はスウォッチと発音しています)というものがあります。
また、オープンソースならではということでは、ソースを公開しているので情報収集をきんとやっていけば、ディスカッションの最中から情報が収集できるという点があげられます。すべての方がそこまでやっているわけにはいかないということで、それをまとめたサービスもあるわけですし、こまめな情報収集とアップデートが可能であると思います。

「身の丈」対策のポイント

・不正アクセス自体は不可避と覚悟する
・目的を絞り込んでそれだけを公開する
・一般に危険なサービスは極力禁止する
・情報収集とこまめなアップデート

この項のまとめとしましては、まず不正アクセス自体は不可避というより必ず来るということです。被害を受けないようにするためには、利用目的を絞り込む、ある意味で非常に消極的なイメージともいえますが、やはり自分の身を守るために目的を絞り込んでそれだけを利用可能にする、公開するということです。一般的に危険だといわれる telnet や FTP は、可能であれば、というよりも原則禁止の方向でまず考えるべきだと思います。それから、情報収集とこまめなアップデートの体制。これはパニックの際の対策とも通じますが、これらを確立しておくということがポイントだと思います。

4. アクセス制御

セキュリティ対策の最初のポイントはアクセス制御です。ここでは主要なサービスごとのアクセス制御について説明します。

ログイン

・ログイン、シェルの提供は両刃の剣
**　リモートからシステム資源の利用や管理が可能**
**　不正アクセスでシステムの「乗っ取り」、「踏み台」化も可能**
・ユーザ名とパスワードでログイン可能に
**　推測しやすいパスワードはきわめて危険**

いわゆるログインを許してシェルを使わせることは両刃の剣です。例えば顧客に対して管理のためのアクセスを許したいという場合、リモートからでもシステム資源などを使えるということは UNIX 系の最大の利点の一つとは思いますが、同時に悪用される可能性もあります。ログインに関しては、通常、ユーザ名とパスワードでログインを認証しますので、推測しやすいパスワードは極めて危険だということになります。
例えばメールなどの情報からすぐにフルネームとかを取り出せますし、そのフルネームの一部を組み合わせたようなパスワードなどはもっての他ということになります。

/etc/passwd

・暗号化したパスワードが保存される
**　シャドウ化すれば/etc/shadow に移される**
・一般ユーザでも読めるファイル
・パスワードは推測可能

・パスワードファイルのシャドウ化は必須

実際に不正アクセスの中でかなりの頻度を占めるのは、システムのパスワードファイルを盗もうというアタックです。特に CGI の古いバージョンのスクリプトを悪用する場合が多いのですが、パスワードがわかれば何とでもなるということです。

Red Hat もそうですが、たしか他のディストリビューションでもインストール直後のパスワードファイル、/etc/passwd はシャドウ化されておらず、暗号化されたパスワードフィールドが入っています。

・/etc/passwd の例

```
root:Xf4xQo72TYXgy:0:0:root:/root:/bin/bash
```

この例では、root:の後ろの英数字を組み合わせた部分が暗号化されたパスワードです。暗号化したものから生のパスワードを復元することはアルゴリズム的にできないにしても、いろいろな辞書などを使った推測は最近のコンピュータの処理技術、計算スピードを考えますと、現実的にある程度可能になってきています。この場合、名前や通常の辞書に登録されているような一般的な単語の組み合わせなどが推測の対象になります。

/etc/shadow

・シャドウ化すると作られるファイル

暗号化したパスワードなどが格納される

・一般ユーザでは読めない

・コマンド一発でシャドウ化が可能

pwconv5 (4.2)または pwconv (5.x)

・ソースで持ってきたソフトウェアでは個々にシャドウ対応が必要

まず推測しにくいパスワードを使うことは必要なことですが、同時にこのパスワードファイルを見せないという対応が現実的に必要になります。それがシャドウ化です。

Linux の場合、デフォルトではシャドウ化されていませんので、インストール後必ずシャドウ化すべきだと思います。シャドウ化を行いますと、先ほどの/etc/passwd ファイルの暗号化したパスワードフィールドはアスタリスクに置きかえられます。

また暗号化したパスワードフィールドは、/etc/shadow というファイルに移されます。この shadow というファイルはユーザ認証などに関わるファイルなので root 特権でしか読めない、一般ユーザでは読めないというパーミッションになります。

以前はシャドウ化には非常に複雑な対策が必要でした。といいますのは、単にこのようなファイルをつくるだけではなく、ログイン認証などをするプログラム全部がシャドウファイルを見に行くように書きかえる、コンパイルし直すという必要があったためです。

現在これは、Linux の Red Hat 社などが割と貢献しているのですが、PAM という機構があり、これを利用してコマンド一発でシャドウ化が可能になっています。Red Hat 4.2 の場合は、pwconv5(なぜ 5 なのかは知らないのですが)、Red Hat 5.x では pwconv というコマ

ンドでシャドウ化ができます。
ただし、PAM 対応になっていないソースで持ち込んだプログラムの場合は当然、個別に対応する必要があります。例えば、Qualcomm の qpopper は、たしか今シャドウ対応のみで、PAM 対応にするには、適当なパッチを使う必要があります。

ネットワーク上の盗聴

・ネットワーク上を流れるデータ(パケット)の盗聴は技術的に可能
・パスワードも盗聴の対象に
・インターネット上で暗号化しない(平文の)パスワードを使うのは危険

パスワードのシャドウ化は守りとして必須ですが、まだ不安は残ります。それはネットワーク上でパケットを盗聴されることがあり得るからです。ネットワーク上のパケットの盗聴は技術的に可能どころか、ネットワーク診断などの目的で商品化もされています。インターネットではパケットは、いくつかのプロバイダなどの中継点を通って流れていくわけですので、そのどこかが誰かに侵入されて盗聴プログラムを仕掛けられているという可能性も、少ないとは思いますが、否定できません。ですから事実上その可能性は常にあると思っていた方がよいと思います。
そうなりますと、先ほどの telnet の危険性が非常にわかりやすくなると思います。telnet でアクセスして、ユーザネーム、パスワードを入力してサーバに送り返すわけですが、それがネットワーク上のどこかで全部盗聴されているとすれば、パスワードを暗号化しないで送るということは、避けるべきだということになります。また同様に popper なども危険で、今は APOP というパスワードを直に送らないものが普及しつつありますので、そちらを使うべきです。

telnet

・リモートからのログインプログラム
・ログインすればシェルがただちに使える
・ユーザ名とパスワードは暗号化されない
・インターネットに telnet アクセスを開放するのは、きわめて、きわめて危険
・パスワードなしでリモートログインやリモートコピーを行う r-cmd はさらに危険

telnet に関してもう少しつけ加えておきます。これは遠隔ログインを行い、相手が UNIX の場合はシェルなどを使えるという通信プロトコルですが、古き良き時代、そういう危険が比較的少なかった時代のものであり、ユーザ名とパスワードは暗号化されていません。つまり、インターネットに対して telnet を露出する、開放するのは自殺行為に近い危険なことだといえます。
また、実際にネットワーク越しで作業するためには、telnet よりももっと便利な rlogin や rcp などというコマンドがあります。これはホスト間の信頼関係により、無条件で受け付け

るホストを登録しておけば、パスワードすらなしで処理を受け付けるというものです。これは言うまでもなく telnet よりもさらに危険だということになりますのでインターネット側に無条件に出すというのは完全に論外です。telnet はケース・バイ・ケースで、どうしてもやむを得ない場合もありますが、r-cmd 類は論外ということになります。

一般ユーザからroot へ

・いくつかの方法がある

/etc/passwd でパスワードを推測する

シャドウ化が有効な対策

侵入後、既知のコマンドのバグを悪用

セキュリティホールはいろいろある

すべてに完全に対応するのは必要だが難しい

これも知っておくべきことだと思いますが、一般ユーザから root になるには、いくつかの方法があります。正規に su というコマンドを使ってスーパーユーザ、root 権限を得るというものは当然ですが、ここでは不正アクセスなどの場合に一般ユーザでログインして root になる方法がいくつかあるということです。

root のパスワードが推測されていれば、当然ながら正規の手順で入れます。また、セキュリティホールということでよく報告されていますが、ログインした後に X などいろいろなローカルなプログラムのバグ、セキュリティホールを突いて root 権限を得るというものもあります。ですので一般ユーザでもパスワードを盗まれるというのは、システムにとって非常に危険だということになります。

ワンタイムパスワード(OTP)

・複数のパスワード(パスフレーズ)を生成しておく

・ログインのたびに新しいものを使う

・OPIE、S/Key などが利用可能

opie-2.22、logdaemon-5.6 が最新

・RPM 化したパッケージはまだ作られていない

登録されているパスワードそのものを流さないための極めて有効な方法として、２つ代表的なものがあります。

一つはワンタイムパスワードです。これはある一定のアルゴリズムに基づいて、毎回異なるパスフレーズ(パスワード)を使うというログインのやり方です。

その違うパスワード、パスフレーズはメモしておくとか、別のプログラムでジェネレートして入力するのですが、RPM 化したパッケージは今のところなかったのではないかと思います。

