NETARC

JPNIC・JPCERT/CC SecuritySeminar 2004

＜不正侵入の実態と具体的対策＞
ログ管理・解析(UNIX系)

株式会社ネットアーク　宮川雄一

2005年2月3日

はじめに

- 本セミナでは
  - なぜログ管理が重要か
  - どのようにログ取得設計をするべきか
  - 実際に使用するツールの種類と概要

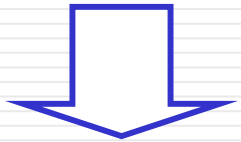
  を解説します。

NETARC

なぜログの管理が重要か

セキュリティ事故発生

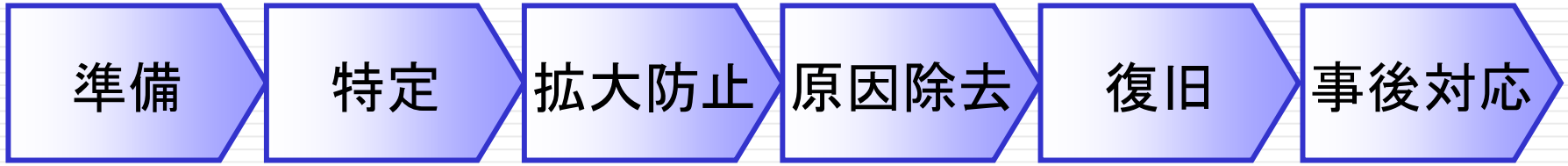
- 何かネットワークがおかしい。
- 重要な情報が盗まれていないか?
- この情報は改ざんされてないか？
- 対応しようにもシステムは止められない。



