Linuxサーバ構築とセキュリティ

荒木靖宏

インターネットイニシアティブ

日本Linux協会(JLA)

構築の前に考えておくこと

運用ポリシ

完全な破壊にいたらない場合の対策

- □UCEメール対策
- □DOS対策
- □対策窓口

対攻撃(対事故)

- □物理的攻撃からの防護
- □電気的攻撃からの防護
- □ユーザからの防護
- □ネットワーク攻撃からの防護

運用ポリシ

全ては運用ポリシ次第

何を守るのか

サーバ設置の目的を明確にすること

- □できるだけ明確に
- □危険なサービスは代替手段を考える
- □今必要なサービスだけを行う

セキュリティと利便性のトレードオフを意識する

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の留意点を挙げる

そもそもなんでLinux?

管理者が慣れているOSだから

多くのアーキテクチャで動く

- Intel PC、Sparc、PowerPC..
- **ミニサーバ専用機**

大規模なマシンから小規模なものまで

- PC**ベースのサーバ製品多数**
- **古い**note**パソコンでも動く**

システムに対する適正なコストで運用できる

Linuxと一言でいうけれど

Linuxの定義

- 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
  - 様々なコマンド
  - アプリケーション

ディストリビューション

沢山あるけど何を選ぶ?

一般的な見方

- **パッケージが豊富**
- **運用を含めた使い易さ**

サーバを構築するなら

- **バグ、セキュリティホールの**update**速度**
- **商用プログラムを使うならその対応**
- Trinux**のような最小限のディストリビューションを選ぶ手も**

インストール時の指針

自分がよくわかる(好きな?)ディストリビューションを使う

自分が把握しているパッケージのみ

必要最小限

最新のパッケージ

- □インストール時は最小インストール
- □後から使用しないようにするよりも確実
- □メモリ、プロセステーブルの節約にもつながる

インストールの必要のないサービス

X

- Xそのものに問題があった例も報告されている

コンパイラ

- できるなら他の同構成のマシンでコンパイルする
- 侵入者の手段となることも

かな漢字変換

- canna,wnn,..などなど
- 特に日本で使うことを意識してつくられたディストリビューションでは注意する。

インストールの必要のないサービス(2)

グラフィックライブラリ

- svgalib,fb,gtk,qt...
- **ハードウェアを直接叩くプログラムや特権の問題がある場合も**

NFS,RPC,NIS

- portmapper(/sbin/portmap)**の停止**

そのほか

- inetd**から起動するサービスを確認**
  - finger
  - talkd

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

- static linkになってない場合も
- トラブルの元になることも

sashの利用

- static link
- 最低限の機能

ファイルシステムの問題

いまのところ、historyが記録できるファイルシステムがない

- XFS (from SGI) **待ち?**
- ext3,afs..**いろいろ開発されてはいる**

NFS**問題**

- **ロックが形式違いなどで動作しないことも**
- knfsd**の登場で解決しつつある**

情報収集をふだんから

Vine Linux http://vine.flatout.org/errata.html

Plamo Linux http://www.linet.gr.jp/~kojima/Plamo/

redhat http://www.redhat.com/corp/support/errata/

Debian GNU/Linux http://www.debian.org/security/

Securityfocus http://www.securityfocus.org/

カーネルでできること

linuxのカーネルは自分が利用するために自由に変更ができる。
これを生かさない手はない。

インストール時のカーネル

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カーネル

- 40種にものぼるSCSIドライバ
- NFSやSMB、NCPなど不必要なプロトコル
- CDROM,ISDNなどなど

古いバージョンのカーネルであることが多い

- セキュリティやパフォーマンスでの問題

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

必要のない機能を削る

- a.out**バイナリ、マウス、サウンド**

ネットワークプロトコル

- NFS、SMB、NCP、appletalk**等を削る**
- CONFIG_FIREWALL**などが必要なら加える**

カーネルによるリソース制限

/proc**以下にあるパラメータで設定する**

殆どの場合は必要ないがチューニングも

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

http://www.openwall.org/

- **メモリの使用を厳しく制限するパッチなど**

TCP syn flood 攻撃への対応

TCPのhand shakeを悪用した攻撃

対策

- **カーネルコンパイル時に** CONFIG_SYN_COOKIES=y
- /proc/sys/net/ipv4/tcp_syncookiesを1にする

カーネル関連情報

http://www.kernelnotes.org/

http://kt.linuxcare.com/ (kernel traffic)

http://www.linuxhardware.net/

http://www.linux.org.uk/diary/ (Alan Cox’s Diary)

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、そしてその対象を限定すること。

安全のためには?..