Secure Shell (SSH)

・通信内容すべてを暗号化

・事前にアクセスを認め合ったホスト間で使う

・シェルアクセス、リモートコピーが可能

・ライセンス上商用利用には使えない

・ssh-2.0.9 が最新

・RPM 化したパッケージも FTP で入手可能

もう一つは、先ほども説明しました SSH（Secure SHell）です。これは基本的には telnet と似たことを全部実現してしまうものですが、パスワードのやり取りや、サーバから送られるプロンプトなどもすべて含めて、通信内容を丸ごと暗号化してしまうというところが特徴です。また r-cmd にも似ていますが、事前にアクセスを認め合ったホスト間で使うというのが前提です。

これらを組み合わせることで、シェルのアクセスや rcp のようなリモートコピーが、かなり安全性を高めた暗号化した経路で使えるようになります。ただし商用利用では商用版を使う必要があり、プライベートな利用でのみオープンソース版を使えるというライセンスになっています。利用および提供形態を考慮する必要がありますが、プライベート利用に関しては RPM 化したパッケージが入手可能になっています。

OTP v.s. SSH

・パスワード盗聴防止にともに有効

・OTP の利点

使い捨てパスワード系列がわかっていれば、出先のホストを借りて
ログインすることも可能

・SSH の利点

通信内容全体を暗号化するので、パスワード以外の秘密も守れる
rlogin、rsh、rcp と同じ操作性で使える

・OTP の欠点

通信内容は保護されない

・SSH の欠点

あらかじめ認証しあったホスト間でしか使えない

・必要に応じて両者を使い分け、telnet でのリモートログインを禁止するのがよい

これらの特徴について比較しておきます。まず、どちらもパスワード盗聴防止には有効です。また OTP にはホスト間の事前登録が必要ありません。例えば、DHCP でランダムに割り当てられる IP アドレスの端末から、会社のホストにアクセスするというときでも使えます。これは大きな利点であると思います。

SSH の利点は通信内容全体が暗号化されることです。またパスワード以外の機密の保護に

も役立ちますし、一度認証関係を確立しておけば、r-cmd と同様な操作がすべて行えると考えて結構です。ですから telnet のかわりではなくて、rlogin、rsh、rcp、これらと同じ感覚ですべての r-cmd が使えるということになります。実際に SSH を使うときのコマンドは r-cmd の先頭の r を s に変えたものになっており、リモートコピーならば scp というコマンドになります。
一方、欠点として重要な理解しておくべきポイントは、OTP は認証で生のパスワードを流さないというだけのもので、通信内容自体は暗号化されない、ということです。SSH の欠点は先ほどの OTP の利点の裏返しですが、あらかじめ認証しておくことが必要で、出先のマシンを借りて使うというわけにはいかないことです。
固定的なマシン間では SSH の方がよいと思いますが、出先の端末から使うこともあるならば、組み合わせて使うことになると思います。
パスワードに関してまとめておきますと、まずパスワードは直ちにシャドウ化するということで、Linux の場合これは必須です。もうひとつは、特に telnet に関して慎重になった上で、できれば SSH または OTP を使うべきであるということです。

スーパーデーモンinetd

・いくつかのサービスを一括して監視
　telnet、ftp、pop、imap、finger など
・クライアントからのアクセスに応じて、実際のサーバプログラムを起動
・/etc/inetd.conf で挙動を制御

続きまして、スーパーデーモンという inetd について説明します。Linux を立ち上げて ps などで動きを見ているとわかりますが、例えば sendmail、WWW サーバ、httpd などというものはブート時に立ち上がって常に待機している、常駐デーモンになります。ところが、例えば pop や telnet などのサービスをするサーバ機能は、常時立ち上がってはいません。アクセスがあって初めて立ち上がるという仕組みになっており、これを制御しているのが inetd です。
流れを説明しますと、telnet などのアクセスが来ますと、この inetd がこれらのアクセスを一括して受け取り、必要に応じて実際の処理をするサーバプログラムを起動するというようになっています。inetd の挙動を制御するファイルが inetd.conf です。

/etc/inetd.conf

・1 行が 1 つのサービスに対応
・行頭に#が付いていない行のサービスが提供される
・提供しないサービスはコメントアウトする
・相手によって禁止/許可するサービスは tcp wrapper を使う
　/etc/inetd.conf

・たとえば imap を使っていないなら、次のように書き換える

```
imap     stream tcp nowait root /usr/sbin/tcpd imapd
↓
#imap    stream tcp nowait root /usr/sbin/tcpd imapd
```

・変更を有効にするには、inetd に SIGHUP シグナルを送る

inetd.conf では、１行が１つのサービスに対応しています。例えば、telnet について１行、pop について１行という形になっています。他の設定ファイルと同様、#記号がついている行はコメント行となります。この inetd.conf は、現在ではすべてのディストリビューションで TCP wrapper 対応になっている思います。

例えば imap は、デフォルトで立ち上がるような設定になっているかと思いますが、もし imap が不要でしたら、頭に#の文字を入れるだけで imap は提供しないということになります。ただし、intetd.conf を書きかえただけでは inetd の動きは変わりませんので、ハングアップ（SIGHUP）のシグナルを送って、この変更を再度読み込ませる必要があります。

TCP wrapper (tcpd)

・実体プログラムは /usr/sbin/tcpd

・挙動は /etc/hosts.allow、/etc/hosts.deny で制御

・設定ファイルの書き方は 2 種類

　拡張記法だと/etc/hosts.allow だけを使う

・man 5 hosts_access を参照

次に TCP wrapper について説明します。

先ほどの inetd.conf の例ですが、これが TCP wrapper を使うという記述になっています。この行の簡単な基本的な流れをいいますと、まず inetd は imap のアクセスを監視します。クライアントから imap でのアクセス要求が来ますと inetd はまず TCP wrapper の本体である/usr/sbin/tcpd というプログラムを起動します。tcpd はアクセス制御のためのプログラムで、アクセス元の IP アドレスなどを照合して、imap のアクセスを受け付けるかどうかを判断します。受け付けてよい場合 tcpd は、右端に書いてある実体の処理プログラム imapd を起動するという流れをとります。

今度は TCP wrapper に話を移しますと、TCP wrapper の実体は、今説明しました tcpd というプログラムです。tcpd の挙動は、/etc/hosts.allow と/etc/hosts.deny というは２つのファイルで制御されます。実は、設定ファイルの書き方には２種類ありまして、ベーシックな書き方と拡張した書き方があります。

私が好んで使っているのは拡張の方で、deny のファイルを使わずに allow の方だけですべての設定が書けます。雑誌などでは両方を組み合わせるケースが多いようですが、１個で済むほうが全貌がつかみやすいので、私はこちらの拡張の方がよいと思っています。

/etc/hosts.allow の例

いくつかの設定を例をあげて説明します。

・すべてのアクセスを許可

```
ALL: ALL: ALLOW
```

まずデフォルトの設定は、何も書いていなければこれと同じになります。これはあらゆるホストからのあらゆるタイプのサービスをすべて認めるということです。つまり、inetd.conf に書いたすべてのサービスが無条件に有効になります。

・内部ネットワーク(192.168.0.0/255.255.255.0)からのすべてのアクセスを許可し、
それ以外はすべて拒否

```
ALL: 192.168.0.0/255.255.255.0: ALLOW
ALL: ALL: DENY
```

次に、例えば内部ネットワーク、社内側、学内側、そちらのネットワークからのアクセスはすべてのプロトコルサービスについて許可し、それ以外はすべて拒否するという設定を書くには、このように書きます。

左端の ALL がサービスのタイプです。telnet、ftp や pop などをここに書くこともできます。192.168.0.0/255.255.255.0 というのがアドレスの範囲ですが、この場合は右端にあります ALLOW というキーワードによってアクセスを許可するということになります。

まず１行目が評価されて、それに合致しないアクセスには２行目の「ALL:ALL: DENY」が適用されますので、指定以外のアドレスからのアクセスはすべて拒否するということになります。

書き方は極めてシンプルですし、アドレス表記の部分は、スラッシュで区切ってネットワークアドレス、ネットマスクを書くという以外に特定の IP アドレスを書くこともできます。他にも特定の IP アドレスやドメイン名で書くなど、拡張はいろいろありますので、実際にマニュアルをごらん頂きたいと思います。

・拒否したアクセスは管理者にメールで通知

```
ALL:     ALL:  ＼
spawn (/usr/sbin/safe_finger -l @%h | /bin/mail -s ＼ "%d-%h" root) &:¥DENY
```

先ほどの設定に加えて、アクセスを拒否した場合に管理者にメールで通知できます。先ほどの DENY の部分を書きかえるという例ですが、単に拒否してしまうというだけではなくて、何らかのアクションを起こすことができます。
この場合には先ほどの最後の行、ALL:ALL、単に DENY にするのではなくて、間にもう一個アクションをコマンドとして記述します。
この場合は finger で、アクセス元に対する情報収集を行って、その結果を管理者、ここでは root に対してメールで送るということを行っています。
これをやりますと、間違えたケースも含めてログインに失敗したケースの報告が結構大量に来ます。最近よくあるのが、限定的に認めている telnet や popper などについて、大体２回ワンセットぐらいでポートスキャンに来ているという場合で、非常によくわかるようになります。