今何が起きている？？
いったいあれは何だったんだ？？

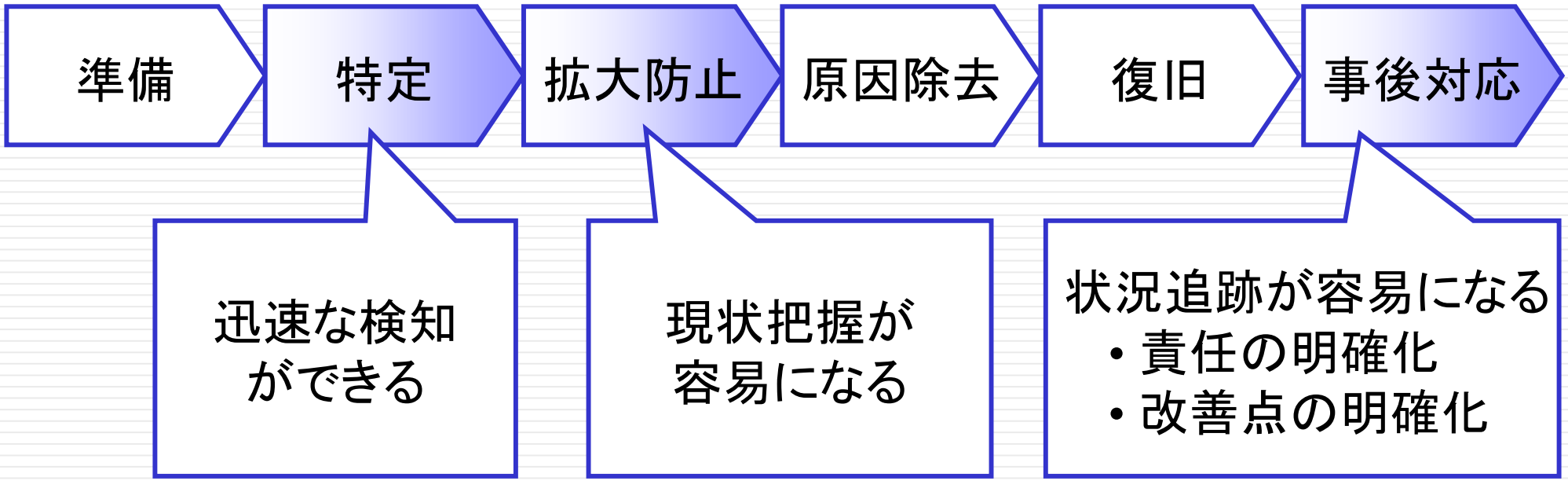
事故対応を整理すると・・・

- 対応の例：
  SANSのインシデント対応手順を参照



- 特定からから復旧までは緊張状態が続く
- 一刻も早く収束させたい

適切なログ管理をすると、対応に役立つ



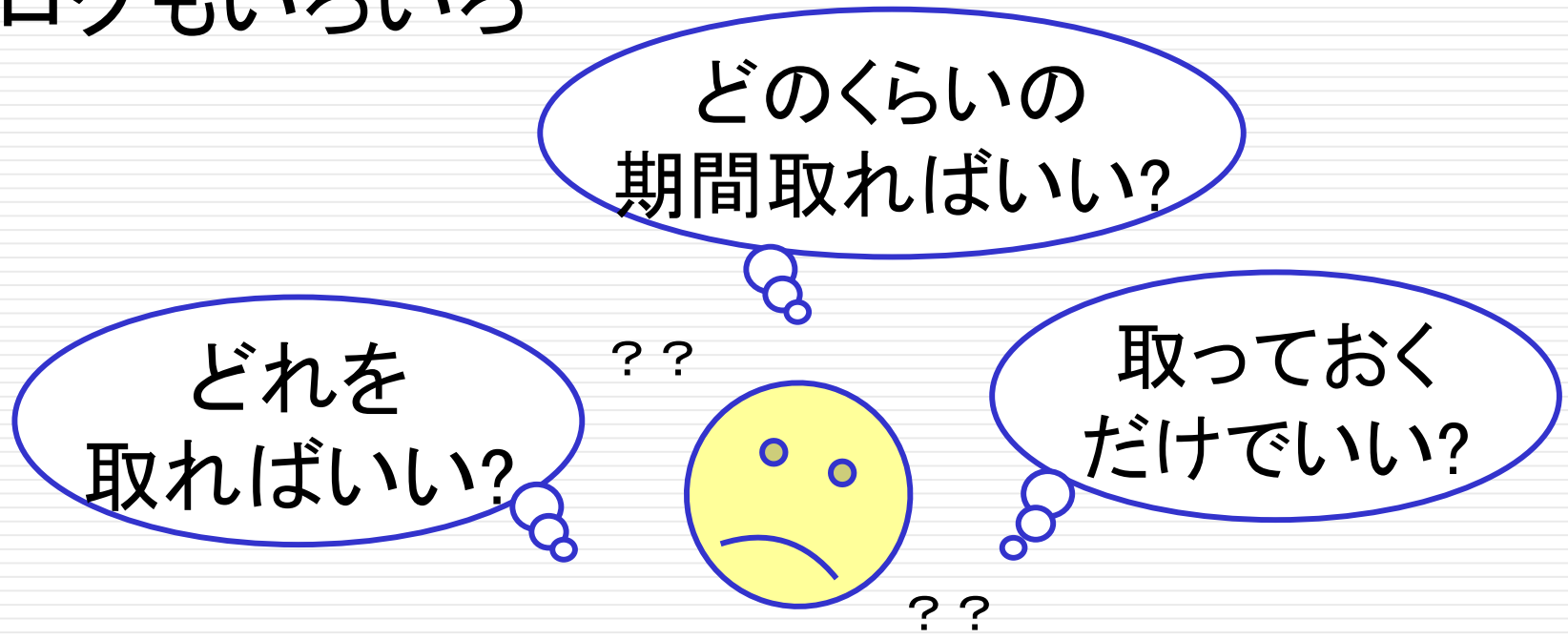
だからログ管理が重要

NETARC

ログ管理の考え方

どこから考える?

