フリーソフトウェアによるネットワーク監視

- 3rd Edition -

矢萩 茂樹
イー・アクセス株式会社
2001/12/5

本チュートリアルの進行

- 第1部 ネットワーク管理の基礎知識
- 第2部 フリーソフトウェアによるネットワーク監視
- 第3部 ネットワーク管理に関するTIPS
- 質疑応答

第1部：ネットワーク管理の基礎知識

eAccess

ネットワークとは

- ネットワークとは、
  様々な中継装置の複合接続により、通信機能を持つ機器を遠隔接続し、多対多のコニュニケーションを実現するシステム
  - 様々な機器による複雑系
    - ネットワークは生き物。状況は刻々と変化する。状況把握はなかなか困難
  - 多岐にわたる構成機器・関連機器：
    - PC、ルーター、スイッチ、WDM伝送装置、ATM交換機、多重化装置、サーバー、DSU、メディアコンバーター、ケーブル、電源、冷却装置、…

eAccess

監視業務の必要性1

- ミッションクリティカル
  - E-mail/DNS/WEBは既に企業活動上、必須の通信手段となった
  - E-mail/DNS/WEBが止まる＝業務が止まる！
- 多岐にわたる構成機器と複雑になる障害要因
  - 全機器の稼動状況を把握するのは困難な作業
  - とはいえ、障害は必ず起こる
    - 泣き言はいってられない
- 止まらないネットワークは実現困難
  - お金と手間をかければ冗長構成をとれるが…、バックアップされただけ
    - 結局復旧作業は必要
  - 止めないようにする努力と止まったときに復旧する努力
    - こける前にささえるのが一番被害が少ない。けど…
    - こけてしまったら、なんとかはやく直さないと…

監視業務の必要性2

- ネットワークをこけさせないためには
  - (障害の)予兆をつかんで、予防保全
- こけたものをできるだけ早く立て直すには
  - 障害発生を速やかに検知し、
  - 原因をできるだけ早く確定する
- これらの実現には、定常的な監視とサポートする監視システムの構築が必要

監視処理とは

- 監視ポイントを設定し、
  ↓
- 閾値を設けて、
  ↓
- それを超えているか判断し、
  ↓
- 通知すること

- このフローの繰り返しとなる
- 通知が障害と判断された後、障害対処に分岐

監視のワークフロー

1. 定常運用
   1. 稼動状況確認
   2. 障害検知
2. 非定常運用
   1. システムメンテナンス
   2. ラインアップ
3. 障害対応
   1. 障害検知
   2. 状況把握
   3. 原因確定
   4. 復旧処理

- 3.1障害検知から 3.3 原因確定区間をシステム化により情報整理を行い、
- オペレータが復旧処理にできるだけ早くかかれるようにしたい

速やかな原因確定を実現するワークフローとは

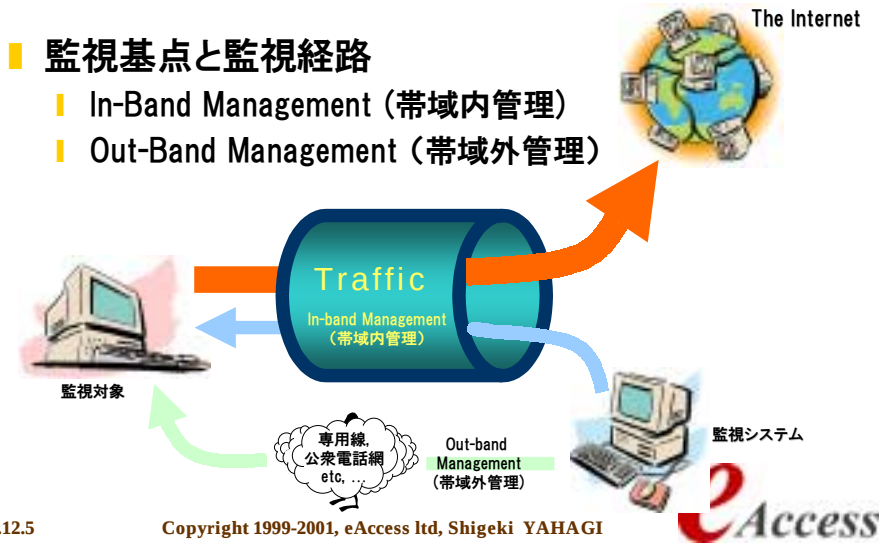
- 的確な参照情報が盛り込まれたアラーム通知が重要
- 必要なのは必要十分情報と障害サマリー画面へのURLリンク
  - 障害発生時刻
  - 障害発生個所・機器
  - 障害状況
  - 障害サマリーページへのURL情報
    - 障害情報のみがまとめられたサマリー画面
- この情報をアラームメッセージに乗せることで、速やかな障害サマリーページへのアクセスを実現

↓

- ハイパーリンクによるワークフローの確立

監視ポイントの設定

- 監視基点と監視経路
  - In-Band Management（帯域内管理）
  - Out-Band Management（帯域外管理）



通知について考える

- 監視通知の方法
  - アラームメッセージ
  - グラフィカル（画面、アイコンの点滅）
  - アラーム音
  - メール
  - 携帯電話にリモートメール
    - 実時間の保証など確実性に少々欠けるが、リモート通知手段はこれしかない

監視における切り口 - 現状と経過

- 現状監視 - 今を見る：Polling
  - 現在のシステム状態を監視する
- 現状検知 - 今を検知する：Event Trap
  - システムからの自律アラームをとらえる
- 経過監視 - これまでを見る：Traffic Analysis
  - トラフィックの推移を監視。トレンドを把握する
- これらは密接にからんだ独立事象
  - どれがぬけても片手落ち

監視する手段 - Polling

- ICMP Polling
  - ICMP echo(ping)による監視。必要最低限どんなノードでもサポートしているために、最低限の監視に使用可能
- TCP/UDP Port Polling
  - 各サービスポートを直接監視する方法。実際に稼動しているかを直接判断できるので、サーバープロセスの監視には有効
- SNMP Polling
  - SNMP(Simple Network Management Protocol)を用いて、ネットワーク機器のより詳細なデータを取得・監視を行う

ポーリング – ICMP/ping

- ICMP echo/echo replyによりIP的動作確認をする
  - 付帯情報: RTT : Round Trip Time
    Packet Loss : パケット到達率
- ホストの動作確認するもっとも簡単な手段
- RTT/Packet lossは、時系列に整理すると回線品質を測る上で重要な情報として利用できる

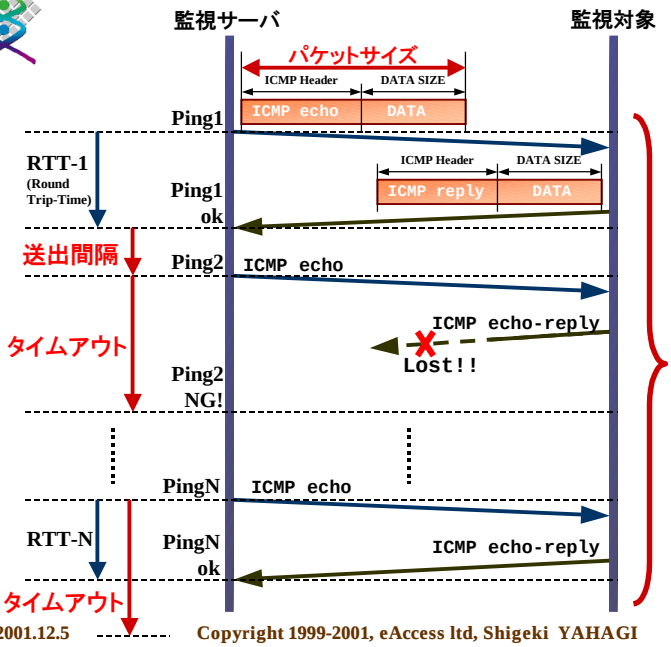
```
$ multiping -t -c 10 www.apple.com www.bose.com
PING www.apple.com (17.254.0.91) (17.254.0.91): 56 data bytes
PING bose.com (146.115.60.42) (146.115.60.42): 56 data bytes
64 bytes from 17.254.0.91: icmp_seq=0 ttl=224 time=144 ms
64 bytes from 146.115.60.42: icmp_seq=0 ttl=237 time=207 ms
      ～省略～
64 bytes from 17.254.0.91: icmp_seq=9 ttl=224 time=151 ms
64 bytes from 146.115.60.42: icmp_seq=9 ttl=237 time=208 ms

-=-=- PING statistics -=-=-
                                        Number of Packets          Round Trip Time
Remote Site                      Sent    Rcvd    Rptd   Lost     Min   Avg   Max
------------------------------  ------  ------  ------  ----    ----  ----  ----
www.apple.com (17.254.0.91)        10      10       0    0%     143   144   151
bose.com (146.115.60.42)           10      10       0    0%     206   206   208
------------------------------  ------  ------  ------  ----    ----  ----  ----
TOTALS                             20      20       0    0%     143   175   20
$
```

ポーリング – ICMP/ping

- pingにまつわるパラメータ
  - パケットサイズ
  - 送出回数
  - 送出間隔
  - タイムアウト
  - （送出内容）
- 監視を正確に効率的に行うためには送出間隔・タイムアウト をきめ細かく制御する必要があり
  - 計測対象が増加すると、pingのパラメータチューニングは必須
    - 計測が終わらない…
  - SLA的な確認を行うためには、短時間に適正なタイムアウトを設定したping試験を行いたい

Ping に関するパラメータ



ICMP/ping - ツール1

- fping
  - http://www.stanford.edu/~schemers/docs/fping/fping.html
  - icmp echoについて特化。複数のホストを一気にチェックすることが、可能
  - 送出間隔・タイムアウトが調整可能など、かなりの項目をきめ細くチューニングが可能
- Multiping (SNIPS (aka NOCOL))
  - http://www.netplex-tech.com/software/snips/
  - ネットワーク監視ツールであるSNIPS(旧名: NOCOL)のサポートツールとして付属。multipingだけでも十分利用価値がある
  - 複数のホストにまとめて試験を行い、測定結果の表形式サマリーとして出力可能

ICMP/ping - ツール2

- sing
  - sing - Send ICMP Nasty Garbage packets to network hosts
  - http://sourceforge.net/projects/sing
  - ICMPパケットジェネレータ
  - ICMPパケットを自在にカスタマイズして、送出することに特化したツール
  - ICMP redirect, unreachableなどのパケットを自在に作成、送出できる

監視する手段 – TCP/UDP Port Polling

- 各サービスポートを直接監視する方法。実際に稼動しているかを直接判断できるので、サーバープロセスの監視には有効
- 各サービス用のクライアントを使ってチェックを行うものが多い

```
$ telnet www.eaccess.net 80    ← http(80)でアクセスしてみる
Trying 211.14.194.242...
Connected to www.eaccess.net.
Escape character is '^]'.      ← session確立
get /                          ← GET request送信
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>501 Method Not Implemented</TITLE>
</HEAD><BODY>
<H1>Method Not Implemented</H1>
get to /index.html not supported.<P>
Invalid method in request get /<P>
<HR>
<ADDRESS>Apache/1.3.12 Server at www.eaccess.net Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
$    ← 結果が返ってきているのでOK! Serviceの起動確認
```

- DNS：nslookup
- radius：radping(DIT Radius), radpwtst(Merit Radius)
- http: lynx, w3m
- telnetでたたいてチェック

監視する手段 – SNMP Polling

- 標準プロトコルベースでの管理方法
- サーバー・クライアント型プロトコル
  - サーバー：SNMPマネージャー
  - クライアント：ネットワーク機器(エージェント)
- ベンダーに依存せず、様々な機器において各種トラフィック・運用状況の監視が可能
- ルーターやインテリジェントスイッチから詳細情報を得るにはもっとも一般的
- アプリケーションサーバーでは個別にSNMP daemonを追加しなければいけない場合が多い
  - 商用製品が多い。例：HP OpenView

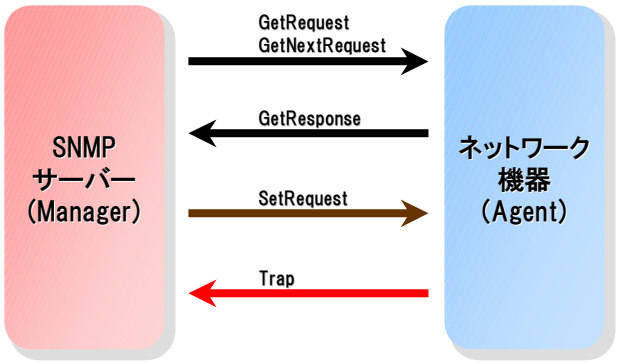
SNMP : Simple Network Management Protocol

- SNMP: Simple Network Management Protocol
  - UDP : polling port 161, trap port 162
- マルチベンダーを実現するための2つのフレームワーク
  - 情報取得のための簡潔なプロトコル
  - 取得情報を標準化するMIB(Message Information Base)
- 情報伝達の2つのモード
  - ポーリング
    - マネージャからエージェントに情報を要求する
  - トラップ
    - エージェントからマネージャーに対してイベントを転送する

SNMP Messages

- GetRequest：manager→agent
  - マネージャが更新情報を要求する
- GetNextRequest：manager→agent
  - マネージャがテーブルの次のエントリーを要求する
- GetResponse：manager←agent
  - エージェントがマネージャからの要求に応答する
- SetRequest：manager→agent
  - マネージャが管理対象機器装置のデータを修正する
- Trap：manager←agent
  - エージェントがマネージャにイベントを通知する