・サービスごとに制御

```
ALL: 192.168.0.0/255.255.255.0: ALLOW
in.ftpd: ALL: ALLOW
popper: 210.123.45.67: ALLOW
ALL: ALL: DENY
```

今度は特定のサービスごとに制御していきたいという場合です。
まず、内部ネットワークからのアクセスはすべて認める。その次の行は、先ほどサービス名と説明しましたが、実際は起動される/usr/sbin の中のプログラム名、ファイル名が必要です。ftp とか ftpd と書くと動きませんので、実際のプログラムのファイル名を書きます。in.ftpd は、ftp はすべて許可するという設定です。その次の行は popper で、これも popper、popd など、実際の使っているファイルに合わせる必要がありますけれども、この場合は popper です。
この例では、特定の固定的なアドレス(210.123.45.67)からの popper は許可します。
それ以外は、この場合では内部ネットワークを含めて拒否ということになります。
それまでの条件で許可した以外のものは、最後の、"ALL: ALL: DENY"という行ですべて拒否することになります。
例えば、特定の顧客のアドレスが特定しきれるものであれば、popper の部分を in.telnetd にして、そのアドレスを指定すれば、そのアドレスからのメンテナンス用の telnet を許可することは可能です。たったこれだけのことですが、内部で必要な telnet は生かしたままで、外部からの telnet を効果的に封じることができます。しかし、これすらやっていないサイトが非常に多いのというのが現状です。

Phf スクリプト(WWW サーバ)

- phf スクリプトはシェルコマンドを実行する
  最近のパッケージには入っていない
- /etc/passwd ファイルなどの窃取に悪用される
- /var/log/httpd/access_log で確認できる

404 ならば大丈夫

```
... "GET/cgi-bin/phf? ... 404 -
```

次にアクセス制御というか、パスワードに絡んだ重要なポイントをもう一つ説明しておきます。それは CERT や JPCERT などの Advisories によく出てきている Phf スクリプトです。最新バージョンの WWW サーバ、例えば Apache などにはもう入っていないのですが、古いバージョンの WWW サーバには Phf というスクリプトがあり、任意のシェルコマンドを送り込んで実行することができるようになっています。

一番多いパターンが、シャドウ化されていないことを期待してパスワードファイルを盗もうというアクセスです。先ほどのポートスキャンのデフォルトのパターンになっているようですが、httpd の access_log にこのようなパターンが入っていれば、このようなアタックを受けているということになります。

定期的に見ないとわかりませんが、access_log の右端に、処理のリターンコード、リザルトが出ています。400 番台というのはアクセスの拒否を意味しており、その場合は OK ということになります。比較的新しい httpd のパッケージをインストールしている場合は、このアタックはリジェクトしているだろうと思います。

ところが実際に WWW サーバ、特に CGI を話題にすると、それだけで半日や１日ぐらいかかるほど、CGI そのものが潜在的にいろいろな危険要素を含んでいます。いわゆるプログラムを書くわけですから、その書き方が悪ければ、クラッカーにとって有益な情報を提供するような穴をいっぱい持つことになります。

ですから、WWW サーバを立てる場合、CGI に関しては完全にそちらの専門的な参考書などでセキュリティのことを勉強する必要があると思います。といいましても、WWW サーバの作り方や CGI の書き方の本は多いですが、セキュリティの盲点などまで書いてある本というのは、まだあまり多くないと思います。

ブート時に起動されるサーバ

・常時サーバを待ち受けるサーバ(デーモン)もある

DNS サーバ (named)

メールサーバ (sendmail)

WWW サーバ (httpd)など

・システムのブート時に自動的に起動される

アクセス制御としまして先ほど説明しましたのは、inetd 経由で立ち上がるサーバです。今度はブート時に立ち上がって常駐するサーバについて、間違えてインストールしてしまった場合や、インストールして動いているが、もう必要ではないという場合について説明します。常時クライアントからの接続を受け付けるサーバもあり、典型的なものが DNS のための named や、メールサーバの sendmail です。これらはシステムのブート時に、一定の手順で自動的にスクリプトで起動されます。このあたりは Red Hat、Slackware などで相当異なるところです。

ブート時の起動を停止する

・ /etc/rc.d/rc3.d/に起動スクリプトが存在

・ファイル名の先頭文字が S(大文字)

S80sendmail など

数字は起動順序を表す

・ブート時に起動しないようにするには

ファイルを削除(勧めない)

先頭文字を "S"、"K"以外にリネーム

いろいろなサーバを立ち上げるスクリプトは、全部この「/etc/rc.d/rc3.d/」というディレクトリに入っています。この rc3 という部分は、実際にはユニットタブに書く起動時のランレベルの番号になるわけですが、サーバとして X や xdm を動かさないという場合はランレベルは通常は３になりますので、このディレクトリを見ればよいということになります。

このディレクトリを確認しますと、その中には複数のファイルが入っています。

多くのファイルの名前は、S で始まり、次に数字、それから内容を表す文字という形式になっています。S は Start の意味で、ブート時に起動するということを意味します。次の 80 というのは、他に 10 とか 20 とかいろいろなファイルがあり、いわゆるソート順になっています。つまり起動順序がソート順になるということです。

S10 は基本的なネットワーク機能そのものの立ち上げです。これは他のすべてのサービスに対する基盤になりますので 一番小さな番号を使っています。

例えばメールサーバではないのに間違えて sendmail がインストールされて起動されるようになっているときは、先頭の文字 S をリネームしてしまえばいいということになります。

ただし K というのは Kill という意味で、他のランレベルから移行してきたときにデーモン

を殺す、落とすという意味がありますので、S または K 以外の文字に変更すればいいということになります。例えば私が好んで使うのは、「_」や「s」にするなどです。

デーモンの起動と停止

・デーモンはコマンドで起動、停止できる

/etc/rc.d/rc3.d/ファイル名 [start | stop]

このようにすれば、次回のリブート時からメールサーバは上がらなくなります。ただし何か変更するたびにリブートするというのは Windows の癖で、Linux ではそのようなことはほとんど必要がありません。デーモンはコマンドで起動、停止できます。

例えば sendmail の場合では、コマンド、S80sendmail にパラメータとして stop をつけて実行すれば、その場で落とせます。これはリモートメンテナンスなどの場合に非常に役立ちます。最近、5.2 ではリスタートというパラメータも追加されており、start、stop、restart となっています。

例えばリモートでメンテナンスするときには、相手方の sendmail などをバージョンアップしてリスタートすることが可能になり、実際にリブートなどは、ほとんど必要がないということになります。

カーネルのカスタマイズ

・Red Hat 5.2 では通常は不要

・Red Hat 4.2 では、ファイアウォール機能などのために必要

・不要な機能を削るのは好ましいこと

余分なサービスをしなくなる

カーネルがコンパクトになる

以前は、カーネルのカスタマイズは、従来の Red Hat 4.2 やそれ以前のものでは、ほぼ必須と考えていました。インターネットサーバなどに使う場合は、ネットワークドライバの対応、余分な機能の削除、またファイアウォールでは IP マスカレードなどの機能を加える必要があったためですが、最近の 5.2 ではどうも、ある程度不要になっているようです。

もちろんセキュリティ対策やプラスアルファの独自のものでは、相変わらず必要だと思いますが、通常は要らないのかもしれないという気がしています。ただし、カーネルをカスタマイズして不要な機能を削る、またはネットワークドライバなどが不安定ならば新しいモジュールに取り替えるということは、必要であったり、好ましいことであったりします。

特に不要な機能を削るという観点でのカスタマイズには、２つの利点があります。まず、不要な機能が悪用されるという危険性がそれだけ減るわけで、もう一つはカーネルサイズ自体がコンパクトになる、ということです。しかし、Red Hat 5.x になってからカーネルのカスタマイズ自体がかなり難しくなっているようですので、内部構造を十分に調査して理解しておかないと、昔の感覚だけでは済まないようです。カーネルのコンパイル自体は通

っても、その後、実際に認識させるまでが複雑なようです。

ファイアウォール

・一般に外部からのアクセスを禁止すべきサービス

tftp	69/udp
finger	79/tcp
sunrpc	111/tcp,111/udp
netbios	137-139/tcp
snmp	161/udp
exec	512/udp
login	513/tcp
shell	t14/tcp
	など

・telnet、imap **などもファイアウォールで禁止しておくのが望ましい**

・アクセス制御に有効だが過信は禁物

適切な設定とメンテナンスは不可欠

ファイアウォールに関して、先ほどポリシーということを説明しましたが、これはあくまでもポリシーの中のアクセス制御の一部分を効果的に進めるためのものだと考えて頂きたいと思います。ファイアウォールはアクセス制御をきちんとやるためには極めて有効です。しかし、その設定に穴があったり、ファイアウォールを設定していても telnet **を認めているという場合が多いようですが、そのパスワードがザルであったりすると全く意味がなくなります。ですので、適切な設定とその後のメンテナンスが不可欠です。**