・ ログもいろいろ



時間も資源も有限。
なんでも取ればいいと言うものでもない

優先順位を考える

ログ管理は「リスク評価」から「運用」に落とすべき

- それは会社にとってどれだけ重要?
- それは現場にとってどれだけ重要?

提供サービスによるリスクの大きさを評価

↓

リスクが大きいものをピックアップ

↓

運用上、現実的な手段を検討、対応

現実的なログ管理を検討するための4項目

- リスク評価
- 即時性
- 広域性
- 外部要因

リスク評価って何？

例

- いろいろなサービスを運用してます
- いろいろな攻撃や2次被害が考えられます
- ・・・・・・

何をどうやって評価する?

確立された理論による「リスク評価」

- リスク＝脅威の発生頻度×被害の大きさ
  - コートニー理論

- リスク＝資産価値×脅威×脆弱性
  - （GMITS: Guidelines for the Management for IT Security）

参考として組み合わせて考える

リスク評価手順例：全体像の把握

リスク評価手順	例
現在の提供サービスを挙げる	http://www.netarc.jp/サイト、netarc.jpのMTA、等
⇩	
各サービスで発生しうる問題を挙げる	サイト改ざん、サイトスローダウン、停止、等
⇩	
各問題が起きたときのダメージの大きさを考える	改ざん＞停止＞スローダウン