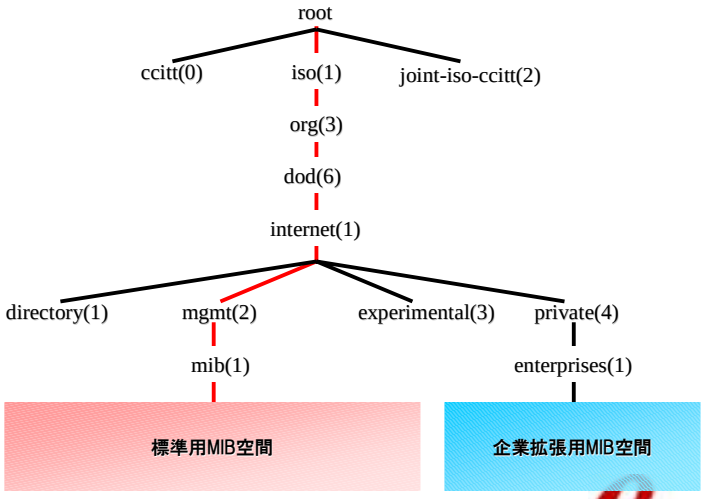
SNMP Message Handling



MIB: Message Information Base

- RFC-1213 インターネット標準 MIBv2
- 階層的な命名体系で管理オブジェクトを定義
- オブジェクト識別子(OID: Object ID)とMIB Symbol
- 標準勧告部分(MIBv2)と企業特有部分(Enterprise MIB)に分かれる

MIB Tree



MIB OID

- 表記法:

```
iso(1).org(3).dod(6).internet(1).mgmnt(2).mib(1).
    1: system       システムグループ
    2: interfaces   インタフェースグループ
    3: at           アドレス変換グループ
    4: ip           IPグループ
    5: icmp         ICMPグループ
    6: tcp          TCPグループ
    7: udp          UDPグループ
   11: snmp         SNMPグループ
```

監視する手段 - Event Trap

- Local Event Trap
  - システム状態監視(CPU,disk,process,memory,...)
  - Process個別処理状況監視
  - Log file監視
  - Packet Monitoring
  - ファイル監査
- Remote Event Trap
  - syslog によるメッセージ伝達
  - SNMP trap
  - NMS Probe Clientからのリモート通知

監視する手段 – Event Trap: Hacking対策

- Packet Monitoring
  - ネットワークインタフェースに流れるパケットを監視
  - 入力パケットのパターンマッチングを行い、不正アクセスを検出する
- ファイル監査
  - 全てファイルのステータス情報を保存し、定期的に変更されているかをチェックする
    - Time Stamp, File Size, Check Sum, ...

監視する手段 – Event Trap: SNMP

- SNMP trap
  - Pollingを待たずにSNMPエージェントからマネージャに情報を送る
  - トラップを利用してエージェントがマネージャに異常イベントの発生を知らせる
  - ポーリングの制御権は、マネージャが保持
  - SNMPのトラップ情報はUDPポート162に送られる
  - 詳細は第2部にてのべる

第一部まとめ

- IP通信はミッションクリティカルなインフラとなった
  - 通信を良好に維持するためには、定常的な監視が必要
  - こけさせないためには→予兆をつかむ
  - こけたら→障害発生を速やかに検知し、原因を早急に特定し、回復させる
- 監視における切り口
  - 現状監視
  - 現状検知
  - 経過監視
- これらは密接にからんだ独立事象
  - 3つの監視をもれなく行うことで、通信を安定維持させる

第2部：フリーソフトウェアによるネットワーク監視

eAccess

何故フリーソフトを使用するか

- 小規模なLANでも、手軽にネットワーク監視
- 必要は発明の母
  - 必要だから作った。みんなの必要が集約
    - 実際に管理している人のノウハウが反映
- アレンジ可能
  - さらに自分の必要を加えることができる
    - ソースに手を加えることが可能
  - 商用NMSですぐに対応できないところに適応する

eAccess

フリーソフト導入の問題

- 保証無し。全て自分の責任で
- マニュアルが完備しているとは限らない
  - 英語がにがてなんだけど…
  - 最後はコードを読めというのかい
- サポートがあるかどうか不明確
  - バグも結構あります。→ だれに相談する？？
  - けど、作者が開発を止めているものもある
- 高機能なものはインストール自体が大変
  - 他のフリーソフトへの依存度が高い
- 結局は不明確というのが最大障壁か

フリーソフト導入の問題 – ならば

- 不明確の整理。せめての一言をうめてみる
  - マニュアルがそろっていないよ
    - 不自由しない程度に完備しているツールを選べばよろしい
  - サポートはどこ？
    - 活発な公式MLがあるツール。作者のやる気も大事！
      - 活発ということはユーザが多いということ。迅速なバグ対処も期待できる
  - インストールが大変？！
    - Ports/Packageで楽々インストール
      - メジャーなツールならPorts/Packageになっているはず
    - インストールマニュアルがしっかり書いてあるツールを選ぶ
    - autoconfのおかげで最近ほんとに楽になってます

フリーソフトで作る監視システム

- 全てを一つで満足することはできない
  - 満足するものに作り上げるための努力は無視できない
  - なんでも(そこそこに)できるは何もできないの法則
- なら、適材適所の組み合わせで簡単に作る！

- 監視システムを構成する3つのアイテム
  - 状態監視
  - 状態検知
  - トラフィック監視
- システムへの統合はWEBで

監視における切り口 - 現状と経過 (再び)

- 現状監視 - 今を見る：Polling
  - 現在のシステム状態を監視する
- 現状検知 - 今を検知する：Event Trap
  - システムからの自律アラームをとらえる
- 経過監視 - これまでを見る：Traffic Analysis
  - トラフィックの推移を監視。トレンドを把握する

- これらは密接にからんだ独立事象
  - どれがぬけても片手落ち

監視システムへの要件1

- 集中監視・一斉通知
  - 人は分散できない・できるだけしたくない
  - 監視は1カ所で。通知はそこから一斉に
- 監視画面は各自の手元で
  - …でも、自分の机で自分のPCでみたい

監視システムへの要件2

- 出先でも状態確認
  - リモートで対応するために遠隔で情報取得したい
  - 他人のPCでも状況確認できないと…
    - だれでも持っているソフト → Web Browser
    - 状態確認に専用クライアントが必要なシステムは、いざというときには使い物にならない
      - X-Window, tcl/tk, Perl/tkも同様
      - JavaはVM相性問題が少々あり

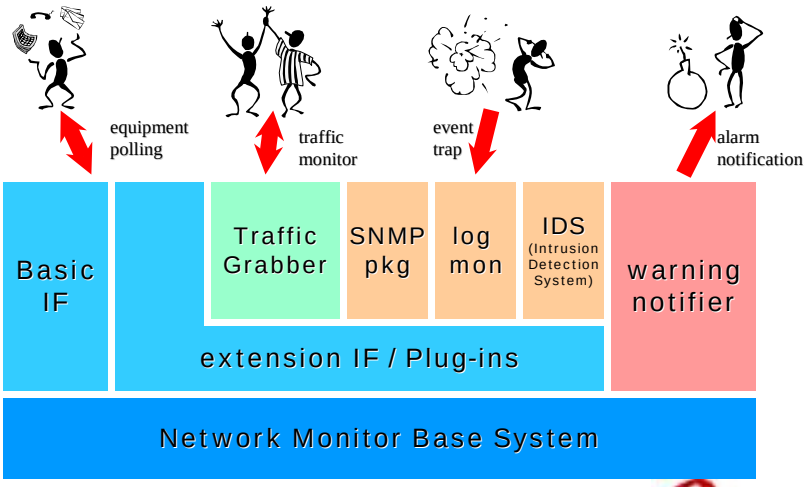
監視システムへの要件3 - 結論

- WEB画面でリモート監視・リモート確認
- E-mailで通知(まずは携帯へPager Call ？！)
- WEB画面でトラフィック監視

監視の切り口と使えるツール

- 今を見る → Polling Base状態監視
  - Big Brother, NetSaint, SNIPS(NOCOL), SPONG, mon,...
- 今を検知する → Trap Base状態検知
  - log監視ツール(SWATCH, LogSurfer,...)
  - Snmptrapd(NetSNMP),...
  - IDS (Snort, Ntop, ...)
  - ファイル監査(Tripwire, ...)
- これまでを見る → トラフィック監視
  - MRTG+RRDTools, nPULSE, seafelt, Shepherd, RRDTools+Frontends(Remstat,Cricket,ORCA,NRG), ...

監視システムのモデル



3-7, December 2001 Pacifico Yokohama Internet Week 2001

フリーソフトウェアによる ネットワーク監視

Big Brother Network Monitor

2001.12.5
Copyright 1999-2001, eAccess ltd, Shigeki YAHAGI

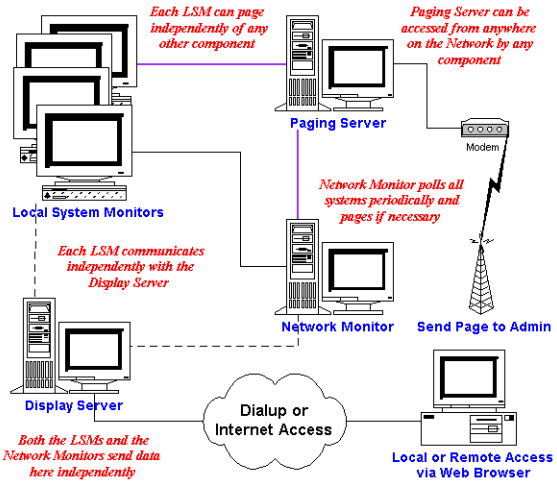
Polling base 状態監視ツール

- ネットワーク監視ツールもいろいろと増えてきましたが、今年もBig Brotherを中心に据えて解説します。
  - その他の有用な監視ツール
    - NetSaint
    - DEMARC
    - SPONG
    - nPulse
    - SNIPS (aka NOCOL)
    - Mon

状態監視ツール - Big Brother

- http://bb4.com/
- WEB Baseの監視システム
  - オープンソースであるが、フリーソフトではない
  - 通常使用においては費用は発生しない
- 監視・表示・通知機能をモジュール分割しており、それぞれを別サーバに分散することで、大規模ネットワークまで適用可能
- ICMP/TCPポーリングによる監視を行う
  - 監視可能サービス：
    - `ping,smtp,http,https,pop3,dns,ftp,telnet,ssh,imap,nntp, …`
  - サーバー個別監視：`CPU,disk,processes,logs`
- 各種Unix/NT/NetWare/Macintoshの監視用プローブがあり、複合OS統合監視が可能

Big Brotherサーバー構成



状態監視ツール - Big Brother 続き

- 監視対象のグループ化機能
- 監視画面の階層化機能(2段階)
- 柔軟なアラーム通知機能
  - E-mailによりアラームを通知する。
  - ホスト単位にシステムの停止時間を設定。自動で監視対象から除外可能
  - ホスト単位で障害通知先を変更可能
  - アラームの検出されている機器のみサマリーした画面を標準で生成。→ 便利！
  - アラームメッセージに障害情報ページのURLが引用されており、迅速に障害情報に到達できる！

状態監視ツール - Big Brother 続き

- 障害履歴機能
- システム稼動状況レポート作成機能
- 拡張インタフェースが公開されており、非常に多彩な拡張監視モジュールが存在する(後述)
  - オープンソースの利点を生かし、BB基本ソフトをそのまま置換する機能拡張版ソフトも存在する
  - 拡張監視モジュール: DBMS, ファイルサーバ, プリンタサーバ, ...
  - 他ソフトとの連係: MRTG, RRDTool, snort, tripwire, ...
  - BBTray：Big Brother監視ツール on Windows
- マニュアルがかなり整っている。:-)
  - 各モジュールの構成にまで踏み込んだ解説付き
- 適用範囲：ネットワーク監視、IDS Frontend、気象情報監視、株価監視(?!),...