・ Linux **カーネルもファイアウォール機能を持つ**

パケットフィルタリングと IP **マスカレード**

ipfwadm **コマンド**

Red Hat 4.2 **ではカーネルのカスタマイズが必要**

ご存じのように、Linux **はファイアウォールとしても使えます。パケットフィルタリングと** IP **マスカレードという非常に強力な機能をカーネルレベルで持っており、これらを設定するのが** ipfwadm **コマンドです。これもカーネルが今度** 2.2 **になると、コマンドを含めていろいろ動きが出てきますけれども、このようなファイアウォール機能を使うことができます。私のところでは商品としまして、このようなカーネル自体のファイアウォール機能をベースにして、ファイアウォールとメールサーバなどのサーバ機能をすべて一体化した、いわゆる「オールインワン」というものを作成しております。ただし** 4.2 **では、このファイアウォールは最初についてくるカーネルには入っておらず、カスタマイズが必要です。**

・一般的に外部からのアクセスを禁止すべきサービス
　telnet、imap などもファイアウォールで禁止しておくのが望ましい
買ったファイアウォールを使った場合でも同様ですが、ガイドラインとして少なくとも先に述べたようなプロトコルのアクセスは禁止すべきと思います。また telnet などの不要なサービスはファイアウォールのレベルでも禁止するほうがよいと思います。

・「オールインワン」も構築可能だが…
　・サーバ自体にファイアウォールを組み込む
　・コストに制約がある場合は有効
　・セキュリティ上はお勧めできない
オールインワンというサーバ構成ですが、これは一つの機械にファイアウォールを含めてメールサーバや DNS などを全部入れ込むというものです。実際に動きますし、何もセキュリティ対策をとらないのに比べると格段にセキュリティ的に強化されたサーバが作れます。要するにサーバ自体にファイアウォール機能を組み込んでしまうということは、現実的な対策としてあると思います。しかし、それはコスト的な面で制約がある場合に限る方がよいとも思っています。やはり原理的にオールインワンのサーバでは、そのファイアウォール部分が破られるとサーバが直ちに危険にさらされます。また、直下にある内部ネットワークなども同様ということになるからです。セキュリティ上はやはり、ある程度分けた方がよいと考えていますが、 実際にコストや運用管理の手間ということで、オールインワンもまだまだニーズがあるのも事実です。

ルータのパケットフィルタリング機能、ファイアウォール製品も検討すべき
どうしてもオールインワンを使うならば、もう一個押さえておくべき点があります。最近はルータ側でも、パケットフィルタリングや NAT/IP マスカレードなどを備えていますので、これらと組み合わせることでファイアウォールの多段構成も可能だということです。またプライベートアドレスの利用も、セキュリティ対策上の観点から有効です。

・ IP マスカレード
　任意の数のプライベートアドレスが利用可能に
　ファイアウォール機能としてもきわめて有効
・実用上十分だが一部のプロトコルに制約も
グローバルアドレスには当然、割当てに制約があり、例えば OCN では 16 個しかもらえません。IP マスカレードにより IP アドレスを変換し、内部ではプライベートアドレスを使うようにしますと必然的にこのような制約から逃れることができます。
IP マスカレードはファイアウォール機能として極めて有効な特徴、性格を持っており、外部から内部への不正アクセスをはじくことができます。しかし残念ながらいくつかの制約

もあり、内部から外部の方はかなり自由にアクセスできますが、例えば streamworks などのマルチメディア系や、UDP を使ったいくつかのプロトコルが通りません。ただこれも先ほどの利便性と制約のトレードオフということで考えていけば、一般のビジネスや通常のサイトに関しては、この IP マスカレードで必要十分であると思います。

5. システム運用管理

サーバの運用やセキュリティ対策に「終点」はありません。主にセキュリティ
対策の観点から、日常の運用管理をメニュー化して説明します。

日常の運用管理項目

・システム資源（ディスク、メモリの使用状況、プロセス管理）
・ユーザ管理（ユーザの追加削除、パスワード管理）
・データ管理（バックアップ）

サーバを作って最初の対策を立てたとしても、セキュリティのレベルを保つためにはその後の運用管理が重要です。日常の運用管理として一般的なのは、システム資源、ユーザ管理、データ管理ということですが、セキュリティ関連ではやはりまず、不正アクセスを受けているかどうかの確認が必要です。様子を見ている程度のアクセスで、それをはじいているということがわかれば、安心していてもよいのですが、その不正アクセス自体がその後の重大なものにつながることがあります。ですから、アクセスを受けているかどうかを検出しておくことは重要です。それともう一つは、システムファイル自体の改竄の検出が極めて重要なのではないかと思います。また、システムの設定内容のバックアップや、データ部分は毎日とるというようなことは最小限のセキュリティ対策の心がけになるかと思います。

セキュリティ関連の管理項目

・「身の丈」でもやっておきたいこと
　不正アクセスの検出
　システムファイル改竄の検出
　バックアップ
・チェックの自動化は管理を楽にする
　管理パターンが決まってきたら、スクリプト化してみよう

運用パターンが大体決まってきてェックする項目も決まってくれば、ある程度それぞれのサイトのニーズに合わせてスクリプト化していくこともできるのではないかと思います。例えば私どものところでは、全国で 20 数カ所のサーバをリモートで見ております。スクリプト化ということでは、ディスクの使用率やメモリの使用状況、その他の一般的な状況、外部からのアクセス状況などをスクリプトで簡単に加工して、メールで毎日受け取って中身を吟味するということをやったりしています。それを必要に応じて改良していくわけですが、そういうことをやっておきますと、運用管理が非常に楽になってきます。また、見逃すということもある程度減らせるようになると思います。私どもの場合でも今後数が増えていくとメールのレポートを生で受け取っていたらたまらないので、データベースにしまえるような形にしていこうということも検討したりしています。手動と併用ですが、このようにチェックや運用管理をできるだけ自動化していくというのが重要かと思います。

ログの点検

・/var/log/messages

もっとも多くの情報が書き込まれる

FAIL、INVALID **などのパターンに注目**

・/var/log/secure

ログイン履歴などが集められる

refuse、warning **などのパターンに注目**

ログイン履歴は last **コマンドでも把握すべき**

・/var/log/maillog

メールの履歴が記録される

このファイルの分析は難しい

・/var/log/httpd/access_log

WWW **サーバへのアクセス履歴**

" 40"、"phf"**などのパターンに注目**

ログ点検のポイントですが、まずシステムのアクティビティのログは大体、多くの情報が /var/log/messages **に書き込まれます。セキュリティという点では、そのログのメッセージ中で例えば** FAIL **や** INVALID **という文字列が出てくる行は、アクセスに失敗したなどということです。例えば** grep **などでこのような行を探してみるだけでも有効ですし、もう少し網を広げて、外部からのアクセスでは、**ftp **などの文字を拾い出していけばよいと思います。このログは単純に眺めるには分量が多いので、**grep **などを使った自動化は必須だと思います。**

また、/var/log/secure **というものがあります。**messages **と内容的に重複することはありますが、こちらの方にはログインの履歴などが集まっています。やはりここでも** refuse **とか** warning **という文字列、パターンに注目していけばいわゆる不正アクセスの検出に役立つかと思います。**

ログインの履歴についてはもう一つ、last **というコマンドで表示させることでも把握できます。**

ただし、このような文字は見つからない、last **などでも異常がないと思っても、ログが改竄されていたということもありますので、チェックしているログファイル、ソース自体の信頼性はまた別の話です。また、**maillog **には、メールの履歴が記録されています。このファイルは１行がメールの** to **や** from **などに対応していますが、**to **と** from **は１対１とは限らず、**cc **の場合は１対多になります。このファイルの分析はまじめにやろうとすれば結構手間を食いますが、メールの不正中継などの場合に、**"we do not relay"**などというような情報が記録されることもあります。また当然ながら、眺めてみることによって、見覚えのないところから見覚えのないところへ非常に多数のメールが送られているというログが見つかったら、これは不正中継に使われたということになります。**

syslog で書き込まれるログとは別ですが、WWW サーバの access_log も重要な分析対象になるかと思います。もしも grep で引っかけるならば、" 40"、これはステータスコードですが、あと phf などというパターンが出てきているかどうかという点です。これらは不正アクセスを表すパターンの典型例だと思います。

ログ監視の自動化ツール

- ログファイルモニタ swatch
- 常駐してログファイルをリアルタイム監視
- パターンを検出したらメールなどで通知
- 定義ファイルの例

```
/FAILED/        mail=admin
/INVALID/       mail=admin
```