利用できるサービスを制限する

安全な代替サービスを検討する

サービスのユーザ権限/file

サービス構成を理解した運用

安全な実装の利用

サービスを絞る

提供するサービスを少なくする

- □使わないサーバは削る
  - ○identは削るとサービスを受けれなくなることも
  - ○/etc/rc3.d 以下をリネームして起動しないように
- □サーバ停止時に他のサービスの影響を受けにくくなる
- □他に影響を及ぼすサービスもある
  - ○ひとつづつ削っていく

login

ログイン、シェルの提供はよく考える

- □リモートからほとんどの資源が利用できる
- □いったん握られたら、「乗っ取り」「踏台」なんでもできる

最低限の認証方法

- □ユーザ名とパスワードのみで利用可能

パスワードファイル(/etc/passwd)

ユーザー情報が詰まっている基本のファイル

- □シャドウ化でパスワードは/etc/shadowに
- □DES暗号がほとんどだがMD5,SHA1等が使われることも
  - ○PAMが採用されているディストリビューション等

あらゆるプログラムが利用する

- □あらゆるユーザが読める

パスワードなどを含む盗聴

多くのパケット盗聴プログラムあり

- **基幹にクラッカがしかけると被害大**
- promisc modeでNICが動作しているときは要注意

対策

- **スイッチ**
- **ネットワークの分割**
- OTP
- **暗号化**

パスワード認証をするサービス

とにかくユーザの特定と書きこみを行うものはほとんど

telnet、ftp

POP、IMAP

- **メールをクライアントがサーバから読みだす最も一般的な方法**
- **だれでも使うだけに危険**

telnet、r系サービス

リモート端末からloginできるようにする

loginと同時にシェル起動しあとは何でもできる

ユーザ名、パスワード、内容一切暗号化されない

.rhostsの設定でパスワード認証を行わないのはさらに危険

OTP(ワンタイムパスワード)

平文パスワードのかわりに使う

パスワードの再利用攻撃はできない

サーバ実装としてOPIE等がある

- telnet,ftpのOTP化
- /etc/opiekeysにMD5化されたパスワード

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

対応クラインアントは増えてきたが、まだまだ全部というわけでもない

SSH(Secure Shell)

通信路を暗号化する

- **最初の利用以後はホスト鍵に変化があると警告**
- **ユーザ認証手段に**RSA**鍵を選択可**

ライセンスに注意

- f-secure**社が商用ライセンスを持つ**
- OpenSSH**の登場でライセンスもクリアに**
  - http://www.openssh.com/

OTPとSSHの長所

OTP:

- **あらゆる環境で利用可能**
- OTP**がわかっていればいい**
- windows、mac、PDA**などの**OTP**計算機**
- **紙にメモっておく手も**

SSH:

- **移行しやすい**
- r**系サービスのほとんどを置き変え可能**
- **より強固な認証手段を選択できる**
- **他プロトコルをカプセルできる**

OTPとSSHの短所

OTP:

- **通信内容が保護されないことを念頭に**

SSH:

- **通信路暗号化のコストがかかる**
  - **最近の高速マシンではあまり関係ない?**

セキュリティ強化ツールの活用

アクセス制御

- **発信元、送信先のアドレス、サービスの種類による**
- **利用者、時間帯による**
- tcp-wrapper(libwrap)**がよく利用される**
- ipchains**も利用できる**

tcp-wrapper

多くは /usr/sbin/tcpd **でインストール**

制御ファイル

- /etc/hosts.allow
- /etc/hosts.deny
- libwrap**を用いるようなプログラムも同じファイルを使用する**
- man 5 hosts_access**を参照**

safe_finger**によるアクセス元追及は今や無意味**

- **攻撃相手は隠蔽をしている**

/etc/hosts.allowの例

全てのアクセス許可

- ALL: ALL: ALLOW

特定のネットワークからのアクセスのみ全て許可

- 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

(ルータ等での)パケットフィルタの利用

サービスに必要なもの以外は塞ぐべき

アクセス制御に有効

- **その限界を知ることが必要**

通常外部からのアクセスを禁止すべきサービス

- tftp,finger,portmapper,NBT,snmp,exec,login,shell,X

必要なものを塞がないように注意

ipchains

どんなもの?