BB - 監視画面 (TOP)

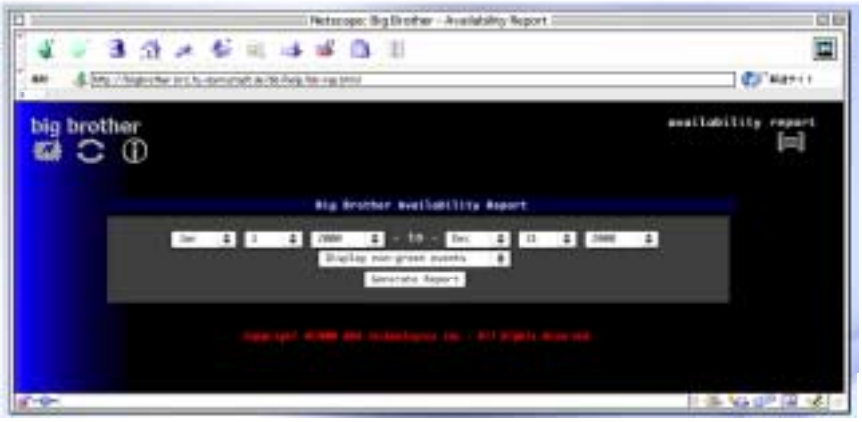
BB - 監視画面 (sub)

BB - アラートサマリ

BB - イベント情報画面

BB - ヒストリ画面

BB - 稼動レポート作成



BB - 稼動リポート(top)

BB - 稼動リポート(sub)

BB - 稼動リポート(text出力)

```
Availability Report
Oct 17 2000 - Nov 1 2000
                              dev01.lab.eccess.net - cpu

                              Availability:   95.52%

                        Red     Yellow  Green   Purple  Clear   Blue
                        4.48%   2.11%   93.41%  0.00%   0.00%   0.00%

                              Event logs for the given period

     Event Start              Event End                Status  Seconds Cause

     Mon Oct 30 20:13:42 2000 Mon Oct 30 20:18:42 2000 yellow  300     up: 12 days, 2 users, 48 procs, load=703
     Mon Oct 30 05:38:43 2000 Mon Oct 30 05:43:43 2000 yellow  300     up: 11 days, 2 users, 48 procs, load=708
     Sun Oct 29 15:03:44 2000 Sun Oct 29 15:08:44 2000 yellow  300     up: 10 days, 3 users, 51 procs, load=725
     Sun Oct 29 10:53:45 2000 Sun Oct 29 10:58:45 2000 yellow  300     up: 10 days, 3 users, 51 procs, load=702
     Sun Oct 29 08:03:49 2000 Sun Oct 29 08:08:46 2000 yellow  297     up: 10 days, 3 users, 51 procs, load=708
     Sun Oct 29 00:28:46 2000 Sun Oct 29 00:33:46 2000 yellow  300     up: 10 days, 3 users, 51 procs, load=773
     Sat Oct 28 20:23:47 2000 Sat Oct 28 20:28:47 2000 yellow  300     up: 10 days, 3 users, 51 procs, load=710
     Sat Oct 28 09:53:47 2000 Sat Oct 28 09:58:48 2000 yellow  301     up: 9 days, 4 users, 56 procs, load=730
     Sat Oct 28 06:18:48 2000 Sat Oct 28 06:23:47 2000 yellow  299     up: 9 days, 4 users, 66 procs, load=715
     Sat Oct 28 03:33:48 2000 Sat Oct 28 03:43:47 2000 yellow  599     up: 9 days, 4 users, 90 procs, load=806
     Sat Oct 28 02:03:46 2000 Sat Oct 28 02:08:46 2000 yellow  300     up: 9 days, 4 users, 75 procs, load=713
     Wed Oct 18 12:41:25 2000 Wed Oct 18 21:21:35 2000 yellow  31210   up: 4 days, 2 users, 43 procs, load=415
     Tue Oct 17 16:06:23 2000 Wed Oct 18 12:41:25 2000 red     74102   up: 3 days, 1 users, 64 procs, load=464

                              Time Critical/Offline: 20 hours 35 mins 2 secs

                              Time Non-Critical: 9 hours 40 mins 6 secs
```

BB - 設定ファイル

- etc/bb-hosts:
  - 監視対象定義ファイル
- etc/bb-warnrule.cfg:
  - 障害通知定義ファイル
- etc/bbdef.sh:
  - システム監視定義ファイル
- etc/notes directory:
  - 注釈機能
- その他
  - etc/security.INFO

BB - etc/bb-hosts - 1

- 監視対象の定義ファイル
- 記述方法は/etc/hosts の拡張版に近い
- 監視対象の記述:
  - `<IP Address> <Host Name> [ # <Service> {<Service>}]`
    `IP Address: 監視対象のIP Address`
    `Host Name: 監視対象のホスト名`
    `Service: サーバー機能及び監視サービス。`

Big Brotherの設定例 - bb-hosts

```
$ cat bb-hosts
#
# THE BIG BROTHER HOSTS FILE
#
192.168.0.10 kansi1.foo.co.jp # BBPAGER BBNET BBDISPLAY http://kansi1/

group-compress <H3><I>foo.co.jp Servers</I></H3>
192.168.0.2  ns1.foo.co.jp  # dns ssh !telnet
192.168.0.3  mail.foo.co.jp # dns smtp pop3 ssh !telnet
192.168.0.5  www.foo.co.jp  # telnet ssh ftp http://www.foo.co.jp/

# router interface entry
page Router-IF “Router Interface”
group-compress <H3><I>Router1 Interfaces</I></H3>
192.168.0.1  gw1.foo.co.jp
192.168.0.50 gw2.foo.co.jp
group-compress <H3><I>Router2 Interfaces</I></H3>
192.168.1.2  tok-yok-ma30.wan.foo.co.jp
192.168.1.6  tok-osa-dr15.wan.foo.co.jp
$
```


BB - etc/bb-hosts - 2

- Serviceには以下のものを記述可能。
  - サーバー機能： BBNET,BBPAGER,BBDISPLAY
    - BBDISPLAY: ネットワーク監視画面サーバが動いていることを指示
    - BBPAGER: ネットワーク警報通知サーバが動いていることを指示
    - BBNET: ネットワーク監視サーバが動いていることを指示
  - ping監視はデフォルトで行われる。以下のアレンジも可能
    - noping: ping監視を行わない。監視対象外の表示はする
    - noconn: ping監視を行わない。表示自体も消す
    - dialup: ping監視結果:NGにて、アラームをあげない
  - 監視サービス： smtp,http,pop3,dns,ftp,telnet,ssh,imap
    - httpはURL指定する。例: http://www.foo.co.jp/top.shtml
    - 以下のアレンジが可能。
      - !telnet : telnet portが開いている際に警告を行う。
      - ~telnet : 試験は通常通りに行い、逆の結果を返す。
        - 例: 試験OK:赤、試験NG:緑

BB – etc/bb-hosts - 3

- 特殊設定項目: `dialup modem-bank`
  - DHCP/ダイアルアップのアドレスプールの使用状況を確認する
    - 例: `dialup modem-bank 192.168.0.92 16`
      - 計測時間がかかるので、あまり多くのプール監視はむかないと思う
- 画面修飾関係の設定
  - 表示グループ指定: group, group-compress
    - group(-compress) <group name>
    - この指定以下の計測対象をひとつの表示サブグループとして固めて表示する
      - group：すべての計測項目を表示する
      - group-compress：サブグループ内にて計測される項目のみ表示する
    - <group name>にはhtmlタグが使用可能
  - サブページ指定: page
    - page <page name> <page title>
    - この項目以下の計測対象をサブページにまとめる
    - 画面上は<page name>の項目にまとめて表示される。状態表示アイコンからサブページにリンクがはられる
    - <page title>にはhtmlタグが使用可能

BB - etc/bbwarnrule.cfg

- 警告通知に対するルールを記述する
- 記述方法:
  - `hosts;exhosts;services;exservices;day;time;recipients`
    - `hosts:` 一致するホスト(“*”はワイルカード)
    - `exhosts:` 除外するホスト
    - `services:` 一致するサービス(“*”はワイルドカード)
    - `exservices:` 除外するサービス
    - `day:` 0-6 (日曜日-土曜日)
    - `time:` 0000-2359
    - `recipients:` メールアドレス
    - `hosts,services`についてはワイルドカード指定可能

Big Brotherの設定例 – bbwarnrule.cfg

```
$ cat bbwarnrules.cfg
# bbwarnrules.cfg
ns1.* mail.*;;*;;*;*;server-admin@foo.co.jp
##  ns1.*, mail1.*については24H/7Dの監視を行い、障害時はserver-adminに通知する
www.*;;http;;*;*;web-admin@foo.co.jp yahagi
##  www.*についてはhttpのみ24H/7Dの監視を行い、
##  障害時はweb-adminとyahagiに通知する
storage.*;;*;conn;0-6;0000-0259 0500-2359;storage-admin@foo.co.jp
##  storage.*はping以外の監視を行い、障害時はstorage-adminに通知する
##  ただし、AM3:00-AM4:59までの間は監視対象外とする
intra.*;;*;;1-5;0800-2000;intra-admin@foo.co.jp
##  intra.*は月曜日から金曜日のAM8:00-PM8:00まで全てのサービス監視を行い、
##  障害時はintra-adminに通知する
*;;*;;*;*;admin@foo.co.jp
##  上記以外のホストの障害検知についてはadmin@foo.co.jpに通知する。
unmatched-*;;*;;*;*;bb@localhost
##  bb-hosts定義外のイベント(unmatched-*)検知についてはbb@localhostに通知する
# end of bbwarnrules.cfg
$
```


etc/bbdef.sh - 1

- Big Brotherシステム定義ファイル
- 稼動に必要な環境変数の定義を設定。監視閾値・挙動指定をし、外部拡張監視(plug-in)の登録もこのファイルに行う
- ディスク容量テスト設定:DFWARN, DFPANIC
  - ディスク容量テストの閾値を%レベルで表記する
    - DFWARN - warning設定値(default:90%)
    - DFPANIC - panic設定値(default:95%)
  - サーバー全体に関する設定であり、パティションごとに閾値を設定・管理したい場合にはetc/bb-dftabファイルに詳細設定を行う
- CPU load averageテスト設定:CPUWARN, CPUPANIC
  - load averageを元にシステムプロセス稼動状況監視のための設定
  - 設定値 = load average(uptimeから)の値 * 100
    - CPUWARN - warning設定値(default:150)
    - CPUPANIC - panic設定値(default:300)

etc/bbdef.sh - 2

- プロセス監視設定:PROCS, PAGEPROCS
  - 起動確認したいプロセスを定義する。後述
- メッセージ監視設定:MSGS, PAGEMSGS, IGNMSGS
  - システムログでエラーメッセージを監視したい場合に利用する
    - MSGS - warning対象キーワード
    - PAGEMSMS - panic対象キーワード
    - IGNMSGS - 識別対象外キーワード
  - それぞれの変数には':'をデリミタとするとことで、複数のキーワードを設定可能
- 警報レベル設定: PAGELEVELS
  - 警報を行うイベントレベルを設定する。デフォルトは"red purple"
    - Red = critical level
    - Purple = target no response
- 外部機能拡張登録: BBMKBBEXT, BBMKBB2EXT, BBEXT
  - 外部機能拡張(plug-in)の登録を行う。詳細は後述

設定例 - bbdef.sh

```
$cat bbdef.sh
#!/bin/sh
# bbdef.sh
【省略】
# LOCAL CLIENT MONITORING CONFIGURATION FOR bb-local.sh
#   WARNING AND PANIC LEVELS FOR LOCAL SYSTEM INFOMRAION
#   YOU CAN SET VALUES ON A SPECIFIC FILESYSTEM BY USING
#       THE etc/bb-dftab FILE
DFWARN=85                               # (YELLOW) DISK % TO WARN
DFPANIC=95                              # (RED) DISK % TO PANIC
export DFWARN DFPANIC
# CPU LEVELS ARE THE 5 MINUTE LOAD AVERAGE x 100
CPUWARN=3000                            # (YELLOW) WARN AT LOAD AVG OF 30 (default:1.5)
CPUPANIC=6000                           # (RED) PANIC AT LOAD AVG OF 60 (default:3)
export CPUPANIC CPUWARN
# PROCESS MONITORING
# THESE VALUES ARE OVERRIDDEN BY THE etc/bb-proctab FILE
PROCS="bbrun snmptrapd httpd !inetd"    # (YELLOW) WARN IF NOT RUNNING
PAGEPROC="cron radiusd sshd syslogd"    # (RED) PAGE IF NOT RUNNING
export PROCS PAGEPROC
# MESSAGE FILE MONITORING (/var/adm/messages or similar)
CHKMSGLEN="TRUE"                        # MAKE SURE MSG FILE IS NON-ZERO LEN
MSGS="NOTICE WARNING"                   # (YELLOW) MESSAGES TO WATCH FORP
AGEMSG="NOTICE"                        # (RED) PAGE IF WE SEE THIS MESSAGE
IGNMSGS=""              # List of messages to ignore if string(s) matches line
【省略 - 続く】
```

設定例 - bbdef.sh 続き

```
【省略 - 続き】
# Default colors to send notification messages on
PAGELEVELS="red purple"                 # Default red purple
export PAGELEVELS
# Specify scripts to execute while running mkbb.sh/mkbb2.sh
# Echo from them will be displayed on the generated web page
BBMKBBEXT="bbradius.sh"
BBMKBB2EXT="eventlog.sh"
export BBMKBBEXT BBMKBB2EXT
【省略】
# EXECUTE LOCAL SCRIPTS FROM HERE...
# SCRIPTS SHOULD LIVE IN $BBHOME/ext DIRECTORY
# BBEXT CONTAINS THE FILENAMES TO EXECUTE
# SEPERATE THE SCRIPTS WITH A SPACE:  BBEXT="ext1.sh ext2.sh"
BBEXT="larrd/larrd.pl larrd/bf-larrd.sh"
export BBEXT
【省略】
$
```


etc/bbdef.sh – プロセス監視定義

- プロセス監視設定:PROCS, PAGEPROCS
  - 起動確認したいプロセスを定義する
    - PROCS - warning対象プロセス
    - PAGEPROCS - panic対象プロセス
- 非起動確認についてもサポートしており、その際にはプロセス名の前に"！"を付加設定する
  - セキュリティー上あがっているとまずいプロセスの監視につかえる
    - ex: !inetd, !sendmail, …
  - 設定例

```
# PROCESS MONITORING
# THESE VALUES ARE OVERRIDDEN BY THE etc/bb-proctab FILE
PROCS="bbrun snmptrapd httpd !inetd"    # (YELLOW) WARN IF NOT RUNNING
PAGEPROC="cron radiusd sshd syslogd"    # (RED) PAGE IF NOT RUNNING
export PROCS PAGEPROC
```

procs監視の話

BB - extensions

- 拡張インタフェースが公開されており、非常に多彩な拡張監視モジュールが存在する
  - オープンソースの利点を生かし、BB基本ソフトをそのまま置換する機能拡張版ソフトも存在する
    - http://www.deadcat.net/
  - Enhancement script to BB
    - モジュールごと拡張版への置換
  - External plug-in script for BB
    - 外部拡張スクリプトによる機能追加

BB – extensions & plug-ins

- 実現されるもの
  - さらなるアプリケーションの監視：
    - radius, ntp, ldap, smb, mqueue, …
    - RDBS (Oracle, Informix, Sybase, PostgreSQL, MySQL, …)
    - 他システム監視：RAS, UPS, RAID, Printer, …
  - 他ソフトとの連係：例えばMRTG、RRDTools
  - モジュール毎入れ替えによる高速化
  - BBTray：Big Brother監視ツール on Windows

BB - Extension Archive

BB enhancement - Japanese Help

BB extensions - Japanese Help

- マニュアルの日本語化されたもの
  - 訳者不明？？
- ちょっと古くてver 1.6ぐらいの内容
- ftp://ftp.deadcat.net/pub/BB/japanese-help.tar.gz
- www/help以下をこれに入れ替えるだけですべてのマニュアルが日本語化される
  - 硬い日本語で情報が少し古いのですが、かなり助かる

Big Brother - extensions

拡張ヒストリー

- ftp://ftp.deadcat.net/pub/BB/bb-hist-2.6.tar.gz
- /cgi-bin/bb-hist.shの置換プログラム
- イベントヒストリ解析を拡張し、日間・週間・月間・年間のイベント状況を棒グラフにて表示する
  - MRTG的イベント解析
  - 長期トレンドにてシステムの稼動状況をみることができ、障害間隔などの状況も把握しやすいことから、かなり重宝する
- bb-hist.plとして提供されており、これを/cgi-binのbb-hist.shと置換することで、追加を行う

BBTray - Big Brother監視サポートツール

- Big Brother Display Serverを常時監視するサポートツール
- ftp://ftp.deadcat.net/pub/BB/BBtray-0.8.3.zip
- Windows9x/NT/2000/XPで動作
  - BBを監視し、状態が変化するとPopup Windowにて通知
    - 派手な警報音付き！
  - Windowをクリックすることで、障害サマリー画面に直接とべるので、即時に現状把握可能
    - BBサーバーとIP通信ができれば、どこでも現状が分かる
      - 客先で鳴ると非常に恥ずかしい☺
  - 類似品にtkBB(Tk-Perl版)あり

BBTray - 続き

Green Window
- this is normal status

Yellow Window
- this is warning status.