システムファイルの改竄監視

- 設定ファイル、実行ファイルは頻繁に書き換えられるものではない
- 管理者が知らない書き換えは、不正侵入の恐れを示す
- Tripwire が有名
  - 指定したファイルの「指紋」のデータベースを作る
  - 定期的に実ファイルの「指紋」と照合する

ログというのはどんどん黙々と書き加えられる、ためていかれるものですが、そのログ監視自体をできるだけ、不正アクセスその他のイベントがあったらリアルタイム的に検出したいというときに使えるのが swatch というソフトです。ログファイルをディスクに書き出すと同時にリアルタイム的に見てくれますので、これは一種の常駐プログラムになります。パターンを検出したら、メールを出したり、ポケベルやサイレンみたいなものをつけておいたらそれを鳴らすとか、そういうアクションをいろいろ定義できます。例えば、ディスクフルやどこかのデバイスが壊れた、などというものもこれで検出させて、管理者のポケベルを鳴らすようにしたり、さまざまな設定をルールとして定義できます。
極めて単純な例ですけれども、定義ファイルであるログ、例えば/var/log/messages をチェックして、FAILED とか INVALID という文字、これは perl の正規表現ですが、が見つかったら、admin というアドレスあてのメールを送るということを定義したりします。
このプログラムは、リアルタイム的にログを見てくれるという意味で非常に役立ちますが、最大の問題点は「perl」です。perl で動かしますので、perl というインタプリタ自体を入れることに対して、クリティカルにならざるを得ない場合があるということです。
それと、もうひとつは perl がメモリを消費するということで、perl を１個動かすのについて１ M やそこらのメモリは覚悟しておいた方がいいかと思います。
現在は、最小構成でも 64M とか 128M ぐらいのメモリがありますので、大きな影響は少な

いと思いますが、古いマシンを使う場合には問題になるかもしれません。
もう一つ、先ほど言いましたようにログを grep などでいろいろチェックするのですが、ログ自体が不正にいじられていないという保証が必要になります。ログなどがいじられるという不正アクセスを受けた場合には、一般的にその他のシステムファイルも書きかえられます。システムファイルとは、例えば、/etc/の設定ファイルや/bin/や/sbin/の実行ファイルですが、これらは頻繁に書きかえられるものではありません。通常は、例えばパスワードコマンドをユーザが実行してパスワードを変更するなどという例外的なことを除いて、システム管理者が知らないうちに書きかえられるということはまずあり得ないはずです。ですから、管理者が知らないシステムファイルの変更というのはこれは不正侵入、不正アクセスの可能性を示すわけです。
システムファイルの改竄をチェックするというソフトとしては Tripwire というのがあります。この動作原理的は、まず、指定したディレクトリの全ファイルに対して、「指紋」に当たるようないくつかのチェックサムのデータベースをつくります。
この指紋に当たるチェックサムのアルゴリズムを複数組み合わせることで、改竄された場合に指紋が偶然一致するという可能性を、極めて小さく、事実上ゼロにしています。
この保存しておいたチェックサムのデータベースと現在の値を照合すれば、改竄、修正が検出できるという仕組みになっています。
ですので、Tripwire などシステムファイルの改竄の監視を行い、その上でログをさらに見ていくということになります。

バックアップ

・インストール直後のバックアップ
　システム全体(万一の修復が楽になる)
　/etc/ (初期設定値、とくに重要)
・定期的なバックアップ
　/etc/ (設定を変更したとき)
　システム全体(ソフトウェアをアップデートしたとき)
・ユーザデータ

最後に、セキュリティ対策としても重要なものはバックアップです。インストール直後のバックアップをとっておくというのは、万一やられたときの修復が楽になります。
Red Hat の場合は初期設定値として/etc/の下をコピーしておけばいいだろうと思います。
また定期的なバックアップとして、設定を変更したら/etc/の下をバックアップしておき、後はセキュリティ以外の通常の日常の運用管理と同じようなことになると思います。
/etc/の下をバックアップしておくことによりシステムの設定そのものは大体保護されますが、例えばメーリングリストを運用していると、送付先アドレスのリストなどは/etc/の下ではなく/var/の下にあったりします。ですから、そのあたりも考慮したバックアップをと

る必要があります。

6. 情報収集の方法

日常の運用管理で、セキュリティに関する脅威、動向を把握して反映していくということは欠かせないと思います。ここではそのための情報源をいくつか紹介します。

CERT

・http://www.cert.org

・コンピュータセキュリティに関心を持つインターネットユーザの情報集約センター(1988 年設立)

カーネギーメロン大学に設置

CERT Advisories というメーリングリスト

セキュリティ上の問題と対策を速報してくれるサービス

過去の CERT Advisories は FTP で公開

過去のログ

CERT Advisories (ftp://info.cert.org/pub/cert_advisories/)

CERT Bulletins (ftp://info.cert.org/pub/cert_bulletins/)

検索ページ

CERT Advisories (http://www.voj.toda.saitama.jp/cert-ca.shtml など)

CERT Bulletins (http://www.voj.toda.saitama.jp/cert-vb.shtml など)

まず CERT です。これは 1988 年に設立されてカーネギーメロン大学に本部があります。CERT Advisories というメーリングリストを運営していまして、特にこれはセキュリティ上の問題や対策についての情報を、速報ではない、ちょっと遅いという声もありますが、知らせてくれるというメーリングリストがあります。過去の CERT Advisories のバックナンバーは ftp で公開されています。

情報処理振興事業協会(IPA)

・http://www.ipa.go.jp/index-j.html

・ウィルス、チェーンメールなどの情報も対象とした

コンピュータセキュリティ対策のページがある

(http://www.ipa.go.jp/SECURITY/index-j.html)

日本の公的な機関という意味で２つ説明します。

一つは情報処理振興事業協会(IPA)です。もちろんコンピュータセキュリティ全体扱っていますが、特にウィルス関係などの情報もいろいろ集計して公開しています。

ここではインターネットセキュリティ、それも Linux を使ってサーバを作ることを主眼に置いていますが、クライアント環境のことを考えますとウィルスというのも非常に重要な

問題点になってきています。といいますのはこの IPA の統計では、最近のウィルスは大半が電子メール経由で、Excel や word の添付ファイルを通じてマクロウィルスが伝わるというものになってきているからです。

コンピュータ緊急対応センター (JPCERT/CC)

・http://www.jpcert.or.jp/
・不正なシステム侵入に対する緊急対応を中心に、インターネットセキュリティの情報収集・分析、再発防止策の検討、セキュリティ技術の教育・啓発活動を行っている組織
・「情報提供用メーリングリスト」も運営

もう一つの団体としましては、日本での CERT のような JPCERT/CC があります。
不正なシステム侵入に対する緊急対応や緊急対応のアドバイス、それからそういう事例の集計、教育・啓蒙活動などを行っています。JPCERT のメーリングリストを運用しておられます。

メーリングリスト

・linux-security-jp
　とくに Linux ユーザを意識してセキュリティ関連の話題を扱っている
　CERT Advisories その他の情報もフォワードされている
　http://www.3ware.co.jp/opensoc/index.html

私の方でも linux-security-jp というメーリングリストを運用しております。話題としては Windows の話題が入ってもよいのですが、Linux のユーザをある程度意識して、セキュリティについて話し合うという目的で運営しています。現在大体 7,800 名ぐらいのメンバーが登録しておられます。先ほどの JPCERT や CERT の Advisories などが出ましたら、このメーリングリストを通じてフォワードするようにしていますので、興味のある方はメンバーになって頂ければと思います。
他にもメーリングリストやセキュリティに関する情報は山ほどあります。例えばオーストラリアの CERT は数人で運営しているそうですが、非常に活発ないい情報を出しているという話ですし、その他にも COAST とか団体のセキュリティ関係の情報はいっぱいあります。ただ、そういうのを全部「身の丈」で追いかけるのは不可能だと思いますので、最小限このあたりをご紹介しておきます。

アップデートモジュールの入手

・Red Hat 社の FTP サーバから入手可能
　ftp://ftp.redhat.com/
　世界中の FTP サーバでもミラーされている

・CERT Advisories などを通じてアップデート情報が入手できる

セキュリティ関係を含むアップデートモジュールは Red Hat の FTP サイトから入手できます。また先ほどの Advisories などでも、例えば Red Hat のアップデートモジュールは、この URL にあるということを書いたりしていますので、メーリングリストなどの情報で入手方法を知ることが可能と思います。

7. 最近のセキュリティ動向から

phf スクリプト、ポートスキャン、不正侵入、サービス不能攻撃、メール不正中継など、最近のアタック動向と対策を紹介します。

現時点からは、本質的に新手のものは出てきておりませんので、このあたりが昨日今日でも実際に起こっているものだと思います。

ポートスキャン

・最近もっとも多い不正アクセス

・telnet、pop3、bind などのポートをチェック

・ポートが空いていたら別のツールでさらにアクセスされることがある

まず最近一番多いものが、ポートスキャンと呼ばれるもので、これはアクセスの中でも、探りを入れてみるというレベルのものです。主にこれが狙ってくるのは、telnet、pop、bind、imap、それから先ほどの phf のスクリプトなどのポートやサービスです。