- **フィルタリングルールのリスト**
- input,output,forward**の評価をする**
- **デフォルトではルール無し**
- source,dist.,protocol,interface**毎にふるまいを指定できる**
- IP**マスカレードで使われることが多い**

ipchainsの適用範囲

filtering router

- □**単なるフィルタリング**

NAPT

アクセス制御

- □tcp_wrapper**のように細かい制御はできない**

NAPT+cipe**と組みあわせて**VPN

- □http://sites.inka.de/~bigred/devel/cipe.html

ipchainsの動作

カーネルでIPを再構成したあとで評価

単純にヘッダを見て(いじって)、流す

パケット内容についての評価はできない

- □アプリケーションレイヤの透過プログラムを別に用意

サービスのユーザ権限/file

限定したユーザアカウントで運用

- inetd.conf**での指定,プログラム内での指定**

- nobody
  - **多くの**www**サーバ**
  - **読み出しだけで済むサービス**

例:

```
skkserv  stream tcp nowait nobody /usr/sbin/tcpd /usr/sbin/dbskkd-cdb
```

chroot環境での運用

危険なサーバを限定して公開するために有効

特定のdirectoryをルートディレクトリとして動作させる

- ライブラリや必要な設定ファイルを忘れずに作る必要がある
- wu-ftpd等標準で利用するプログラムもある

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をざっととらえてみる

内部ユーザからの防護

攻撃者がrootになる壁にもなる

- **攻撃者はまず一般ユーザ権を盗る**
- **その後rootになるのがほとんど**

ファイルパーミッション

- world writable**になってないか**?
- **起動しないように安全にふってるプログラムもある**

- umask**を厳しく**
- immutable **ビットの利用**

内部ユーザからの防護(2)

ファイルシステム

- nodev,nosuid,noexec**などのオプションを活用**

setuid**されたプログラム**

- **ほんとうに、必要な**setuid**なのか**?

内部ユーザからの防護(3)

資源占有を防ぐ

- shellでのリソース制限
- 普段の教育

サーバで使うならユーザを限定することをまず考える

マシンの前に座られたらおわり

- /etc/inittabをいじってctl-alt-delでのリブートを無効に

sudo

制限されたコマンド群をrootの権限で動作させる

- 必要上root権限を与える必要があるとき
- 例:
  - web作成者にapacheのrestartをさせる

あらゆるコマンドが記録される

注意点:他のプログラムを起動できる可能性

- シェルエスケープができる場合など

PAM

認証方法を交換可能にするモジュール

できることの例:

- **パスワードの暗号を**DES**以外に**
- **ユーザリソースの細かな制限**
- **ユーザの**login**制御**

設定は /etc/pam.d/ **以下に**

chfn chsh kbdrate linuxconf login other passwd ssh su sudo wu-ftpd

ネットワークからの防護

攻撃対象

- □ カーネルそのもの
  - ○ アプリ層より下の攻撃
- □ サービス
  - ○ 無駄なサービスはあげない
- □ DoS攻撃
  - ○ 完全に防ぐ方法がない。/* 実に悩ましい問題 */
  - ○ 普段のリソース監視

物理的攻撃からの防護

Linuxは物理的攻撃に強いか?

- 単なるPCが多いが、サーバ機もでてきている
- 設置場所は運用次第
- まあPCだから部品のストックには困らないかも
- 別にLinuxだからといって特別なことはない

電気的攻撃からの防護

耐サージ(家では特に重要かも)

- **商用電源**
- **イーサネット**
- **電話線**

Linux**で使える**UPS

- APC(apcupsd **などがフリーである**)
- **三菱**(FREQUPS **に附属してプログラムがある**)

破壊にいたらない場合の対策

UCEメール

- 踏台などになって加担するとメールが送れなくなるおそれ
- LinuxはMail管理が甘いのが一般の認識

DOS攻撃

- 十分なサービスが提供できなくなる
- networkが埋まると大変
- 攻撃のためのscanは日常的なもの

生活が困難になる状況があるので、セキュリティ上の問題と同じように扱うべき

コトがおこったら?