Red Window
- this is critical status!!

フリーソフトウェアによるネットワーク監視

Net-SNMP snmptrapd

eAccess

Net-SNMP Package

- http://net-snmp.sourceforge.net/
- さまざまなUnixプラットフォームで稼動するSNMP Package
- 以下のコマンドを提供
  - snmpd, snmptrapd, snmpbulkwalk, snmpget, snmpset, snmptest, snmpusm, snmpcheck, snmpgetnext, snmpstatus, snmptranslate, snmpwalk, snmpdelta, snmpnetstat, snmptable, snmptrap

eAccess

NET-SNMP snmptrapd

- SNMP trap eventを監視するdaemon
- trap eventごとに処理を規定することが可能
- Trap受信後、以下の処理を行う
  - 外部コマンドがアクションとして規定されている際には、アクションである外部コマンドの標準入力に受信したTrap eventを渡し、コマンドを起動する
- Trap受信によりアラートなどの通知を行うことが可能
- Snmptrapd.confの記述
  - `traphandle <OID> <action> <parameters….>`
  - `traphandle default <action> <parameters….>`

snmptrapd.confの例

```
# SNMP Trap : Cold Start
traphandle .1.3.6.1.6.3.1.1.5.1 /usr/bin/mail -s "coldStart Trap"
admin@foo.co.jp
# SNMP Trap : Warm Start
traphandle .1.3.6.1.6.3.1.1.5.2 /usr/bin/mail -s "warmStart Trap"
admin@foo.co.jp
# SNMP Trap : Link Down
traphandle .1.3.6.1.6.3.1.1.5.3 /usr/bin/mail -s "linkDown Trap"
admin@foo.co.jp
# SNMP Trap : Link Up
traphandle .1.3.6.1.6.3.1.1.5.4 /usr/bin/mail -s "linkUp Trap"
admin@foo.co.jp
# SNMP Trap : Authentication Failure
traphandle .1.3.6.1.6.3.1.1.5.5 /usr/bin/mail -s "authFail Trap"
admin@foo.co.jp
# SNMP Trap : Other
traphandle   default  /usr/bin/mail -s "Other Traps" yahagi@foo.co.jp
```

snmptrapd - CISCOルータでのsnmp関連config例

- CISCOルータでのSNMPv2設定例
  - access-list 30 permit 192.168.100.1
  - snmp-server contact admin@foo.co.jp
  - snmp-server location YOKOHAMA-IW2001
  - snmp-server community himitsu RO 30
  - snmp-server enable traps config
  - snmp-server host 192.168.100.1 NAISHO tty config envmon snmp

snmptrapd - 通知結果

- From: log-admin <root@log.foo.co.jp>
- To: admin@foo.co.jp
- Date: Thu, 1 Nov 2001 22:01:49 +0900 (JST)
- Subject: linkDown Trap
- nspixp2-gw.foo.co.jp
- 192.168.244.21
- system.sysUpTime 24:10:03:09.12
- .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapOID .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTraps.linkDown
- interfaces.ifTable.ifEntry.ifIndex.1 1
- interfaces.ifTable.ifEntry.ifDescr.1 "Fddi1/0/0"
- interfaces.ifTable.ifEntry.ifType.1 Fddi
- enterprises.9.2.2.1.1.20.6 "administratively down"
- .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapEnterprise enterprises.9.1.48

フリーソフトウェアによる ネットワーク監視

Snort IDS
(Intrusion Detection System)

Snort IDS

- http://www.snort.org/
  - **Version 1.8.2** (2001/10/20現在)
- パケットモニタ型IDS(Intrusion Detection System：侵入検知システム)
- Libpcapを用いてパケットをモニタし、侵入パターンルールセット(シグネチャ)とマッチングをすることで、不正侵入を検出する
  - Preprocessor：パターンマッチングの前処理モジュール
    - Portscanチェックやdefragされたパケットの再構成などを行う
  - Ruleset：障害検出ルールセット
  - Output Module：検出されたイベントの出力加工を行う
    - SQL DBMS, syslog, SMB/WinPopup
    - XML形式, Tcpdump形式

Snortプリプロセッサ

- minfrag　小さなフラグメントパケットの検出を可能とする
- http-decode　httpプロトコルで使用されるURIを正しく識別させる
- portscan　ポートスキャンを検出を行う
- portscan-ignorehosts
  ポートスキャン検知から特定のホストを除外する
- defrag　プラグメント化されたパケットの評価を可能とする
- stream　ストリームパケットの評価を可能とする
- spade　統計的手法による異常検出を行う。(実験用)

Snortルールセット1

- exploit.rules　バッファーオーバーフローなどの攻撃を検知
- scan.rules　一部のステルスポートスキャンやスキャナツールを検知
- finger.rules　fingerサービスに対するプロービングや攻撃を検知
- ftp.rules　ftpサービスに対するプロービングや攻撃などを検知
- telnet.rules　telnetサービスに対するアクセスや攻撃などを検知
- smtp.rules　smtpサービスに対する攻撃などを検知
- rpc.rules　rpcサービスに対するプロービングや攻撃などを検知
- rservices.rules　r系のサービスに対するアクセスや攻撃などを検知
- backdoor.rules　さまざまなバックドアツールを検出

- web-misc.rules　その他のhttpに対する攻撃などを検知
- icmp.rules　icmpに対するプロービング等を検知
- misc.rules　traceroutやrootkitなどに関する通信を検知

Snortルールセット2

dos.rules	DoSを検出するためのルール
ddos.rules	分散DoSのクライアント・サーバ間のコントロール通信を検知
dns.rules	dnsサービスに対するプロービングや攻撃などを検知
netbios.rules	netbios関係のアクセスや攻撃などを検知
web-cgi.rules	httpサーバ上で動作するcgiに対する攻撃等を検知
web-coldfusion.rules	Webアプリケーションサーバ ColdFusionへの攻撃等を検知
web-frontpage.rules	MS-Frontpageを悪用した行為を検知
web-iis.rules	MS-IISサーバに対する攻撃などを検知

Snort.confの例

```
var HOME_NET 192.168.0.100/32
var EXTERNAL_NET any
var DNS_SERVERS [192.168.0.2/32,192.168.0.3/32]

preprocessor http_decode: 80 -unicode -cginull
preprocessor rpc_decode: 111
preprocessor telnet_decode
preprocessor portscan: $HOME_NET 50 5 e/portscan.log
preprocessor portscan-ignorehosts: $DNS_SERVERS

include classification.config

include bad-traffic.rules
include exploit.rules
include scan.rules
include finger.rules
include ftp.rules
include telnet.rules
include smtp.rules
include rpc.rules
include rservices.rules
include dos.rules
include ddos.rules
include dns.rules
include tftp.rules
include web-cgi.rules
include web-coldfusion.rules
include web-frontpage.rules
include web-iis.rules
include web-misc.rules
##include sql.rules
##include x11.rules
##include icmp.rules
##include netbios.rules
##include misc.rules
include attack-responses.rules
include backdoor.rules
include shellcode.rules
# include policy.rules
# include info.rules
# include icmp-info.rules
# include virus.rules
include local.rules
```

Snortの不正アクセス検出例

```
[**] [100:1:1] spp_portscan: PORTSCAN DETECTED to port 22 from
XXX.XXX.163.130 (STEALTH) [**]
10/29-16:39:12.711564
```

```
[**] [1:624:1] SCAN SYN FIN [**]
0/29-16:39:12.473149 XXX.XXX.163.130:22 -> XXX.XXX.XXX.XXX:22
TCP TTL:33 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x1F5DE10  Ack: 0x45EC1B65  Win: 0x404  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS198]
```

```
[**] [100:2:1] spp_portscan: portscan status from XXX.XXX.163.130: 2
connections across 1 hosts: TCP(2), UDP(0) STEALTH [**]
10/29-16:39:23.569441
```

```
[**] [100:3:1] spp_portscan: End of portscan from XXX.XXX.163.130:
TOTAL time(11s) hosts(1) TCP(2) UDP(0) STEALTH [**]
10/29-16:40:04.928347
```

```
[**] [1:620:1] SCAN Proxy attempt [**]
10/29-19:32:43.527741 XX.XXX.61.20:3501 -> XXX.XXX.XXX.XXX:8080
TCP TTL:115 TOS:0x0 ID:5422 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1B488653  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1380 NOP NOP SackOK
```


SnortによるPortscan検出例

```
Oct 15 02:55:04 XXX.XXX.132.61:21 -> XXX.XXX.XX3.131:21 SYNFIN ******SF
Oct 15 02:55:22 XXX.XXX.132.61:21 -> XXX.XXX.XX7.43:21 SYNFIN ******SF
Oct 15 02:55:22 XXX.XXX.132.61:21 -> XXX.XXX.XX7.46:21 SYNFIN ******SF
Oct 15 02:55:23 XXX.XXX.132.61:21 -> XXX.XXX.XX7.78:21 SYNFIN ******SF
Oct 15 02:55:23 XXX.XXX.132.61:21 -> XXX.XXX.XX7.90:21 SYNFIN ******SF
Oct 15 21:52:22 XXX.XXX.132.61:53 -> XXX.XXX.XX3.131:53 SYNFIN ******SF
Oct 15 21:52:43 XXX.XXX.132.61:53 -> XXX.XXX.XX7.90:53 SYNFIN ******SF
Oct 16 18:19:49 XXX.XXX.249.75:21 -> XXX.XXX.XX3.131:21 SYNFIN ******SF
Oct 16 18:20:08 XXX.XXX.249.75:21 -> XXX.XXX.XX7.90:21 SYNFIN ******SF
Oct 23 04:31:12 XXX.XX.214.61:21 -> XXX.XXX.XX3.131:21 SYNFIN ******SF
Oct 23 04:31:31 XXX.XX.214.61:21 -> XXX.XXX.XX7.46:21 SYNFIN ******SF
Oct 23 04:31:32 XXX.XX.214.61:21 -> XXX.XXX.XX7.90:21 SYNFIN ******SF
Oct 23 08:39:38 XXX.XXX.132.61:53 -> XXX.XXX.XX3.131:53 SYNFIN ******SF
Oct 25 11:54:07 XX.XX.57.77:22 -> XXX.XXX.XX3.131:22 SYNFIN ******SF
Oct 25 11:54:17 XX.XX.57.77:4111 -> XXX.XXX.XX3.131:22 SYN ******S*
Oct 25 11:54:25 XX.XX.57.77:22 -> XXX.XXX.XX7.39:22 SYNFIN ******SF
Oct 25 11:54:25 XX.XX.57.77:22 -> XXX.XXX.XX7.40:22 SYNFIN ******SF
Oct 25 11:54:25 XX.XX.57.77:22 -> XXX.XXX.XX7.46:22 SYNFIN ******SF
Oct 25 11:54:25 XX.XX.57.77:22 -> XXX.XXX.XX7.47:22 SYNFIN ******SF
Oct 25 11:54:25 XX.XX.57.77:22 -> XXX.XXX.XX7.48:22 SYNFIN ******SF
```

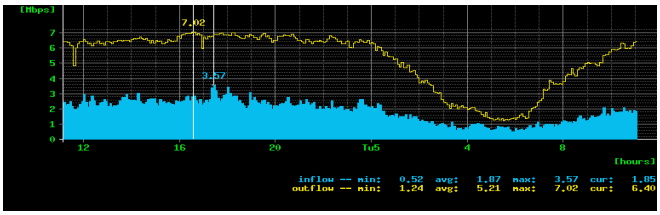
フリーソフトウェアによる ネットワーク監視

MRTG
(Multi Router Traffic Grapher)