ポートスキャンには、mscan またはマルチスキャンなどという名前のツールがあり、デフォルトのままでも telnet や pop などをセットにしてチェックするということになっています。そのツールの readme、使い方には JP ドメインを例にしたサンプルが書かれているということで、特に日本も狙われているということになります。

アクセス元でドメインがわかったものを見てみますと、世界中から来ています。こちらから逆にポートをたたいてみようということで、telnet のポートにアクセスしてみたら、Linux というのが現状は大半です。例えば bind のセキュリティホールを突く場合に Linux のサーバを攻略するためのプログラムになっているものがあり、結局正しいサーバで使われるのも、このような踏み台で使われるのも Linux が現状は一番多いという気がいたします。

不正侵入の実態

・ポートスキャンの多くが、踏み台にされたサーバからきている

　残念ながら Linux サーバが多い

・最近は named へのアタックが多いようだ

　公開されているツールで、root 権限でアクセス可能になる

・情報収集とこまめなアップデートが必要

最近 named へのアタックが多いというのは、やはり Linux のサーバの古いバージョン、このあたりを特に具体的にねらうためのソースコード、コンパイルしたらそのままアタックできるというプログラムが出ているということです。

named などのセキュリティホールというのは大体がバッファオーバーフローなどでスタック領域を壊すというものです。長い文字列を引数で送り込んで、コンピュータ内部の戻りアドレスなどを書きかえてしまいます。その結果、本来の正しい動きとは違うところに戻らせて root 権限をつかむ、というものです。これは実際にインプリメントするのは手間が

かかりますが、このようなツールを使えば、実際の渡すべきパラメータがデータとして書かれており、そのままアタックできます。ですからバッファオーバーフローというような、アタックする側にも割と技術が必要なセキュリティホールも、ツールが公開されて簡単にアタックできるようになってしまっています。私の知り合いでもこの named で何カ所かやはりやられておりました。
この named アタックの場合は DNS サーバが、不調を起こして死んでしまい外にアクセスできなくなるということで、気がつくことが多いようです。しかし、入られたことには変わりないということになります。

DoS アタック

・標的サーバを動作不能にする
・手法はさまざま
　大量のメールパケットを送りつける
　不正なパケットを送りつける
　サーバプログラムや OS のセキュリティホールを衝く

それから最近のアタックで多いのが、DoS（Denial of Service）、サービス不能攻撃というもので、相手側のサーバを動作不能にさせたりします。例えば、メールに限らず大量のパケットを相手にどんどん送りつけるというものもあれば、サーバプログラムや OS のセキュリティホールを突いてハングアップさせたりということもあります。
それから、SYN フラグ攻撃と呼ばれるものがあります。TCP セッション確立の際にはコネクションを張る段階で、最初に syn フラグを送るのですが、これは SYN フラグだけを送りつけて、あとは知らん顔をしてしまうというもので、SYN フラグだけを何 10、何 100 と送りつけたりします。

SPAM とメール不正中継

・一方的に送りつけられてくるメール
　送付先が数万、数十万に及ぶことも
　第三者からのメール中継を受け付けるサーバ(オープンリレーサーバ)が狙われる
・不正中継対策は必須
　/etc/mail/ip_allow、/etc/mail/relay_allow (5.2)

これは政府まで動き出そうとしていますが、セキュリティ面の脅威以外に、不正メール、SPAM というものがあります。SPAM というのは、一方的に送りつけられてくる自分には意味のない有害なメールのことです。送付先が数千、数万、数十万というのはざらにあるようで、送る側からすると正規に契約しているプロバイダからは許可されません。そこで、第三者のメール中継を受け付けてくれるサーバ、オープンリレーサーバを世界中探し回って、そこを使っていくわけです。

自分の sendmail のサーバで不正中継の対策をしていなければ、ある日突然数十万通のメールを送るのに悪用されるという可能性があります。中継サイトに使われるとメールサーバの資源も食われますし、回線も非常に消費しますが、先ほどの踏み台と同様に信用の低下ということにつながり、場合によると世界中から抗議メールを受け取るようなことになります。従って、不正中継の対策が必須になってくるだろうと思います。
WIDE の sendmail などに入れかえる方は多いと思いますが、Red Hat の 5.2 のデフォルトの sendmail で説明します。
/etc/mail/というディレクトリがあり、不正中継に関係するファイルは ip_allow と relay_allow の二つで、メールの正当な送信元アドレスや社内のアドレスを書きます。

・/etc/mail/ip_allow メールの正当な送信元アドレスのリスト

```
127.0.0.1
192.168.0
```

この場合は、ローカルホスト、つまり自分自身、それから 192.168.0 というのは、ネットワークアドレスの意味になりますが、192.168.0 から送られてきたメールは受け取るということです。

・/etc/mail/relay_allow の例

```
mydomain.co.jp
```

外部から受け取るメールのドメイン名として、自分のドメイン名を書いておきますと、sendmail はそのドメイン宛のメールを受け取り、それ以外はメールの中継を拒否します。
雑誌でも２、３カ月前の"Software Design"などでメールの不正中継対策をかなりのページを割いて特集していました。こういうものを参考にして手を入れていくことも有効と思います。

8. 最近のビジネス状況

最近のビジネス状況について簡単に説明します。

プレインストールPC

まずハードウェアですが、プレインストール PC というものが、以前から秋葉原のプラットフォームなどから販売されており、現在も非常に好調に売れているようです。また、どのような製品形態になるかわからないのですが、日立は、いわゆる大手メーカとして初めて、Linux 対応という方向を出しています。その他のメーカでもいろいろ水面下では考えているようです。

ですので来年ぐらいになれば、プレインストール PC や、どこまでのサポートが得られるかは別にして Linux 対応というような PC が、非常に入手しやすくなっているのではないかと思います。しかし、私どものような立場で１台ずつサーバを組むときは、それで十分なのですが、数十台、数百台というロットで製品化、シリーズ化していくという場合にはやはり厳しい状況で、それは今後も続くだろうと思っています。これは最近のハード寿命が大変短くて２、３カ月ぐらいという点が問題で、特にネットワークカードなどは一番辛いところです。

周辺機器ベンダの対応

周辺機器ベンダでも例えばメルコや I・O データなどでは、Linux 対応に対して割と積極的になっています。その場合、メーカでドライバを出すなど、どういうところまでやるのかは、各社のそれぞれの思惑なのでわかりませんが、ドライバソースをオープンソースにするところも出てくるでしょうし、今後広がっていくことを期待しています。

ソフトウェア

・DBMS
・ビジュアル開発ツール
・グループウェア
・ミドルウェア
・日本語対応のクライアント用パッケージ
・オープンソース化の進展

ソフトウェアの状況ですが、データベースはご存じのように主要な RDB のベンダなどが Linux 対応を表明しまして、DB2 もダウンロードが始まっています。いわゆる商用のデータベースは、ほとんどのものが使えるようになっていく方向です。日本オラクルはまだ態度が決まっていませんが、流れからすると対応すると思います。

Linux はコンパイラなどの言語環境は非常に豊富ですが、ビジネスということで考えていきますとビジネスアプリケーション、また開発生産性ということでは、ビジュアル開発ツ

ール、例えば VB や Delphi などが必要と思います。名前は忘れましたが、いくつかのビジュアル開発ツールが移植されつつあります。それから、グループウェアは Notes が移植、対応方向の表明というレベルになってきています。
それからミドルウェア的なレベルでは、富士通が自社製のものをいくつか Linux に移植していくという予定のようです。
後は日本語対応のクライアント用パッケージということで、オムロンソフトの Wnn6 や dp-Note あたりが皮切りだったのですが、つい昨日のニュースでは一太郎アークというのが Java 対応になって、サポートプラットフォームに Linux がのっているという状況になってきています。クライアント用のパッケージというのは、もう少しかかるかなと思っていましたが、予想に反して案外早くなるのかもしれません。
まだもちろん主流まで行っていませんけれども、これら全体を通じて、いくつかのソースがオープンソースで出てくるという流れがやはり出始めています。例えば IBM が、sendmail に代わる MTA、sendmail よりもセキュリティ面で有利だという MTA をオープンソースで出す、ということが報道されていたと思います。ドライバレベルでも先ほどのような話が出ていますので、自社の中にノウハウを囲い込むよりもオープンソースの方がメリットがある、というものからオープンソース化が進んでいくと思います。

組み込み機器

・インターネットサーバ
・ファイルサーバ
・プリンタサーバ
・ファイアウォール
・ユニークな機器

組み込み機器ということですが、こちらも好調にいろいろ動いておりまして、インターネットサーバでしたら、デザイン面で一番おもしろいというのは Cobalt Cube です。
ファイルサーバ、プリントサーバ、ファイアウォールなどの専用機などもいくつかのところで現実に動き始めています。ユニークな機器としては、インターネット冷蔵庫やゲーム機器のようなところに、どんどん入っていくのではないか予想しております。例えば、プリクラのようなゲームマシン的なものです。