対策窓口

- **いわゆる「お客様相談センター」的なものをおいて、姿勢を示す。**
- RFC2142**にあるサービスを設置する。**
  - abuse,postmaster,hostmaster,...

JPCERT/CC**、契約**ISP**、当事者等へ報告**

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

日常の管理項目

全体のリソース

- □Disk、メモリ、プロセス

ユーザ管理

- □ユーザの更新、パスワード

データ管理

- □バックアップ

そのほか

- □マシンの物理的な環境

全体のリソースの監視

自分で提供しているサービスを使ってみる

- 意外とわすれがち
- つかってみて気がつくこともある

ps、top、df などをしてみる

管理者宛てに送られるメールには最低限の情報が

プロセス間の関係

pstreeの利用

```
init-+-apmd
     |-atd
     |-bash---startx---xinit-+-XF86_SVGA
     |                       `-sawmill-+-kterm---bash-+-bash---emacs
     |                                 |              |-gqmpeg---mpg123---mpg123
     |                                 |              `-mgp2ps---pstree
     |                                 `-xv
     |-cardmgr
     |-cron
     |-dhcpcd
     |-5*[getty]
     |-inetd---dbskkd-cdb
     |-kflushd
     |-klogd
     |-kpiod
     |-kswapd
     |-kupdate
     |-lockd
     |-master-+-pickup
     |        `-qmgr
     |-named
```