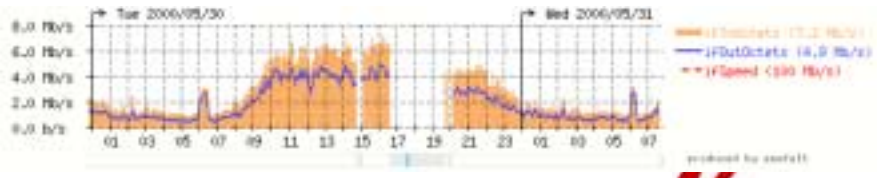
3-7, December 2001 Pacifico Yokohama Internet Week 2001

これまでを見る監視ツール

- Shephard

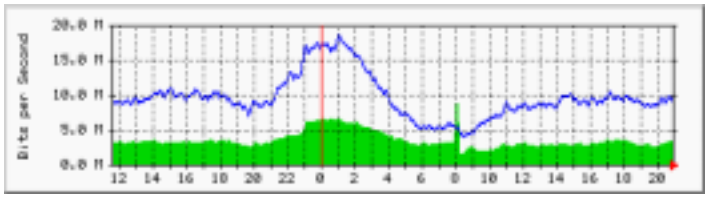


- Seafelt



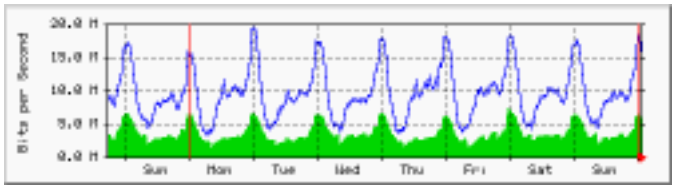
これまでを見る監視ツール

- MRTG (Multi Router Traffic Grapher)



MRTGとは

- http://www.mrtg.org/
- http://www.mrtg.jp/doc/ (日本語翻訳サイト)
- MRTG : Multi Router Traffic Grapher
- 2系列のデータを基に集計を行い、短期・中期・長期トレンドグラフを生成するツール



MRTGの特徴

- ほとんどのUnixプラットフォームとWindowsNT上で稼動
- 独自にSNMPを実装。外部のSNMP Packageは不要
- 定期的にログをサマリーするデータ管理を行っており、ログファイルのサイズが大きくならない
- 半自動のコンフィグ作成ツールが付属
- 日・週・月・年ごとにデータを集計したWEBページを結果として生成する
- コンフィグからindexを簡単に生成するツールが付属
- インストールにはLibgdが必要
  - `http://www.boutell.com/gd/`

MRTGの新機能 － 1

- v 2.9より以下の機能が追加
  - Target[ ]への新たな指定方法追加
    - 以下のInterface属性をキーに、当該インタフェースを特定する
      - MAC address指定
      - Description指定
      - Interface Name指定
      - Interface Type指定
  - Threshold指定
    - 測定対象に対して最小値・最大値を設定し、この閾値を超えたデータ計測された際に外部スクリプトを起動することができる。
      - あふれた際にメールで通知するなどが可能
  - Daemon化指定： RunAsDaemon: , Interval:
    - MRTGのPerl ProcessをDaemonにすることにより、Perl起動にかかるFork処理をなくし、高速化する。計測周期はInterval:により指定する

MRTGの新機能 – 2

- v 2.9より以下の機能が追加(続き)
  - 並列照会: Forks:
    - 照会処理の並列化度を指定する。このオプション指定により、無応答の照会処理がボトルネックとなって、プロセスの完了時間が延びることを防ぐことが可能となる
    - Unixプラットフォームでのみサポート
  - RRDToolsとの統合：LogFormat: rrdtool
    - logの管理をRRDToolを使用することにより、劇的な高速化を実現するこれまでのlogについては本オプション指定により自動的にデータ移行がなされる
    - このオプション指定することで、グラフの作成は測定時はなされず、付属の14all.cgiによりon the flyで(要求のたびに)作成をする
    - 10倍以上高速になることも

MRTG – cfgmaker - 1

- mrtg付属の簡易設定ツール
  - `cfgmaker { <option> } <community>@<target>`
  - `<community> : snmp community string`
  - `<target> : target address or hostname`
  - 例: `$ cfgmaker himitsu@ix-gw.foo.co.jp > ix-gw.cfg`
- communityとtargetを指定するだけで機器に存在するインタフェースをサーチし、`ifInOctets/ ifOutOctets`を測定する設定の大部分を作成する
  - `syscontact/location`などの情報からコメントも自動作成
  - 保守停止しているインタフェースについてはコメントとして作成
  - 追加設定は `WorkDir:` だけでほぼ動く

MRTG – cfgmaker - 2

- v2.9から–ifref optionが追加され、以下のTarget指定のコンフィグを作成可能
  - `--ifref=nr ... interface references by Interface Number(default)`
  - `--ifref=ip ... by IP Addres`
  - `--ifref=eth ... by Ethernet Number`
  - `--ifref=descr ... by Interface Description`
  - `--ifref=name ... by Interface Name`
  - `--ifref=type ... by Interface Type`

MRTG - cfgmaker の出力結果

```
# Add a WorkDir: /some/path line to this file

######################################################################
# Description: Cisco Internetwork Operating System Software IOS (tm) GS ...
#     Contact: admin@foo.co.jp
# System Name: ix-gw.foo.co.jp
#    Location: PA, CA, US
#.....................................................................…

Target[ix-fddi.foo.co.jp]: 1:himitsu@192.168.98.133
MaxBytes[ix-fddi.foo.co.jp]: 12500000
Title[ix-fddi.foo.co.jp]: ix-gw.foo.co.jp (ix-fddi.foo.co.jp): Fddi1/0/0
PageTop[ix-fddi.foo.co.jp]: <H1>Traffic Analysis for Fddi1/0/0
 </H1>
 <TABLE>
   <TR><TD>System:</TD><TD>ix-gw.foo.co.jp in Otemachi 5F</TD></TR>
   <TR><TD>Maintainer:</TD><TD></TD></TR>
   <TR><TD>Interface:</TD><TD>Fddi1/0 (1)</TD></TR>
   <TR><TD>IP:</TD><TD>ix-fddi.foo.co.jp (172.16.0.2)</TD></TR>
   <TR><TD>Max Speed:</TD>
       <TD>12.5 MBytes/s (fddi)</TD></TR>
  </TABLE>
```

MRTGの使い方

- 独立コマンドとして作成されており、通常はcronにて定期的に起動する。(default：5分間隔)
  - `# crontab -l`
    `0-59/5 * * * /usr/local/sbin/mrtg /usr/local/etc/ix-foo.cfg`
    `#`
- RunAsDaemonしている際には以下のような設定をコンフィグに投入し、コマンドを投入
  - `RunAsDaemon:Yes`
    `Interval:5`
  - `$ mrtg --user=mrtg_user --group=mrtg_group mrtg.cfg`
- データ収集指定はconfigファイルのTargetレコードにて指定

MRTG - Targetの指定法

- Keyword: Target - データ収集項目を指定
  - 例:
  - `Target[gw1-3]: 3:himitsu@gw1.foo.co.jp`
  - `Target[gw1-err-3]:`
    `ifInErrors.3&ifOutErrors.3:himitsu@gw1.foo.co.jp`
  - `Target[gw1-if-1]: -/10.0.0.101:himitsu@gw1.foo.co.jp`
  - `Target[gw1-pingloss]: `/usr/local/bin/check_loss.sh gw1``
- SNMPデータの収集
- 外部コマンド結果の埋め込み収集

MRTG - Targetの指定法:SNMP 1

- SNMPデータの収集
  - Target[<target name>]:
    <target kind>:<community>@<address>
    - <target name> : 測定機器の名称
    - <target kind> : 測定項目
    - <community> : 測定機器に設定している community string
    - <address> : 測定機器のアドレス・ホスト名

MRTG - Targetの指定法:SNMP 2

- SNMPデータ収集指定方法
  - Port指定(ifIndex指定)
  - SNMP OID指定 / SNMP MIB symbol指定
  - Interface Address指定
  - 組み合わせ指定
  - 新規追加の指定方法
    - MAC address指定
    - Description指定
    - Interface Name指定

MRTG - Targetの指定法:SNMP 3

- Port指定(ifIndex指定)
  - SNMP Client側で管理しているPort番号(ifIndex)を使ってデータ照会する。
  - ifInOctetsとifOutOoctetsを測定
- 例1:Target[gw1-3]: 3:himitsu@gw1.foo.co.jp
  - gw1.foo.co.jpに収容されているifIndex=3のInterfaceに関して ifInOctets/ifOutOctetsを測定
- 例2:Target[gw1-3]: -3:himitsu@gw1.foo.co.jp
  - 例1のIn/Outを逆にしてデータ収集する

MRTG - Targetの指定法:SNMP 4

- SNMP OID指定 / SNMP MIB symbol指定
  - SNMP OID(Object ID)またはMIB symbolを指定し、データ照会する。
  - 変数1、変数2は"&"で連結指定する
- 例3: Target[gw1-err-3]: ifInErrors.3&ifOutErrors.3:himitsu@gw1.foo.co.jp
  - gw1.foo.co.jpに収容されているifIndex=3のInterfaceに関して ifInErrors/ifOutErrorsを測定
- 例4: Target[gw1-err-3]: 1.3.6.1.2.1.2.2.1.14.3& 1.3.6.1.2.1.2.2.1.20.3:himitsu@gw1.foo.co.jp
  - 上の例のOID指定

ちょっと脇道 - よく使うSNMP OID/MIB Symbols

- [interfaces.ifTable.ifEntry] group
  - 1.3.6.1.2.1.2.2.1.1 : ifIndex
  - 1.3.6.1.2.1.2.2.1.2 : ifDescr
  - 1.3.6.1.2.1.2.2.1.3 : ifType
  - 1.3.6.1.2.1.2.2.1.7 : ifAdminStatus
  - 1.3.6.1.2.1.2.2.1.8 : ifOperStatus
  - 1.3.6.1.2.1.2.2.1.10 : ifInOctets
  - 1.3.6.1.2.1.2.2.1.16 : ifOutOctets
  - 1.3.6.1.2.1.2.2.1.11 : ifInUcastPkts
  - 1.3.6.1.2.1.2.2.1.17 : ifOutUcastPkts
  - 1.3.6.1.2.1.2.2.1.13 : ifInDiscards
  - 1.3.6.1.2.1.2.2.1.19 : ifOutDiscards
  - 1.3.6.1.2.1.2.2.1.14 : ifInErrors
  - 1.3.6.1.2.1.2.2.1.20 : IfOutErrors

MRTG - Targetの指定法:SNMP 5

- Interface Address指定1
  - パッケージタイプのルーター・スイッチはインタフェースの増減設によりPort番号(ifIndex)が変化する
  - loopbackやtunnel Interfaceのような仮想インタフェースもSNMP上では一つのポート番号をもつ
    - → ifIndexの割付が変化する可能性がある
  - 機器の構成変更の度に設定変更をさけるためにインタフェースに割り振られたアドレスをキーにしてデータ照会を行う
    - numberedで使われていることが前提！
  - デフォルトでは`ifInOctets`と`ifOutOctets`を測定

MRTG - Targetの指定法:SNMP 6

- Interface Address指定2
- 例5:`Target[gw1-if-1]: /10.0.0.101:himitsu@gw1.foo.co.jp`
  - gw1.foo.co.jpに収容されている10.0.0.101のInterfaceに関してifInOctets/ifOutOctetsを測定
- 例6:`Target[gw1-if-1]: -/10.0.0.101:himitsu@gw1.foo.co.jp`
  - 例5のIn/Outを逆にしてデータ収集する

MRTG - Targetの指定法:SNMP 7

- 組み合わせ指定
  - Interface address指定とOID/MIB symbol指定を組み合わせる
- 例7:`Target[gw1-if-1-disc]: ifInDiscards/10.0.0.101& ifOutDiscards/10.0.0.101:himitsu@gw1.foo.co.jp`
  - gw1.foo.co.jpに収容されている10.0.0.101のInterfaceに関してifInDiscards/ifOutDiscardsを測定
- 例8:`Target[gw1-if-1-disc]: 1.3.6.1.2.1.2.2.1.13/10.0.0.101& 1.3.6.1.2.1.2.2.1.19/10.0.1.101:himitsu@gw1.foo.co.jp`
  - 例7のOIDパターン

MRTG - Targetの指定法:SNMP 8

- Interface Name指定
  - Interface Address指定はIP Addressをキーにしているために、switching hubのようにポートごとにアドレスをもたないものには適用できない。
  - これの状況に適応するためにInterfaceに割り振られたInterface名前をキーにしてデータ照会を行う
  - デフォルトでは`ifInOctets`と`ifOutOctets`を測定
- 例9:
  - `Target[sw1-2-11]: #2/11:himitsu@sw1.foo.co.jp`
    - `Target[sw-2-11]: -#2/11:himitsu@sw1.foo.co.jp`
    - `Target[sw-3-7]: 1.3.6.1.2.1.2.2.1.14#3/7&1.3.6.1.2.1.2.2.1.20#3/7:himitsu@sw1.foo.co.jp`
    - `Target[sw-3-7]: ifInErrors#3/7&ifOutErrors#3/7:himitsu@sw1.foo.co.jp`