サポート

・書籍、雑誌
・ユーザ教育
・ヘルプデスク(Q&A)
・コンサルティング
・アプリケーション開発

ハード、ソフトなどの製品以外の面に関してはサポートということがあります。私どもの社名のサードウェアという由来でもあるんですが、ハード、ソフトに次ぐものという面につきましては、まず書籍、雑誌が非常に好調ですし、来年にもいくつかまた新しい専門誌的なものが出ると思います。日本ではまだわかりませんが、IDG などはアメリカでは予定しているようです。
また、ユーザ教育が本格的に立ち上がる傾向が見えてきています。もう既にいくつかの会社では Linux のインストールの教室などをやっておられて好調に動いているようです。多分近々アナウンスされると思いますが、セミナー専門の大手ではないですが、大手が割と大規模に始めるという予定を聞いています。
ヘルプデスクということに関しては、日本で Red Hat を出しているレーザファイブが、Q&A という会社と組んでコンシューマ向けのヘルプデスク､電話対応のサポートを始めるようです。私どもの方でもビジネスユーザマーケットなどの方面をねらったものをやっていきたいと考えております。
それに付随してさまざまなコンサルティングやアプリケーション開発も順調に立ち上がってきておりまして、現状においていろいろな会社で、人が足りない、技術者が足りないという状況になっているようです。

「ビジネスはチョイス」

・Windows-NT 対抗
・商用 UNIX 対抗

Linux がここまで急に出てきた背景には２つの側面、NT 対抗として見られている面と、それから商用 UNIX 対抗として見られている面があると思います。
まず、NT、Microsoft に対しての不満として、コスト面とか信頼性(よく落ちる)の問題があり、また Microsoft 自体が今叩かれているということもあり、そのあたりが一気に露出してきたという側面があると思います。商用 UNIX 対抗ということでは、やはりコスト面、また、安くても機能や性能面で十分というところから出ていると思います。
止まっては困るけれども、瞬断も許されないほどクリティカルなものではないという点で、インターネットサーバは今、非常にもてはやされる、一番注目される使い道だと思います。実際、機能面など実績を含めてインターネットサーバというのは Linux が商用 UNIX に十分対抗していける分野だろうと思います。

増える仕事

Linux のシェアは最終的に何%になるのかという疑問ですが、ユーザ登録も何もないものでシェアという数字自体がわからないのですが、NT、UNIX のマーケットの中で感覚的にいうと 10〜30 パーセントまでいく可能性はあるのではないか思います。現在は「0.何%」か、もっと下ぐらいかなと思いますが。

今後、ビジネスは急激に広がっていくのではないかと思います。そうすると、これまで経験のない SI ベンダも参入して、いろいろなところで Linux がわかる人が足りないという現象が起きると思いますが、既に私のところではそのような状況です。

オープンソース革命

そのような最近の流れをいくつか突き詰めていくと、今後オープンソース革命というのが起きるのではないかということが予想されます。この言葉は中村正三郎という方が Web ページでも使っておられます。背景には、ソフト、ハードが性能と比べて相対的に低価格化したこと、そしてオープンソースのソフトの実績が、品質が高い、安定している、基本的な機能は十分に備えているなどの点で実際に認められてきているということがあります。それが冒頭にも出て来ました公共財という見方とも相まってきて、やはり基盤的なソフトは、オープンソースの方がよいという見方が出てきています。

オラクルの副社長も最近インタビューで、オープンソースにしなくても、自社のデータベースパッケージの値段がただになっていくということが可能性としてあるだろうと言っています。当然オープンソースというのは一般的に、コスト面ではただで使えるということですので、そうなってきますとハードには一定の価格はつくがソフト、特に基盤的なものはただの方向に向かうということになります。

そうすると、本来、人が頭や体を使って働いている部分、インテグレーションやアプリケーション開発などの部分できちんとした対価を取っていくように変わっていくというのが、オープンソース革命の中身ではないかと思います。情報サービス自体、ハードやソフトでは売り上げが十分に取れなくなりますが、インテグレーションのニーズはますます複雑化し、変わっていくわけです。そうなるとアプリケーション開発、インテグレーション、教育、サポート、コンサルティングなどで付加価値をつけていける、そのような体力づくりが今後必要になってくるのではないかと考えています。

しかし商用のパッケージはいくつかの分野で残るものがあるだろうと思っています。実際にオープンソースのソフトを見てみますと、基本性能とか安定性その他、基本的なスペック部分は非常にいいものが多いですし、現実に Linux もそうです。しかし使い勝手ということになりますと、やはりノウハウやデザインなどまた別の問題があり、オープンソースですべていけるとは思っていません。ですからオープンソースと商用パッケージのすみ分け、組み合わせ、使い分けということになると思います。ただ、そういうことを通じて基盤的な部分はますますオープンな方向になっていくのではないかと思います。例えばデータフォーマットでも、DOC ファイルなどのスペックがオープンになれば、さまざまなものの出力フォーマットで DOC ファイルがあってもよいわけです。

当面の課題

・基幹業務への進出

Linux にとっての当面の課題、ビジネスに関わる人の課題ということですが、具体面はよく知りませんが、基幹業務への進出というのが現在いくつかのところで進んでいるようです。今後の進展のために必要なことは、初物に手を出してくれる勇気ある会社も必要ですが、開発生産性を高めるようなビジュアル開発ツールとか Case ツールというもの、それから高信頼性です。例えば本当のフルのクラスタなどを含めた信頼性の問題、さまざまなコンポーネントをニーズに応じて組み合わせていくという点でのミドルウェア。それから技術的ではないですけれども、経営トップを巻き込んだアプローチ、このようなものが、今後特に必要ではないかと最近考えております。

・オープンソースの維持

もうひとつ、オープンソース自体の開発体制やモラルを維持していくということも非常に重要な課題だと思っています。

私どもはオープンソースのソフトを使ってビジネスとして利益を追求していくわけですが、同時にオープンソースの利点を理解して開発体制やモラルを維持していくということに対しても十分な働きかけをしていきたい、この２つの調和を図っていきたいと考えています。

私自身の課題でもあるのですが、オープンソースに対して一種の貢献、寄与というものを考えなければいけません。どういう形になるのかといいますと、金銭的なものも一部入るでしょうが、それよりも何らかの形での推進というものをやらなければいけない、やりたいと考えています。

・サポート

サポートに関しても、とにかくこれから一気に立ち上がっていく分野だと思っていますが、規模も広げて質も上げていかなければなりません。

例えば Linux では、その気になればカーネルレベルからサポートできるわけです。カーネルレベルからのサポートというのは昔のメインフレームの SE 張りつきでのサポートと同じことになります。やるかやらないかは別にして、カーネルのレベルまで含めたカスタマイズというサポートも可能です。もちろんもっと表面的なサポートも可能です。

語弊があるかもしれませんが、現在はまだインストール本でもサーバの構築本ばかりで、価値は非常にあるのですが、もっとバラエティに富んださまざまな本やサポートなどが出てくる必要があると思います。ただ、それをやるためにはまだまだ人が足りません。また、人がいても、これも語弊があるかもしれませんが、マニアではちょっとだめだなというところもあります。マニアの方は自分が興味を持ったところはプロもかなわないぐらい深くなるという、よいものを持っています。しかしプロというのは、そこまでは部分部分で及

ばないにしても、全体的に自分の気が向く、向かないではなくて、しっかりとニーズに合わせて対応できるという違いがあると私は思っています。そういう意味でのプロがこれから必要だと思います。
かなり私の主観も交えて、オープンソース革命は進むだろうという前提で、最近の状況をかいつまんでご紹介いたしました。

9. Q&A

質問: 例えば ISP など、WWW サーバなどを提供するような会社ではクライアントに対して管理のためのアクセスを提供する必要があると思いますが、そのような場合に telnet はどうしているのでしょうか。
回答: TCP wrapper を使って、クライアント用に登録されたアドレスに限定して telnet を開くという運用が可能です。しかし、アドレスの詐称、スプーフィングという可能性もありますので、やはり SSH などでより安全なアクセスに切りかえていくのがよいと思います。特に最近では、ユーザが Windows クライアントであることが多いと思いますが、TeraTerm という端末エミュレータは SSH を使えるような実装ができていると聞いています。やはり SSH を使用するべきで、telnet 自体はインターネットに向けては公開しないというようにすべきと思います。

質問: 実は昨日、一昨日と BOF とチュートリアルに参加したのですが、そこでインターネットのサーバを何で立ち上げているかという司会者からの質問に対して、半分が Solaris、あと半分が FreeBSD、Linux は数%、１人か２人しかいないという現状でした。正直なところ非常に驚いてしまったのですが、御社ではなぜ Linux を選ばれたのかということを参考に聞かせて頂けますか？
回答: 多分一つは今までの流れで、特にここに参加されている方々の母体になるような団体の中では、インターネット向けのサーバは、実績として Solaris、FreeBSD が多いだろうと思います。
Linux と FreeBSD という話に関しても、海外でのユーザ数の推定では、いろいろなところの利用すべて含めて大体一桁違う、Linux のほうが一桁多いだろうと言われており、日本でも多分同じぐらいだろうと考えています。そのようなことを考えていきますと、ここの中ではやはり現時点で Solaris、FreeBSD などでやっておられる方が現状として多いのではないかと推測します。
次に、私どもがなぜ Linux を扱っているかということですが、一つは、はっきり言いましてたまたまで、私が最初に触れたオープンソースの UNIX が Linux であったということです。ただそれは出発点であって、これは FreeBSD でも同じだと思いますが、現在までこのようにビジネスを続けてきたのは、Linux がオープンソースでフリーな実装であり、その