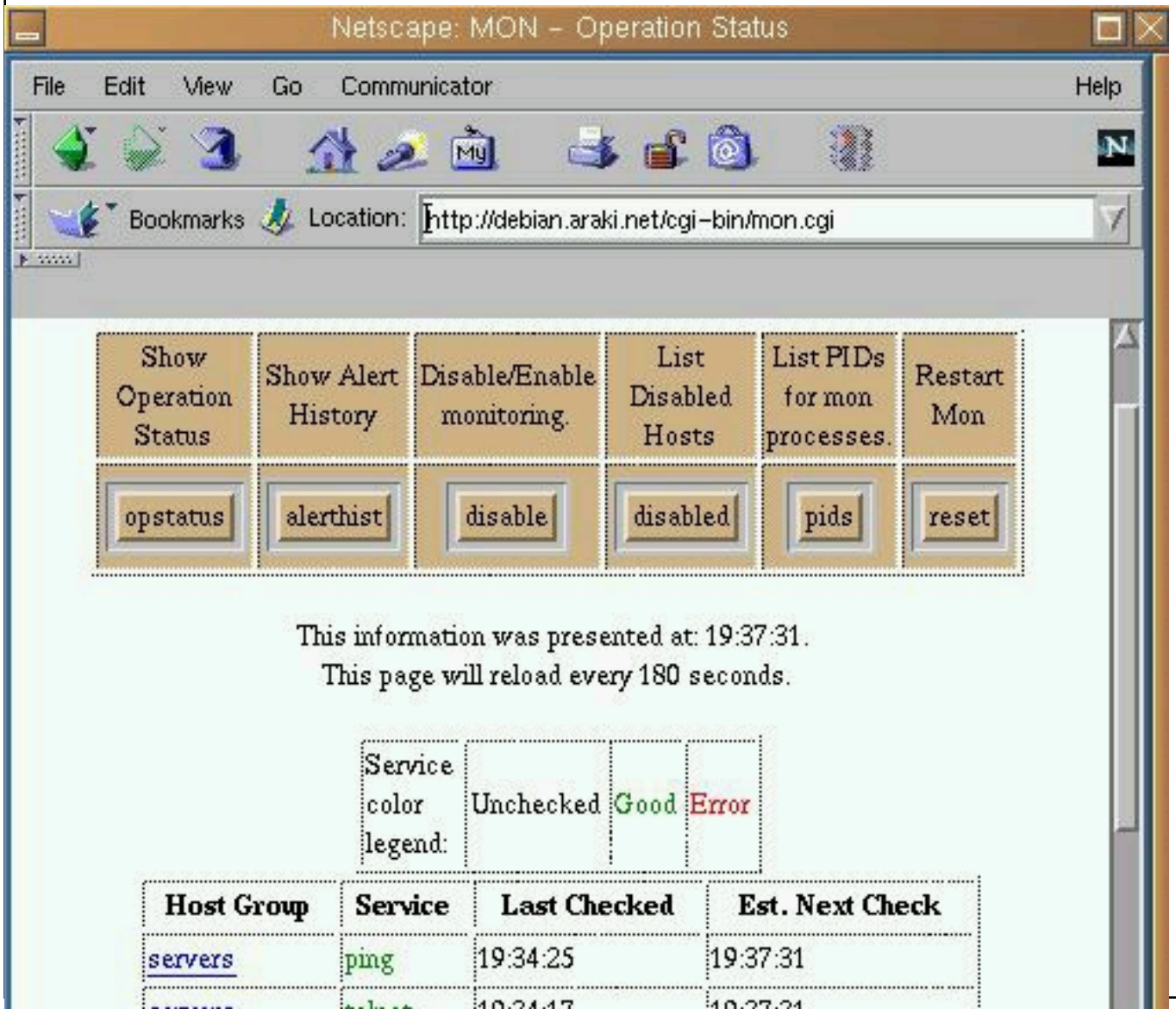
なんでもまとめて監視するツールもある

mon

- 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...
- **ホスト情報**
  - **ディスクスペース**
- SNMP**と連携**
  - uptime,**プロセス**...

mon(2)

wwwブラウザからコントロールできる



監視/記録の必要性

ホスト、ネットワークの常態を知る

- □己を知れば。。
- □攻撃なのかどうなのかの判断材料

ホスト拡張の予測材料

- □サービスを行うことが第一

攻撃から守る

ホストの監視

何はともあれ syslog

/etc/syslog.conf**の書きかた確認しましょう**
パフォーマンスが欲しければcyclog**等**

注意点:

- syslog**自体が攻撃されたら大変**
- **とりこぼしの可能性も**

アプリケーション独自のlog

慣れる

- **そのサービスの特徴がでているので利用価値は高い**

webや**ftpは独自のlogを出す**

- **専用のlog処理ツールも多数ある**

iplogger

- TCP, UDP, ICMP **接続そのものの監視**

異常が発見されるとき

普段と違う挙動の発見

- □異常に同じ行為をくりかえしおこなわれる
- □サービス外portへのアクセス

クラッカの典型的な手法の発見

- □クラッキングツール
- □ポートスキャン

ログの監視を楽にする

syslog-summary

- **同じメッセージがあればカウントして表示**
- **く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**
- /proc**や**syslog**のリアルタイムモニタ**
- **フィルタ設定で注意するべきものの抜き出し可能**
- **音も鳴らせます**

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

- **メール、ビープ音、端末出力等**

定義ファイルの例

- /authentication failure/ echo,mail

典型的な使用法

- swatch -c /etc/swatch/auth -t /var/log/auth.log

特定アプリ用log解析ツール

MTA用:

- sendmail用 -- syslog_stats.pl
- postfix用 -- pflogsumm.pl
- qmail用 -- qmailanalog

www用:

- analog
- webalizer

ftpd用:

- xferstats

ファイルの改竄、変更監視

設定ファイルや実行ファイルの変更頻度は低いことを利用

不正侵入の発見に

- 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

tripwireが有名

- ファイルのハッシュ値DB作成
- チェックして変更を検知
- 商用にも

networkの監視

promiscモードでの監視

- なんでも見える
- 見てはいけないもの/こども
- 監視用のマシンはサービスマシンと別にする手も

かんたんなパケットモニタ

- iptraf
- tcpdump

networkの監視(2)

NFR(Network Fright Recorder)

- **商用の侵入者検出プログラム**
- **非商用利用はフリー(ソースも)**
- **パケット収集と解析**
- JAVA**対応ブラウザでの参照**

バックアップ

ある時点に戻せることが重要

上書きできないものに

- □**今なら**CD-R
- □**大容量の**MO(**爪をスライド**)

HDD**等へのバックアップ**

- □**バックアップ処理後は外して安全なところへ**

何をバックアップするか

インストール直後

- **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

- **ホスト全体**

まとめ

安全はあたりまえのことの積み重ねから

- □まめな情報入手
- □自分でサービスを使う/叩く

複雑なシステムで危険を覆うより、危険の除去を考える

- □ソースも入手できる
- □リソースを食うのははばからしい

機械化できるところはやる

- □ミスの減少
- □人は人ができるところを

まとめ(2)

最後は人間

- □いろんな環境をつくってみる
- □とにかく慣れ
- □攻撃してくるのは人間

おしまい