MRTG - Targetの指定法:SNMP 9

- Interface Description指定
  - Interface Address指定では、故障時にポートの入れ替えなどが発生した際に、MRTG側の設定を修正しなければならない
  - サーバー側で対応するよりも収容変更先の装置の設定情報を元に変更できたほうが適応範囲が広いことから、これらのキーとしてInterfaceに割り振られるDescriptionをキーにデータ照会を行う
  - デフォルトでは`ifInOctets`と`ifOutOctets`を測定
- 例9:
  - `Target[sw1-2-11]: ¥to_web1:himitsu@sw1.foo.co.jp`
    - `Target[sw-2-11]: -¥to_web1:himitsu@sw1.foo.co.jp`
    - `Target[sw-3-7]: 1.3.6.1.2.1.2.2.1.14¥to_web1&1.3.6.1.2.1.2.2.1.20¥to_web1:himitsu@sw1.foo.co.jp`
    - `Target[sw-3-7]: ifInErrors¥to_web1&ifOutErrors¥to_web1:himitsu@sw1.foo.co.jp`

MRTG -

Targetの指定法:コマンド埋め込み

- コマンド埋め込み指定
  - `Target[<target name>]: `<command>``
    - `<target name>` ： 測定機器の名称
    - `<command>` ： 測定コマンド
      - “`“：バックシングルコーテーションでくくるのがミソ
  - コマンドの結果として4行の値が必要
    - 1行目：第1変数、通常 `incoming bytes`数
    - 2行目：第2変数、通常 `outgoing bytes`数
    - 3行目：文字列、`target`の`uptime`
    - 4行目：文字列、`target`の名称

MRTGによる品質計測

- 埋め込みコマンドによりSNMPでは計測が難しい品質測定なども可能となる
- 例：特定の2点間のpacket lossの定常監視
  - 一定間隔でpingによる定期監視を実施

```
# ping -i 0.02 -c 100 ftp.foo.co.jp
PING ftp.foo.co.jp (192.168.101.238): 56 data bytes
.
--- ftp.foo.co.jp ping statistics ---
100 packets transmitted, 95 packets received, 5% packet loss
round-trip min/avg/max/stddev = 0.161/0.164/0.221/0.006 ms
#
```

  - `-i 0.02 : supervisor only option.`
    FreeBSDのpingにおける指定。送出間隔を20ms。
    ネットワークに高負荷を強いることから取り扱い注意

MRTGによる品質計測 - check_loss.sh

- pingの出力結果からpacket lossのデータを抽出
  - 100 packets transmitted, 95 packets received, 5% packet loss

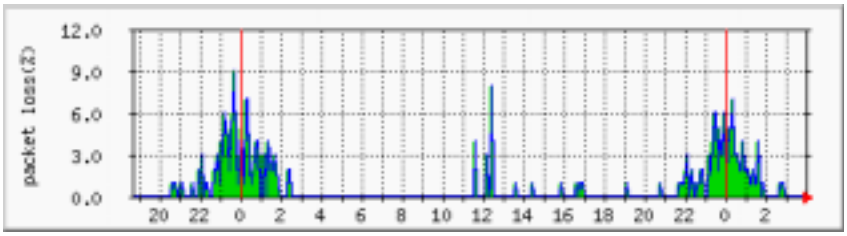
```
# cat /usr/local/bin/check_loss.sh
#!/bin/sh
/sbin/ping -f -c 100 $1 | /usr/bin/sed 's/%//g' | /usr/bin/awk '
    /packet loss/ { printf("%d¥n%d¥n", $7, $7)
  }'
echo 0 ; echo $*
# /usr/local/bin/check_loss2.sh ftp.foo.co.jp
5
5
0
/usr/local/bin/check_loss.sh ftp.foo.co.jp
#
```


MRTGによる品質計測 - ping-loss.cfg

```
# cat ping-loss.cfg
WorkDir: /usr/local/etc/www/mrtg/ping-loss

Target[pingloss-ftp]: `/usr/local/bin/check_loss.sh ftp.foo.co.jp`
Title[pingloss-ftp]: ftp.foo.co.jp - pingloss
MaxBytes[pingloss-ftp]: 100
PageTop[pingloss-ftp]: <H1> ftp.foo.co.jp - pingloss </H1>
YLegend[pingloss-ftp]: packet loss(%)
ShortLegend[pingloss-ftp]: %
LegendI[pingloss-ftp]:  loss:
LegendO[pingloss-ftp]:  loss:
Legend1[pingloss-ftp]: packet loss
Legend2[pingloss-ftp]: packet loss
Legend3[pingloss-ftp]: Maximal 5 Minute packet loss
Legend4[pingloss-ftp]: Maximal 5 Minute packet loss
Options[pingloss-ftp]: noinfo, growright, gauge, nopercent
#
```

MRTGによる品質計測 - 結果



3-7, December 2001 Pacifico Yokohama Internet Week 2001

フリーソフトウェアによる ネットワーク監視

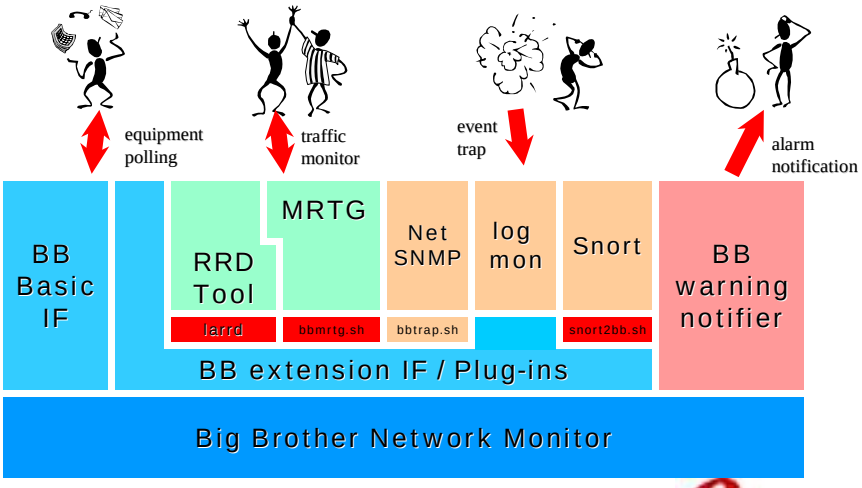
ツール間連係：

Big Brother --- RRDTool
MRTG
Snort

BigBrother – RRDTool/MRTG/Snortとの連携

- 現状監視と経過監視はともに必要
- 経過監視し、その傾向によりアラームを自発的に通知したい
  - System稼動状況をグラフ化する
  - トラフィックに閾値を設けて監視したい
  - IDSのアラームを集中監視したい
- Big Brotherのextensionによりツール連携を行う
  - larrd - RRDToolsとの連携
  - bbmrtg.sh - MRTGとの連携
  - Snort2bb.pl – Snortとの連携

フリーソフトによる統合監視システム



RRDToolとの連係：larrd

- larrd: loadavg rrdtool -> latest v 0.42
- http://larrd.packetpushers.com/
- Big Brother Clientが各監視対象から取得したデータをRRDToolによりグラフ化する
  - 対象データ：load average, Disk Usage, Memory, SWAP, bind, TCP Connection Time, (Memory Usage, CPU idle,) …
- グラフ作成のみに特化しており、larrdは閾値を設定したトラフィックアラーム監視は行わない
- 反面、設定は簡単であり、以下の設定だけで動作する
  - 指定ディレクトリへの展開
  - シンボリックリンクの作成
  - $BBHOME/etc/bbdef.shへの登録
    - $BBEXT変数へのエントリー追加

eAccess

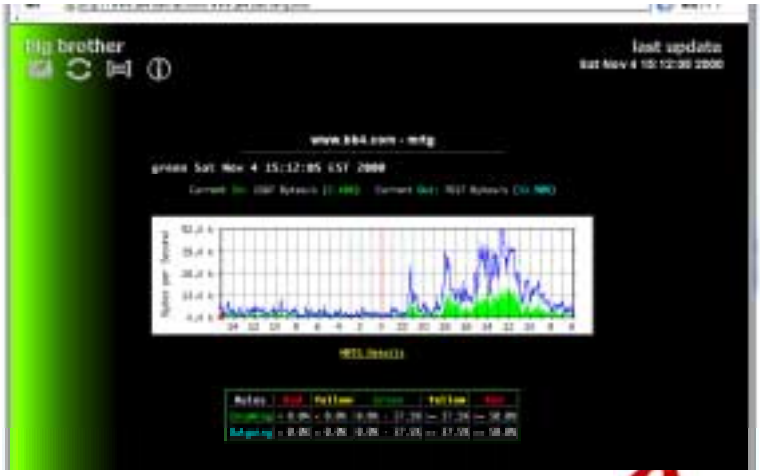
larrd - 画面

eAccess

bbmrtg.sh

- MRTGとの連携を図るPlug-in
- http://www.deadcat.net/BB/ext/bbmrtg.sh
- MRTGが動いているサーバーもしくはMRTGのlogファイルにアクセスできるBB監視サーバー上で稼動
- データの取得はMRTGが行い、このPlug-inはMRTGが生成したlogファイルのチェックを行う
- 閾値を設定して、それを超えたらアラームをあげることが可能

bbmrtg.sh



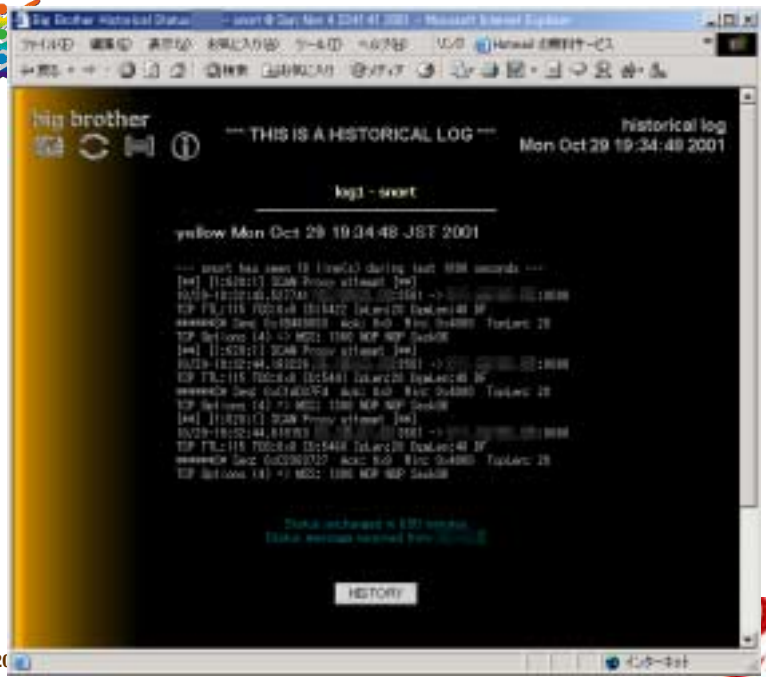
設定例 - bbmrtg.sh

- 閾値の設定はbbmrtg.shの中に埋め込まれている
- 新規追加、閾値の変更の際にはPlug-in自体を直接変更しなければいけない
  - 更新頻度の多いネットワークでの使用は無理ではあるが、SOHOにおいてはかなり有用

```
BBMRTGCFG="¥
# MRTG device    BB Host        svc  Yellow              Red                 Max   Unit
mrtg1.foo.co.jp  bb1.foo.co.jp  mrtg 24000               32000               64000 Bytes/s  In Out
mrtg1.foo.co.jp  bb1.foo.co.jp  mrtg 750:24000           500:32000           64000 Bytes/s  In Out
mrtg1.foo.co.jp  bb1.foo.co.jp  mrtg 750:24000:700:24500 500:32000:550:32500 64000 Bytes/s  In Out
mrtg1.foo.co.jp  bb1.foo.co.jp  mrtg 50%                 75%                 64000 Bytes/s  In Out
"
```


IDSとの連携：snort2bb.pl

- http://www.deadcat.net/cgi-bin/download.pl?section=1&file=snort2bb-000831.tar.gz
- Snort IDS(Intrusion Detection System)のと連携するための BB Extension Script
- Snortが稼動しているホストにてDaemonとして稼動させる
- デフォルトでは/var/log/snort/alertを監視し、新規イベント検出により BBDISPLAYサーバにアラームを上げる
- 稼動させるためには以下のperl追加モジュールが必要
  - Time::HiRes (Time-HiRes-01.20.tar.gz以上)
  - File::Tail (File-Tail-0.98.tar.gz以上)
- 複数のIDSの結果を一括して監視できる
- 監視制度をあげるためには誤検知をなくすためのシグネチャーファイルの調整が必須



Snort2bb.pl検出例

第2部のまとめ

- フリーソフトウェアはみんなのニーズが集約された成果物
  - 現場のノウハウが集約 → 使わない手はない
- 全てをひとつで補うことはできないが、適材適所の組み合わせで簡単にシステム化可能
- 要はやる気と根気。
- それとちょっとしたコツさえつかめれば、かなりの部分まではフリーソフトで幸せになれる

第3部：ネットワーク管理に関するTIPS

TIPS - まずは

- ツールの挙動確認はまずオフラインで
  - 監視・測定ツールでネットワークに障害を与えることができる
- 時間を合わせましょう
  - 絶対基準のひとつは当然、時刻
  - NTPでサーバーと監視機器の時間を同期させる
  - Timezoneも一致させないとわからなくなる
    - UTC or JST-9

TIPS - ping編1

- ping checkの間隔に注意
  - ルーター/スイッチは以外にflood pingに弱い
  - 下手すると内部からのDoS Attackになりかねない
- 広域で遠隔監視する際にはチューニングが必要
  - 手早く数をこなすためには、送出間隔とタイムアウトとリトライ回数のバランスが必要
  - ローカルと違い、単純に一発落ちたからといって死んでいるわけではない。
  - リトライをうまく使い対応する
- SwitchのAuto Negotiationは信じない！
  - これは基本中の基本
  - 今、no errorだからといって、明日、no errorが続くとは限らない
    - 突然、duplexが full <-> half と変わってしまうことがある