中でセキュリティ対策やさまざまな情報が自分自身で入手してチェックできるという点があります。つまり私にとっては、自信を持ってやっていける唯一の OS であるということです。また結果としてビジネス的に見ましても、コスト面やいろいろな点でユーザからも評価して頂けているということで続けてきていますし、今後もそういう両面で続けていくだろうと考えています。

質問: TCP/IP のセキュリティ対策については、TCP wrapper の説明がありました。私どもの方で、LAN の障害などが起きた場合にデータを回収できないということがあり、シリアルポートに直接端末といいますか相手をつないでデータを回収する必要があります。また接続相手は、PC などではなく、TTY しか話せないもので、今のダイヤルアップルータを使うことも難しいという問題があります。その観点でシリアルポートを守るセキュリティ対策のものがあれば、ご紹介頂きたいのですが。
回答: シリアルポートのセキュリティですか?
質問: その後ろに電話がつくと思って頂きたいんですけれども。
回答: いわゆる電話回線とモデムを間に挟んでというものですね。こちらは私も正直言ってそれほど詳しくないのですが、今一番問題が生じているのは電話番号そのものが盗まれているということだと思います。
例えば会社の代表電話番号がありますと、その近くの番号にやみくもに電話をかけて、それでモデムらしきトーンが返ってくるかどうか調べるということが現実にあります。それがコンピュータのモデムの音ということになれば、本気でやる場合には屑籠あさりまでしてパスワードのヒントを探すようです。いわゆるソーシャルエンジニアリングというらしいですが。ですから、まずは電話番号を秘密にするということが必要です。
接続が PPP でなく TTY ということになると後は、例えばログインだと通常のパスワードや認証のあたりをしっかりするしかありません。もしも PPP ならば少なくとも PAP、それから CHAP を使っていくことになると思います。そのあたりをきちんとロギングして把握することがもう一個重要だろうと思います。そういう意味では商用のものになると思いますが、大規模にやる場合には、Radius の認証サーバなどを組み合わせることになるのではないでしょうか。NTT に電話番号を新たにもらうときには、会社の代表や他の部署で使っている電話番号とはできれば局番も分けて、全く脈絡のつかない番号にするということが必要と思います。

質問: 要点は、セキュリティというものはビジネスになりますか、ということですが、御社の場合はサーバ構築が主な商売だと思います。その場合、セキュリティというものは、客からするとあって当たり前というもので、破られた場合の責任保証という問題が生じると思います。また逆に、これだけの費用を頂ければこれだけのセキュリティを保証します、というビジネスの形もあると思います。特にサーバを表に出す場合は、セキュリティも一

つのサービスポイントになるかと思いますが、このような点については、どうされているのでしょうか。
回答: 細かいところはノウハウ的な話もありますが、考え方としては今おっしゃいましたのと全く同じです。私どもはサーバを売っておりますが、基本的に売り切りはしていません。実際は運用管理のために契約して頂くという形にしています。あくまでそこでうたっているのは基本的には運用管理です。運用管理の細かい専門的な知識を要する部分をアウトソーシングして頂くという考え方で、セキュリティもその中の一部として位置付けています。セキュリティも一部ということで、当然ながらセキュリティ絡みのモジュールのアップデートなども全部契約の範囲で対応するということになります。
ですからセキュリティをある程度強調したい方は致しますが、絶対に落ちない、侵入されない、というセキュリティを提供するというのは怖くていえないですし、セキュリティ以外の要素もいろいろ運用管理には伴いますので、そういう程度の位置付けでやっています。

質問: 運用管理のサービスの中でのアウトソーシングということですので、リモートによるログの監視などもやられると思いますし、先ほどの説明にも SSH などが出ました。私どもでも今は NT をベースにして、リモートのメンテナンスもサービス品目にしていますが、先ほどの説明にもありましたように Linux などの場合は非常に怖い場面が多いと思います。そのあたりのポリシーはいかがでしょうか。
回答: 私どもは一昨年ぐらい前に、SSH の商用版の開発元の Data Fellows と SSH の運用について e-mail で交渉しました。私どもに関しては、当時のバージョンのライセンス規定でいうところのコンサルタントなどが付加料金を取らずにメンテナンスのために顧客との間で SSH の経路を作るのはフリーでよいという、延長の解釈に対する OK の回答を引き出しております。ですので私どもは現在その解釈上で SSH を使っています。もちろん顧客からの要望があればダイヤルアップで専用回線を用意するという対応もありますが、通常の一般的な運用管理は SSH を実際に使っております。
このあたりについてフリーかどうかということは、もちろん価格などにも影響しますのでこだわらざるを得ないとは思いますが、確か F-Secure のサーバ製品を使ってやっておられる会社もあったと思います。

質問: 落ちにくい Linux のサーバを運用するという上で、冗長的な構成、さまざまな機材、リソースなどが必要になってくると思います。私どもではずっと Solaris をメインで使っていますが、DiskSuite など割と安い値段で手に入るソフトウェア RAID の製品やその他のさまざまな冗長を構成できるような製品があります。Linux に関してはあまり深くは知らないのですが、堅牢なサーバを組む上でのこのような情報を教えて頂けますか。例えばディスクのソフトウェア RAID、ミラーリングができるようなもの、その他の冗長構成ができるようなものです。UPS については APC に問い合わせたところ Linux はサポートして

いないということでした。
回答: セキュリティを離れてこのあたりになると私の別の得意分野でもあるんですが、まずカーネルレベルでソフトウェア RAID はサポートしております。ですから、ハードに比べるとスピードとか、ホットスワップ、オートリカバリなどでの制約はあるにしても、日常的なミラーリングは Linux カーネルと周辺のコマンドを設定すれば使えます。RAID3 や 5 になると実際パフォーマンス的には、あまりいい影響は出ないとは思いますが、RAID は可能です。その他の冗長ということになりますと、例えばいわゆるクラスタのフェールオーバーなどというのは、多分これからの課題だと思います。
しかし例えば、WWW サーバを複数台置いて１台がダウンしたら、他のものが自動的に引き継ぐというのはサーバの OS は関係ありません。そのようなものを自動的に切りかえるハードウェア製品、何かハブみたいなレベルでフェールオーバーするようなものも出ているようですので、それらと組み合わせることになると思います。
それから UPS については、APC からはまだ会社としてのサポート姿勢は出ていませんが、APC のパワーシュートというユーティリティや、その他いくつかのベンダでは Linux で実際に動くというものがあります。私どもでも数社の UPS については実績を持っております。また来週ぐらいにお会いするんですが、ある会社では明確にドライバをオープンソースにして、Linux 対応で UPS を出したいという動きをしています。
本当に安定稼働を長時間保証するということでは、現実的な価格で考えますと、まずソフトウェア RAID を使うということです。万一落ちた場合でも、リカバリーがスムーズに進むようになります。また UPS は電源異常時のオートシャットダウンまで当然対応しています。そういうものを組み合わせていくわけですが、私どものサーバで大体長いものでしたら、数百日連続稼働しているものはいくつもあります。

質問: 職場で２年ほど前から Solaris と FreeBSD と Linux でインターネットサーバを運用しております。組み上げたホワイトボックスの AT 機を使っているのですが、やはり 24 時間稼働で何年も運用するということになると、パーツの信頼性がだいぶ低くて、半年に一回ぐらいは部品やディスク交換、冷却の強化など、何らかの対策が必要になっています。
そうなるとメーカ製のいわゆるサーバ製品、NT サーバを載せるために設計されたようなものを導入するということになるんですが、その費用を考えますと Ultra5 などを買って Solaris にしてしまった方が、ということになってしまいます。
今後 Linux やフリーの PC UNIX プラットフォームが伸びていく上で、その辺のバランス、自社組み上げまたはメーカから購入という点で、コストが必ずしも安くならないという問題点があると思いますが?
回答: ハード絡みというのは、ちょっと辛いというか、確かに今の PC は電源ファンなどはひょっとしたら１年ぐらいでへたってしまいます。予算を本当に切りつめる場合には PC UNIX のメリットはあると思いますが、安定性重視ということを考えますと、コスト面で

は、わずかには安くできるとは思いますが、確かに大差なくなる可能性があると思います。どこまでの程度で現実的に満足するかによりますが、かなり高いレベルを求めるならばおっしゃる通りで、OS の構築コストはほんの一部ですから差がなくなると思います。

以上