ダメージの大きいものが重要な問題

リスク評価手順例：具体化

重要な問題を
リストアップ

その原因と
発生頻度を具体化

例：

改ざん

サイト停止（閲覧不能）

スローダウン

⋮

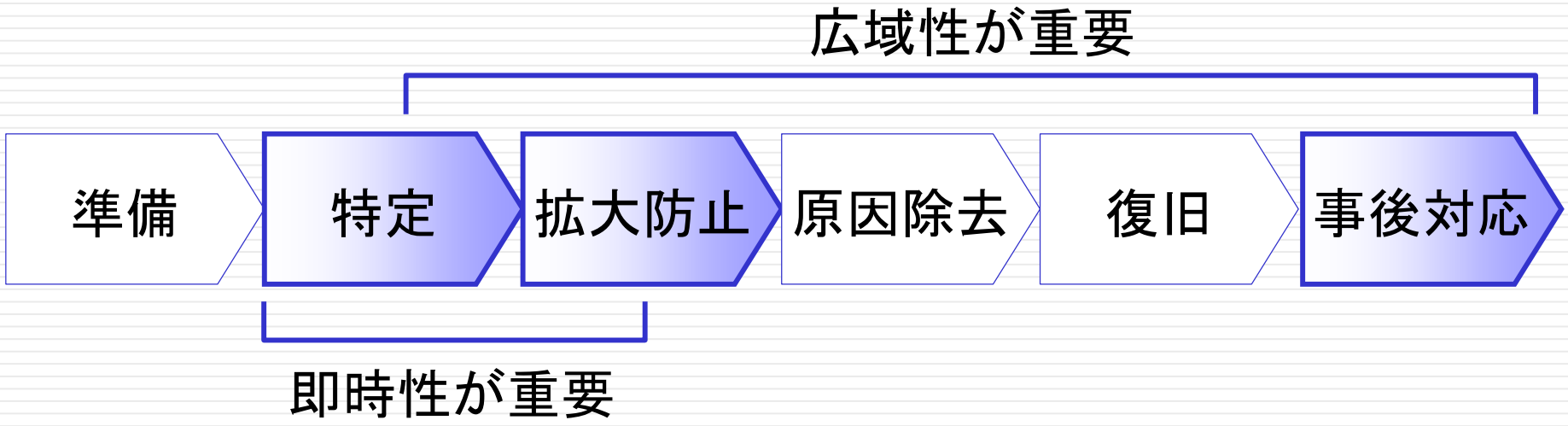
- ネットワーク障害　少
- 機器障害　少
- プロセス障害　中

⋮

具体化の結果、リスクの大きさ評価が可能となる

実際の対応を考える

・「即時性」と「広域性」



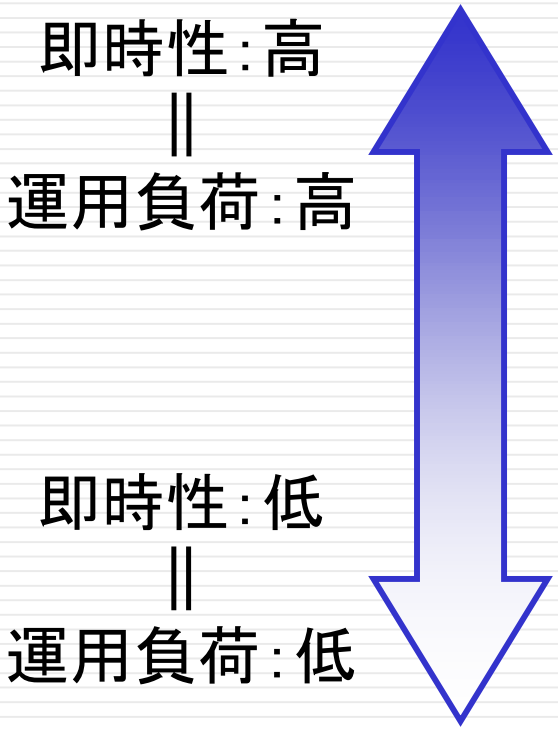
被害を最小限に抑える～即時性

- 「何か起きたら即対応」

- 異常状態の継続＝損害拡大
- 次の段階に進んだらもう遅い事も
  - 初期で対応しないと手遅れ(ワーム等)
  - 証拠を消されてしまう(侵入者等)