TIPS – ping編2

- ショートパケットが通ったからといって安心できない。開通確認はロングパケットで
  - トラフィックが多くなってくるとパケットが落ちるところも多々ある
  - ATM Megalink回線では必須。シェーピングレートの設定が失敗しているといきなり品質劣化して、通信障害となる
    - 私はpacket size=1500byte, count=1000以上、送出Interval=40msとかで試験してるかな
  - スイッチのduplexミスマッチもこれならはっきり検知できます
- Internet経由の監視は タイムアウト＞1000msec
  - テレホタイムは特に揺らぎが大きいため、マージンをとらないと誤検出が増える

TIPS – traceroute編

- tracerouteは絶対ではない
  - 行きと帰りは非対称。同じ道を通るとは限らない
  - ソースルートオプションは絶対ではない
- looking glass/traceroute siteを使えば外から確認できる
  - http://www.traceroute.org/
  - http://nitrous.digex.net/
  - http://neptune.dti.ad.jp/
  - http://www.geektools.com/traceroute.html

TIPS - 監視サーバー編1

- 監視サーバの置き場所には注意
  - より詳細な監視をするためには最もコアになる装置のそばに置く
- 監視サーバーの画面は外に公開するものか？
  - .htaccess規制もやっておきましょう
  - Proxyに注意。
    - 「.htaccess」で規制していても、proxyが中にいてopenな状態だと意味がない。
    - ACLでno-cacheにしましょう
- http portを変更する (http port != 80)

TIPS - 監視サーバー編2

- 監視対象拡大に伴う問題
  - 規模が大きくなると、NMSがポーリングして統計処理を行う時間も増加する
  - 監視対象機器を適正な数に抑えないと…
    - 次のポーリングタイミングまで計測が終らない
  - 適正範囲に分割が必要
    - 規模拡大時に見落としやすいので注意

TIPS - BB編1

- Longer than Sleeptime XXXがでたら環境限界の印
  - BBのシステムログは$BBHOME/BBOUT。これをチェック！
  - Longer than Sleeptimeメッセージは監視間隔以内に監視が終わらないというシステムメッセージ
    - `Thu Nov 1 06:12:07 JST 2001 bbrun: (/usr/local/bb/ext/fping.sh) Runtime 517 longer than Sleeptime 300`
    - `Thu Nov 1 06:13:21 JST 2001 bbrun: (/usr/local/bb/bin/bb-network.sh) Runtime 346 longer than Sleeptime 300`
  - マシンスペックのグレードアップ・監視サーバ分割を視野にいれた、システム環境・チューニングを含めた見直しが必要

TIPS - BB編2

- Big Brotherの高速化：fping + fping.sh
  - `http://www.fping.org/`
  - `http://www.deadcat.net/cgi-bin/download.pl?section=1&file=fping.sh`
  - fpingによりping試験を高速化
    - bbdef.sh内にて“CONNTEST=FALSE”としてBBのping試験を停止する必要あり
- Big Brotherサーバのシステム監査ログには注意が必要
  - BBの基本はshell scriptとなっているために一回の監視フェーズにおいて数十のプログラムが起動される
    - Accountingログが短時間に巨大になる
    - ログ領域の拡大。細かなメンテナンス
    - もしくは容量をアカウンティングを停止

TIPS - SNMP編1: アクセス規制

- SNMPに関する規制
  - SNMPは便利。しかし便利なものには必ず穴がある
    - セキュリティーホールになりやすい
    - SNMPでネットワークを落とすことも可能！
- Default communityはつかわない
  - Read only community != `public`
  - Write community != “private”
- 不要なrw、rwaはできるだけ使えないように設定する

TIPS - SNMP編2: アクセス範囲の限定

- SNMPクライアントにはアクセス規制が必須
  - 意外に狙われているルーター・スイッチ・www server
- SNMP package
  - libwrapをlink。hosts.allow/hosts.denyでアクセス規制する
    - ./configure --with-libwrap=...
- Cisco
  - SNMPアクセス規制用access-listの設定
- そんな機能のない装置は...
  - Private address blockにいれてしまう
  - ガードの低い装置をルーティング的にInternetから隔離する (例:Switching Hub, ...)

TIPS - SNMP編3: Interface高速化に伴う問題

- カウンター 一周問題
  - ifInOctets/ifOutOctes は32bit正数
  - 5分ごとに各数値を集計する場合、約114Mbpsを越えるトラフィックが生成されるネットワークではカウンターが一周する
  - MRTG 2.9系列にてSNMPv2c 64bit counter MIBを使用する
  - それ以外だと測定周期の調整という方法が必要となる
    - Default=5分以下の間隔にて測定を行う
    - `0-59/3 * * * /usr/local/sbin/mrtg ./ix-foo.cfg`

TIPS - SNMP編4: ifIndex問題

- パッケージタイプのルーター・スイッチは以下の事象においてifIndexとinterfaceの割付が変わる可能性がある
  - パッケージ障害交換
  - パッケージの増減設
  - 仮想インタフェースの増減設
  - その他...
- インタフェースの増減設が伴う際には監視ツールの設定を合わせて見直す

TIPS - SNMP編5: 使えるNet-SNMPコマンド例(V 4.2.2)

- `$ snmpwalk 10.0.0.1 himitsu 1`
- `$ snmpwalk 10.0.0.1 himitsu 2`
- `$ snmpwalk 10.0.0.1 himitsu ifDescr`
- `$ snmpwalk 10.0.0.1 himitsu ifType`

- `$ snmptranslate -IR ifInDiscards`
  - OIDを表示
- `$ snmptranslate –Tdp -IR ifInDiscards`
  - OIDの他にMIB Tree及び詳細説明を表示
- `$ snmptranslate –Tp 2`
  - Interface(2) MIB配下のMIB Treeを表示
- `$ snmptranslate -On .1.3.6.1.2.1.2.2.1.1`
  - OIDをMIB Symbolに変換して表示
- `$ snmptranslate -On -Tda .1.3.6.1.2.1.2.2.1.1`
  - 上のコマンドに詳細説明を追加

TIPS – MRTG編1

- データの方向性に注意
  - 対向している装置で同じポートを測定するとIn/Outが逆の結果がでる
  - 対外線を出口として、ここを起点にデータが流れるように設定すると考えやすい
- データの単位に注意
  - ifInOctets/ifOutOctetsはOctet単位系
  - 回線・物理接続速度はbps。つまりbit単位系
    - Options[hoge] bitsした上でMaxbytes[hoge]を8倍する
- IP address/MAC address/Comment指定Targetを効果的に使う

TIPS – MRTG編2

- Cronからのメッセージには気をつけろ！
  - 必ずMRTGのエラーメッセージは取得できるようにする
    - /etc/aliases
    - ~/.forward
- まずいメッセージ
  - Config Error
  - No Response
  - Lockfile found

TIPS – MRTG編3

- 非常にまずいメッセージ

```
From: root@mrtg1.eaccess.ne.jp (Cron Daemon)
To: mrtg@mrtg1.eaccess.ne.jp
Date: Fri, 13 Oct 2000 02:03:16 +0900 (JST)
Subject: Cron <mrtg@mrtg1> /usr/local/mrtg/mrtg
/usr/local/mrtg/conf/mrtg.cfg
--

ERROR: I guess another mrtg is running.
A lockfile (/usr/local/mrtg/conf/mrtg.cfg_l) aged 303 seconds is
hanging around.
If you are sure that no other mrtg is running you can remove the
lockfile
```


TIPS – MRTG編4

- じゃ、逆手にとって、エラーメッセージによるネットワーク監視
  - 5分に毎に起動されるSNMP health checkという観点もある
    - MRTGのエラーメッセージを/dev/nullにするのはもったいない
  - 経験的予兆
    - 同じインタフェースのno responseエラーが続いて上がってきたら、該当インタフェース回線のダウンか故障の可能性がある
    - どっと、まとめてエラーが帰ってきたら、ルータやスイッチなどのネットワーク障害が発生している可能性が高い

MRTG番外編: 限界への挑戦1

- まずはメモリ追加
  - なってたってオンメモリ
- Forks: 指定で並列Query
  - 測定対象がこけてだまったときにも保険になる。
- 起動順番を調整する。スタート基準は1分間隔
  - 0,5分スタート組、1,6分スタート組、2,7分スタート組、3,8分スタート組、4,9分スタート組
- ブロックサイズの変更による速度改善
  - 最近のOSは大きくなっているからあまり気にしなくていいのですが、FreeBSD4.1 1024byte, Solaris2.6 4096byte, …
    - ちょっと小さいので、フォーマット時には大きくとる必要あり。
    - 32k/Blockぐらいでいいのでは
  - i-node数の減少には注意

MRTG番外編: 限界への挑戦2

- ひとつのconfigにできるだけつっこむ。
  - 作業ディレクトリ指定: directory[]: で複数の測定をひとつのconfigにまとめる。
  - あるディレクトリを基準に各測定項目ごとに違うdirectoryをworkdirに設定。複数のtargetを一つのファイルに記述可能となり、計測プロセスを減少させることが、高速化に貢献する
- RunAsDaemon！
- それでもだめなら “LogFormat: rrdtool” + 14all.cgi！

TIPS – MRTG番外編：ほんとの限界への挑戦

- じゃ、どこまで耐えられるか。
- 耐える基準: loadavg ?, Proc idle ??
  - loadavgはあくまでも起動プロセス数。数が多くてもかまわない。
  - proc idle=0。いそがしいのはいいことだ
- 結局は時間内に処理が終われば良い
- 経験的にいえるのは、loadavgが中期的に拡散方向にいかなければ良いと思う
  - P3 650MHz, 768M Mem、SCSI2ぐらいなら、loadavg=100でも結構耐える

TIPS – MRTG番外編: 敗者の記録

TIPS - Free Software探し方

- ツール・ソフトをどうやって発掘するか
- 一押しは http://www.freashmeat.net/
- Linux plat homeとしているが、他のOSでも稼動するものがそろっている
- ツールの種類ごとの分類がしっかりしており、以下のような検索が可能
  - カテゴリー検索
  - ソフト名称検索
  - 関連検索

TIPS – freashmeat Top Page

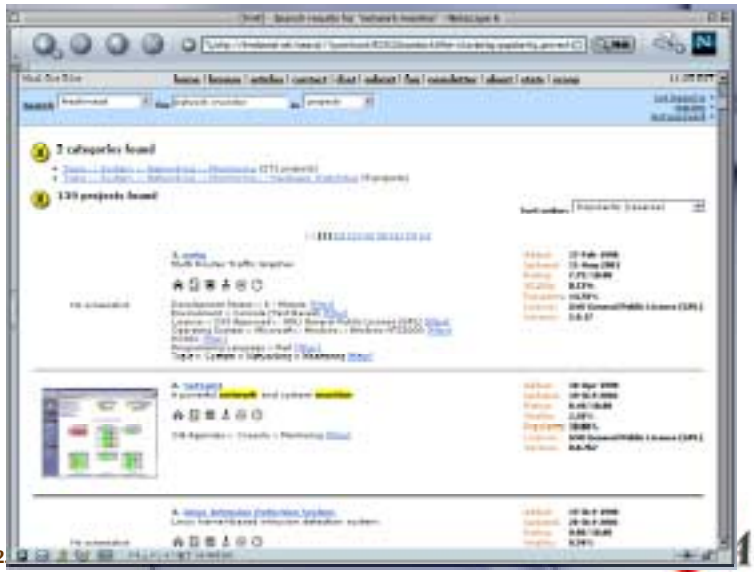
TIPS – 155

freshmeat 私の検索方法 - 例

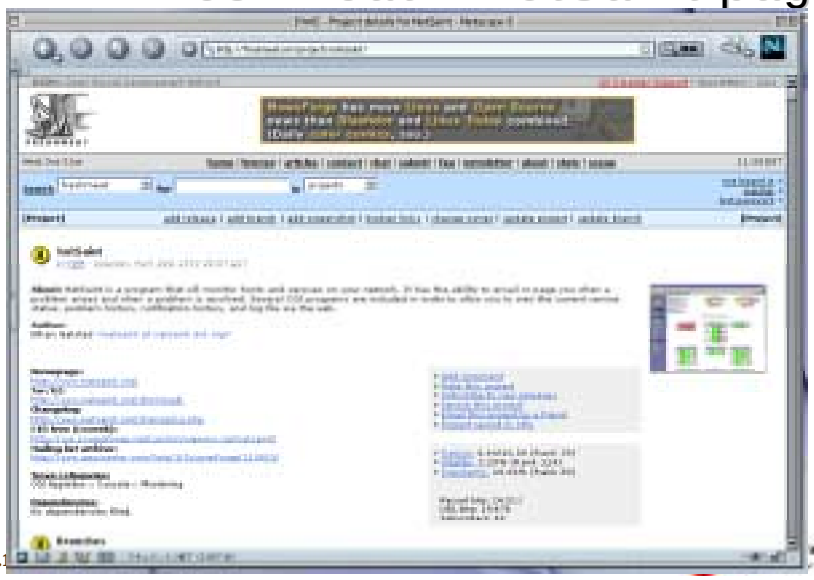
- 課題：ネットワーク監視ツールが欲しい
  - トップで “Network Monitor” をキーにして検索
  - ↓
  - Sort Orderを人気度あいの逆順に指定して検索
  - ↓
  - NetSaintがTopで見つかる。
  - ↓
  - Home Pageを見に行ってみる
  - ↓
  - 使ってみる！

TIPS – 156

freshmeat – NetSaint found！



TIPS – freshmeat – NetSaint page



TIPS – freshmeat ゴミの判別

- 判断基準：なにはともあれ、Popularity（人気度）！
  - 10%以上はメジャーソフト
  - 5%以上はかなり使えるレベル
  - 3%以下は…
    - ちなみに、Big Brother 8.22% !, MRTG 14/93% !!、apache 23.68%!!!
- 更新頻度は重要なファクター
  - まだ開発が続いているものがうれしい
    - バグ対応もしてもらえるし
    - 質問に答えてくれるかもしれない
  - 公式MLのあるツールにしましょう
- まず落として使ってみる。あわせてMLに入って様子見しましょう
  - MRTG, Big BrotherのMLはかなりのthread量
  - 議論のないものはhome pageのヒットも少ない傾向あり

おまけ 今後の期待

NetSaint
DEMARC
RRDTools＋FrontEnds

eAccess

今後の期待 - NetSaint

- http://www.NetSaint.org/
- WEB Baseの監視システム。現在、version 0.0.7
- Web Baseでの管理・変更が可能
- ICMP/TCPベースの監視を行う
  - 監視可能サービス
    - ping, smtp, http, pop3,dns,ftp.telnet,…
- 詳細画面に操作用コントロールパネルがあり、監視停止などの動作が簡単に出来る。→これはBBにはない機能
- Plugin形式をとっており、外部拡張可能
  - MRTG plugin
  - remote server management plugin
  - ...

eAccess

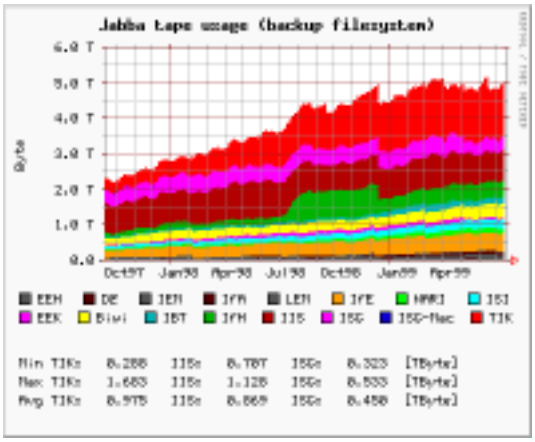
今後の期待 - DEMARC

- http://demarc/.org
- Web interface base network monitor
  - Version 1.05 Release Candidate 2 (2001/10/20現在)
  - *NIXにて稼動(FreeBSD, Linux, OpenBSD, Solaris and NetBSD)
  - Client/Server型構成
- System healthとIDSチェックに使える
  - IDSのチェックにはMySQLベースのSNORT使用可能
- Webベースで構成管理が可能
- User毎にアクセス管理することができ、アクセス情報を限定することが可能
- 以下のソフトが必要
  - Snort version 1.8 or higher
  - MySQL 3.23 database server
  - Perl with the following Perl modules: CGI, DBI, DBD::MySQL, and Digest::MD5

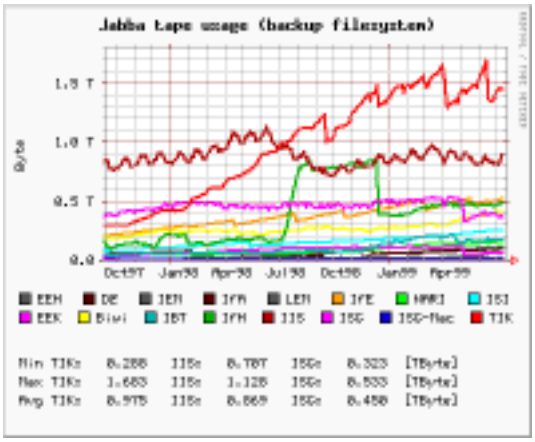
今後の期待 – RRDTools+Frontends

- http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
- RRDTools :Round Robin Database Tools
- MRTGの作者Tobi OetikerによるMRTGの後継プロジェクト
- MRTGのログサイズが変わらないという利点を受け継ぎつつ、より柔軟に、より高速に、より多彩な表現ができるように、をコンセプトに開発
- データベース管理、グラフ作成に特化
  - RRDToolsだけではMRTGのようなWEB画面はできない
  - FrontEnd Programが必要
    - MRTG/Big BrotherもFrontEndのひとつとなる
    - Remstat, ORCA, Cricket, NRG, ...
  - やはりひとつの画面で複数の関連情報を集約確認するにはこれしかない。うまく取り込めれば非常に強いツールとなる

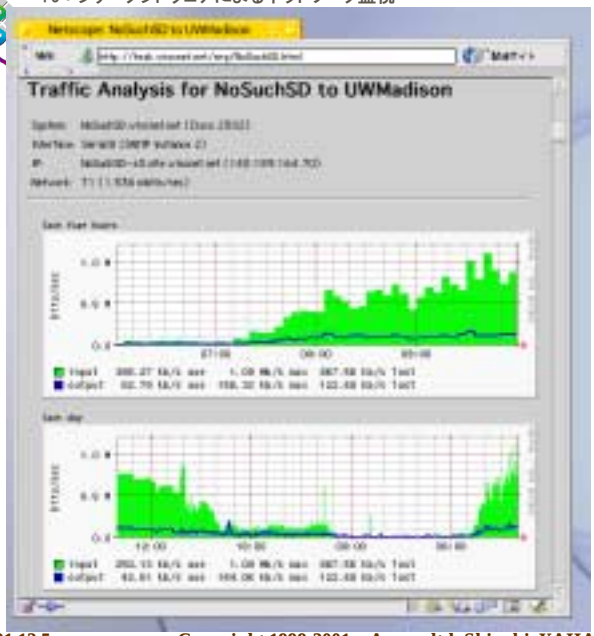
RRDTools - 例1



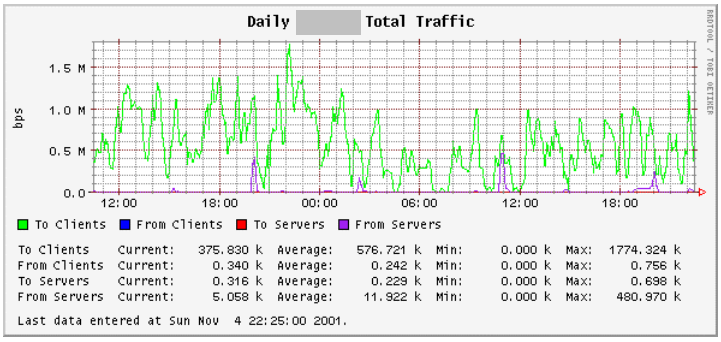
RRDTools - 例2



RRDTools - 例3：NRG



RRDTools - 例4：ORCA



RRDTools - 例5：Big Brother LARRD

参考資料:文献/URL

参考:書籍1

- UNIX MAGAZINE
  - 連載「Unix Communication Notes」山口 英 1998.3～
  - 「倉敷芸術科学大学のネットワーク構築」小林和真 1997.12
- OPEN DESIGN No.10
  - 「ネットワーク管理技術のすべて」
- Software Design 1999.9
  - 「フリーソフトウェアでネットワークをチェック trafshow, MRTG, ntopの導入」 田村吉章
- Software Design 2000.7-10
  - 「FreeBSDサイト管理風雲録 – SNMPの話 その1－4」 工藤智行

参考：書籍2

- SNMP
  - "Snmp, Snmpv2, Snmpv3, and Rmon 1 and 2" -- William Stallings; 3rd edition (January 1999) Addison-Wesley Pub Co; ISBN: 0201485346 ;
  - "Practical Guide to SNMPv3 and Network Management, A" -- David Zeltserman, Dave Zeltserman; (May 4, 1999) Prentice Hall; ISBN: 0130214531
  - 「SNMPバイブル - インターネット管理への実践ガイド -」William Stallings著、大鐘久生、Addison-Wesley Pub Co; ISBN-7952-9651-0

参考：書籍3

- Security
  - 「不法侵入の検出と対策 ネットワーク侵入検知」武田圭史/磯崎宏 著, SOFTBANK Publishing, ISBN4-7973-1253-X
  - 「クラッカー迎撃完全ガイド」Anonymous著/トップスタジオ訳, インプレスコミュニケーションズ, ISBN4-8443-1360-6

参考: 文献1

- "Yet Another network command/tool/system"
  - 向阪 正彦 ファストネット株式会社
  - http://www.janog.gr.jp/meeting/janog6/pdf/command/janog6_kosaka.pdf
  - JANOG6 in 下丸子 2000/6/16
- "Building Network Monitoring Systems with RRDTool"
  - Tobias Oetiker, CAIDA
  - http://www.nanog.org/mtg-9910/tobi.html
  - NANOG17 in Montreal 1999/10/4
- "Using Remstats for Network and Server Monitoring"
  - Thomas Erskine, Communications Research Center
  - http://www.nanog.org/mtg-9910/erskine.html
  - NANOG17 in Montreal 1999/10/4

参考: 技術関連サイト

- ミッキーのネットワーク研究所
  - Snort/iplogなどをやさしく(^^; 解説。
  - http://www.hawkeye.ac/micky/
- Network Security Portal
  - http://www.whitehats.com/index.shtml

参考:性能評価

- Communication Traffic Project
  - http://www.mmlab.tnl.ntt.co.jp/
- Distributed Benchmark System
  - http://www.kusa.ac.jp/~yukio-m/papers/

Network Management

- http://wwwsnmp.cs.utwente.nl/Docs/software/
- http://netman.cit.buffalo.edu/index.html
- http://www.nemoto.ecei.tohoku.ac.jp/~nitou/snmpdocs/tutorial1.html

ツールURL集1

- Big Brother
  - http://bb4.com/
  - Extensions Archive: http://www.deadcat.net/
  - Larrd: http://larrd.packetpushers.com/
- DEMARC
  - http://demarc.org/
- Expect
  - http://expect.nist.gov/
- fping
  - http://www.fping.com/
- IPTraf
  - http://cebu.mozcom.com/riker/iptraf/index.html
- logsurfer
  - http://www.cert.dfn.de/eng/logsurf/
- logtrend
  - http://www.logtrend.org/english/

ツールURL集2

- MRTG
  - http://www.mrtg.org/
  - http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/
- mon
  - http://www.kernel.org/software/mon
- NeTraMet
  - http://www.auckland.ac.nz/net/Accounting/ntm.Release.note.html
- MTR
  - http://www.bitwizard.nl/mtr/
- nPULSE Network monitor
  - http://www.horsburgh.com/h_npulse.html
- netplot
  - http://netplot.sourceforge.net/
- NetSaint
  - http://www.netsaint.org/

ツールURL集3

- Net-SNMP (UCD-SNMP)
  - http://net-snmp.sourceforge.net/
- NISCA
  - http://freshmeat.net/redir/nisca/12683/url_homepage/
- Ntop
  - http://www.ntop.org/
- Pong
  - http://poe.eekeek.org/?Poing
- RRDTool
  - http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
  - RRDTool Frontend - CRICKET
    - http://cricket.sourceforge.net/
  - RRDTool Frontend - NRG
    - http://nrg.hep.wisc.edu/
  - RRDTool Frontend – Ntop
    - http://www.ntop.org/RRD/

ツールURL集4

- RRDTool - continued
  - RRDTool Frontend - ORCA
    - http://www.orcaware.com/
  - RRDTool Frontend - Remstats
    - : http://remstats.sourceforge.net/release/index.html
- Scotty
  - http://wwwhome.cs.utwente.nl/~schoenw/scotty/
- seafelt
  - http://seafelt.unicity.com.au/
- shepherd
  - http://atrey.karlin.mff.cuni.cz/~clock/twibright/shepherd/
- sing
  - http://sing.sourceforge.net/
- SNIPS aka NOCOL/multiping
  - http://www.netplex-tech.com/software/snips/
- Snort
  - http://www.snort.org

ツールURL集5

- SPONG
  - http://spong.sourceforge.net/
- ssh
  - http://www.ssh.com/about/company/index.html
- statscout
  - http://www.statscout.com
- SWATCH
  - http://www.oit.ucsb.edu/~eta/swatch/ syslog-ng
  - http://www.balabit.hu/products/syslog-ng/
- Treno
  - http://www.psc.edu/~pscnoc/treno.html
  - Experimental TCP Implementations
    http://www.psc.edu/networking/tcp.html
- Tripwire
  - http://www.tripwire.com
- visualroute
  - http://www.visualroute.com

参考：URL集1

- General network management portal
  http://netman.cit.buffalo.edu/index.html
- “The Simple Times”
  http://www.simple-times.org/
- SNMP FAQ
  http://www.cis.ohio-state.edu/hypertext/faq/usenet/snmp-faq/part1/faq.html

参考:URL集2

- Sample Cisco device security configs
  http://www.cisco.com/warp/public/700/tech_configs.html#SECURITY
- Cisco device SNMP configuration tips
  http://www.cisco.com/warp/public/490/index.shtml

参考:Free software link

- Fresh Meat – Free Software Index
  - http://www.freshmeat.net/
- SOURCE FORGE
  - http://sourceforge.net/
- Solaris Freeware Project
  - http://sunsite.sut.ac.jp/sun/solbin/
- CPAN – Comprehensive Perl Archive Network
  - http://www.perl.com/CPAN/

参考：looking glass & traceroute site

- Looking glass
  - http://neptune.dti.ad.jp/
  - http://nitrous.digex.net/
- Traceroute site
  - http://www.traceroute.org/
  - http://www.geektools.com/traceroute.php

参考：組織

- IETF http://www.ietf.org/
- NANOG http://www.nanog.org/
- JANOG http://www.janog.gr.jp/
- CAIDA http://www.caida.org/tools/
  - cflowd ,RRD …etc
- LBNL's Network Research Group
  - http://ee.lbl.gov/
    - tcpdump,libpcap,arpwatch,traceroute,pathchar