⋮

即時性の種類はさまざま

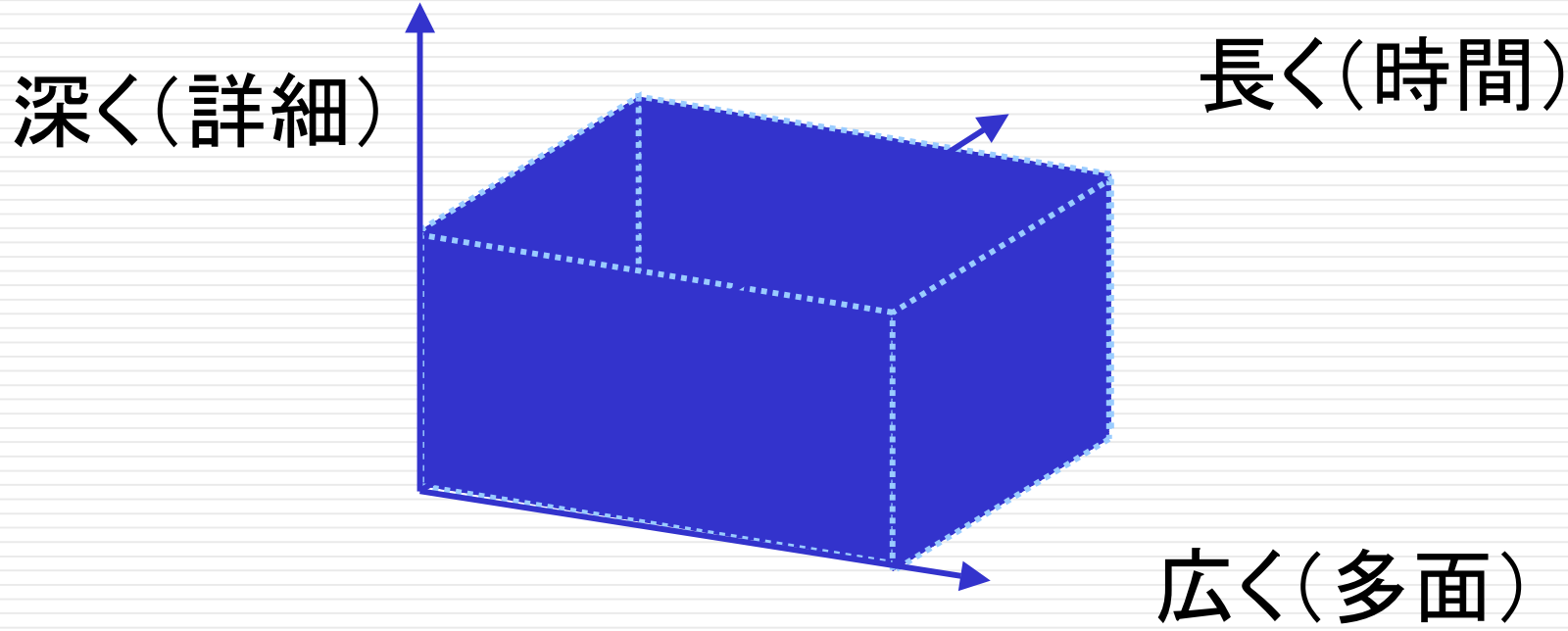


検知間隔	通知方法
・ リアルタイム？	・ 電話連絡？
・ 5分おき？	・ メール連絡？
・ 毎時チェック？	
・・・	

「即時性」と「運用負荷」とのトレードオフを考える

状況を把握し、改善に繋げる～広域性

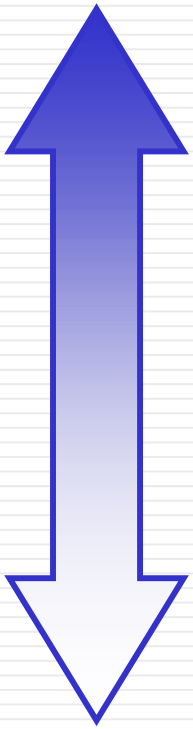
- いつでもどこでもデバッグログ?
- 情報が少ないと、状況把握は難しくなる
- どこまでログを取ると効果的か？



広域性の種類もさまざま

広域性：高
‖
効率：低

広域性：低
‖
効率：高



広さ	深さ	長さ
・サーバ+ネット+クライアント ・サーバ＋ネット ・サーバ	・Debug？ ・Verbose？ ・Warn？ ・Info？	・7年？ ・90日 ・1ヶ月 ・1週間 ・今日分だけ

「広域性」と「効率」とのトレードオフを考える

外部要因～時には、別の視点からも

- 経営者はどう考える？ ※
  - 対応（低減）
  - 回避
  - 保有
  - 移転（保険等）

⇨ このうち、対応（低減）、保有が現場に関わる

※ ISMS認証基準（Ver2.0）リスク対応についての選択肢より

外部要因～ISMSより※

- 9.（7） システムアクセス及びシステム使用状況の監視
  目的：許可されていない活動を検出するため

  9.（7）① 事象の記録
  9.（7）② システム使用状況の監視
  9.（7）③ コンピュータ内の時計の同期

ISMSやPマーク等の組織内ポリシーにより取得が義務付けられているログは取る必要がある

※ http://www.isms.jipdec.jp/doc/JIP-ISMS100-20.pdf

検討結果を整理

例

順位	サービス	問題	原因	発生頻度	即時性	広域性			外部要因
						広	深	長	
1	http://www.netarc.jp/	Webサイト停止	ネット障害	少	要	要			ISMS要請
2			プロセス障害	中	要		要		
3			侵入	少	要	要	要	要	
4			ディスク不足	少					

- 上位の事象に対応する為のログは取得必須
  - 必要に応じて
    - リアルタイム検知
    - デバッグログ取得
    - 長期間保存　　等を行う

NETARC

ログ管理　実現方法

ログ取得ツールの分類

- ログを取るツール
- ログを保存、管理するツール
- ログから情報を抜き出し、通知するツール
- 時刻同期のツール

サーバ上でログを取る

- OS
  - syslog(-ng)、msyslog
    - コマンドラインツール(logger)
  - ログイン履歴(wtmp)
  - プロセスアカウンティング(acct、pacct)
  - コマンドヒストリ
  - ホストファイアウォールのログ

- サーバから直接ログ出力
  - Apache、Bind、OpenSSH、ProFTPD・・・
  - 各種DB

サーバ上でログを取る

- 独自アプリケーション、スクリプトの場合
  - 特定キーワードで出力させると良い
    - 例
      - [INFO]:xxxx
      - [WARNING]:xxxx
      - [ERROR]:xxxx
  - 時間で出力ファイル名を分けると良い
    - 例 日付で分ける
      - prefix-20050203.log
      - prefix-20050204.log
  - syslogを使うことも検討すべき

バイナリログ

- バイナリフォーマットのログを用いるものもある
  - テキストに比べて不正に改変されにくい
  - 独自ツールで読むか、テキスト変換
  - 例
    - 各種DB
    - wtmp

    等

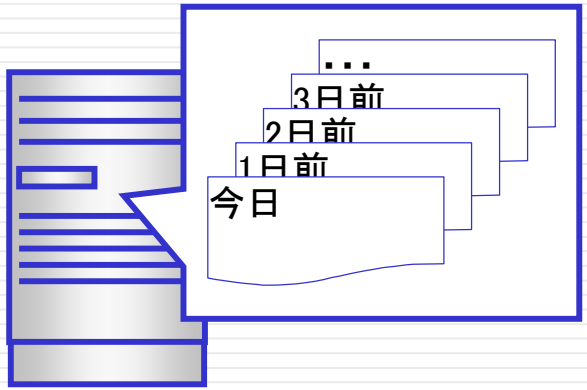
ネットワーク上でログを取る （参考）

- ファイアウォール型
  - IP Filter、iptables、IPFW、PF 等
  - 各種商用Firewall
- スニッファ型
  - tcpdump、ngrep、ethereal、snort 等
  - 各種商用IDS、モニタリングソフト

サーバログに比べてノイズが多いという欠点がある
ただし、攻撃者には発見されにくい

ログローテーションと保存

- ログを日時毎の別ファイルにして管理する
  - 日時をファイル名の一部にして容易に検索
  - ログサイズを小さくして管理
- ツール
  - newsyslog
    - http://www.weird.com/~woods/projects/newsyslog.html
  - logrotate
    - Redhat
  - cron＋findコマンド

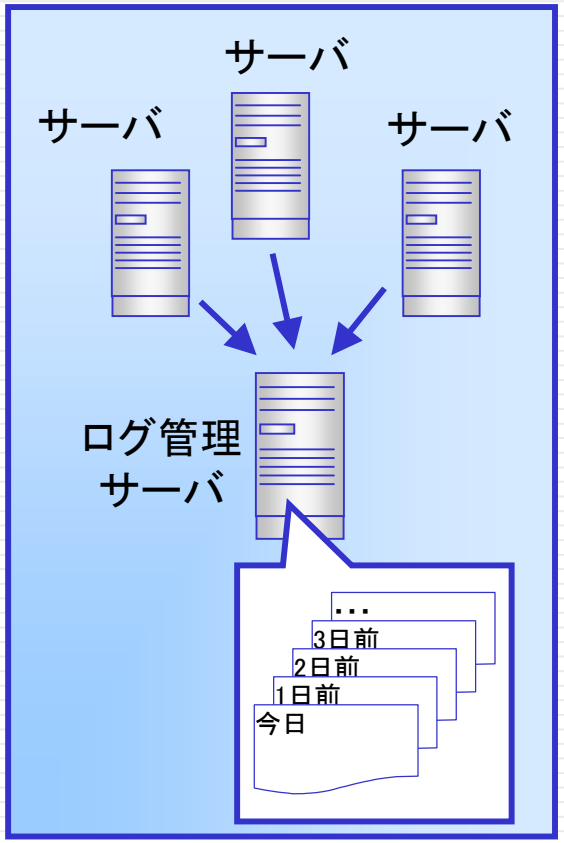


サーバ

集中管理

- ログを単一のサーバで集中管理する
  - 管理が容易になる
- ツール
  - syslog(-ng)
  - scp、ssh等＋スクリプト
  - NFS等ファイルサーバ
- 課題
  - ログ転送性能
  - ログの量
  - 転送リンクの信頼性

⇨ ログ専用セグメントの作成を検討すべき。
その際そこが脆弱にならないように注意



ログから情報を抜き出し、通知する

- 大量のログから意味のあるものを抜き出して、通知
  - 管理作業の効率化
  - 「慣れ」の防止

- リアルタイム
  - logsurfer
    - (http://www.cert.dfn.de/eng/logsurf/)
  - swatch
    - (http://swatch.sourceforge.net/)

- 定期的
  - logcheck
    - (http://sourceforge.net/projects/sentrytools/)
  - logwatch
    - (http://www.logwatch.org/)

情報の抜き出し方

	既知のエラー検出	異常検出
方法	• 問題発生時のログ文字列を検知してアクション（メール送付等）	• 未知のログ文字列を検知してアクション（メール送付等）
長所	• 誤検知が少なく効率的	• 設定者が知らないエラーも検知できる
短所	• 設定者の知識にないエラーは見落とす可能性がある	• 当初は全てが検知対象 • 継続して設定追加していく

状況に応じて使い分ける事が重要

SIM（Security Information Management）ツール

各種セキュリティ関連情報等を一元管理、相関解析等をする

- netForensics(商用)
  - http://www.netforensics.com/
- ArcSight（商用）
  - http://www.arcsight.com/
- FLAG
  - http://pyflag.sourceforge.net/
- OSSIM
  - http://www.ossim.net/

時刻同期

ログは時刻が合っていることが前提
時刻を鍵にデバイスの相関確認や解析をする

- ntpd
  - Daemonとしてほぼリアルタイムに同期可能
- rdate もしくは ntpdate＋cron
  - ポートを開かずに同期可能

既存ツールが不便だったら

- 監視してログを吐くスクリプト書きましょう
  - サービス監視
  - プロセス監視
  - ディスク容量監視
  - トラフィック監視
  - ・・・

ディスク閾値監視スクリプト例 (On Redhat9)

```
# /bin/sh

DF_THRESHOLD1=80
DF_THRESHOLD2=90
EXCEPT=/mnt/cdrom

for OVER in `df -k | grep ^/ | grep -v $EXCEPT | ¥
  awk -vdft=$DF_THRESHOLD1 '{split($5,df,"%");if (df[1] > dft) print $6}'`
do
        logger -p local4.debug "INFO $OVER is over $DF_THRESHOLD1 %"
done

for OVER in `df -k | grep ^/ | grep -v $EXCEPT | ¥
  awk -vdft=$DF_THRESHOLD2 '{split($5,df,"%");if (df[1] > dft) print $6}'`
do
        logger -p local4.debug "WARNING $OVER is over $DF_THRESHOLD2 %"
done
```

まとめ

まずは始めてみましょう

ログ管理はもう十分、それでも判らない事がある。
ログで追えなくなったら?

次セッション「不正侵入の発見」に続く

参考文献

・ システム監査 情報セキュリティ監査ハンドブック	日本システム監査人協会 日本セキュリティ監査協会
・ セキュリティマネジメント戦略	監査法人トーマツ
・ ISMS認証基準 (v2.0)	日本情報処理開発協会
・ ITセキュリティマネジメントのガイドライン(TR X0036)	日本工業標準
・ セキュリティウォリア	Cyrus Peikari、 Anton Chuvakin
・ ネットワークセキュリティ Expert(Software Design)	技術評論社

NETARC

＜不正侵入の実態と具体的対策＞
ログ管理・解析(UNIX系)

ご清聴ありがとうございました

株式会社ネットアーク
宮川 雄一
u@netarc.jp