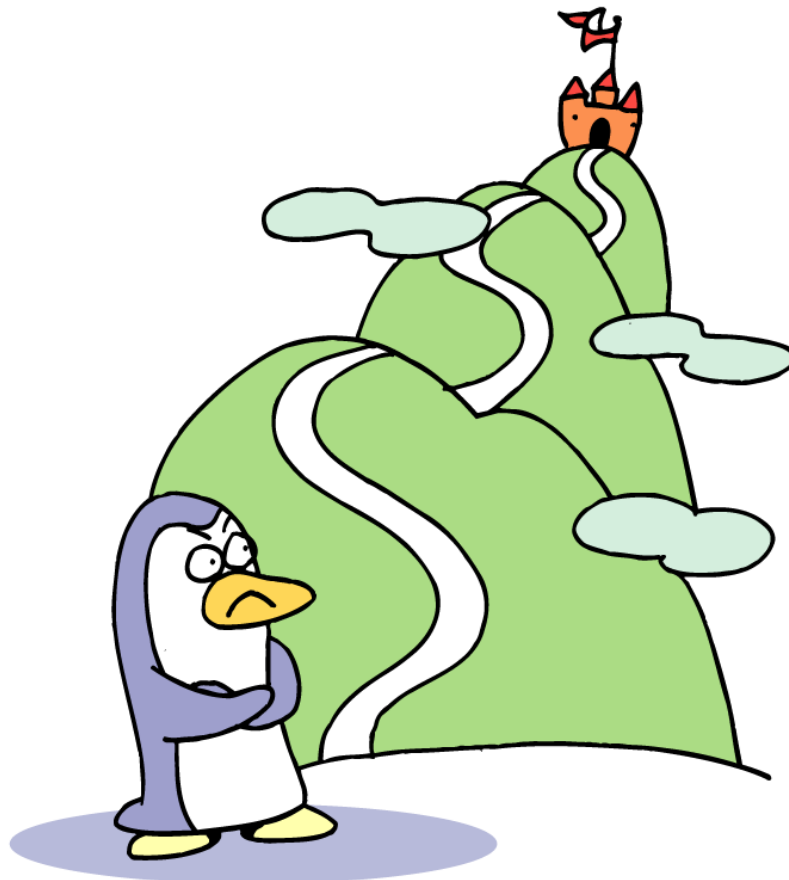


TOMOYO Linuxへの道

“使いこなせて安全なLinuxを目指して”



2005.11.11
株式会社NTTデータ
原田季栄



はじめに

TOMOYO Linuxのプロジェクトは2003年3月に発足しました。カーネルのコンパイル経験もない状態から、文字通り手探りで開発してたどりついた結果が、ポリシーの自動学習機能を持つセキュリティ強化Linux, “TOMOYO Linux”です。

TOMOYO Linuxについては、これまで5本の論文を発表し、またビジネスショウやセキュリティ・スタジアム等で概念の説明やデモを行っています。この資料では、いかにしてTOMOYO Linuxにたどりついたか、そこに至るまでの道筋と私たち開発メンバーが考えたことを伝えようと思いまとめてみました。TOMOYO Linuxはまもなくオープンソースとして公開される予定ですが、ソースコードや論文と合わせて参照いただくことにより、より深く私たちの経験を(間違いも含めて)共有してもらえることを希望しています。

歩んだ道のりは曲がりくねっていても変わらないことがひとつだけありました。それは「使いこなせて安全なLinuxを目指す」ということです。TOMOYO Linuxがその選択肢のひとつとなり、私たちの試みが他の方々の参考になることを願ってやみません。

平成17年11月11日

プロジェクト発足



Stage 1

OSの脆弱性について調査せよ!

2003.4~2003.5



脆弱性について学ぶ

- Linuxにはどのような脆弱性が存在するか
- exploitプログラムと情報の収集
 - 社内のURLフィルタリングにひっかかりまくる
- 環境の構築と実験
 - ローカル権限昇格
 - samba, ftpサーバからのシェルコード
 - シェルコードの自作にも挑戦
 - 「exploitできる環境」の維持に苦労 (ちょっとしたはずみでexploitできなくなってしまうw) →以来VMwareの愛用者に・・・

脆弱性について学ぶ

- セキュリティ強化OS、高信頼OS、ツールの調査
 - “secure OS”, “trusted OS”等で検索、ヒットしたサイトを片端から・・・
- SELinux, LIDS, Immunix, RSBAC等を重点に
- SELinuxは実際にインストールしながら
 - 当時はカーネルへのパッチ
 - configにつまづき苦労
 - 概念が難解でなかなか使える気がしなかった

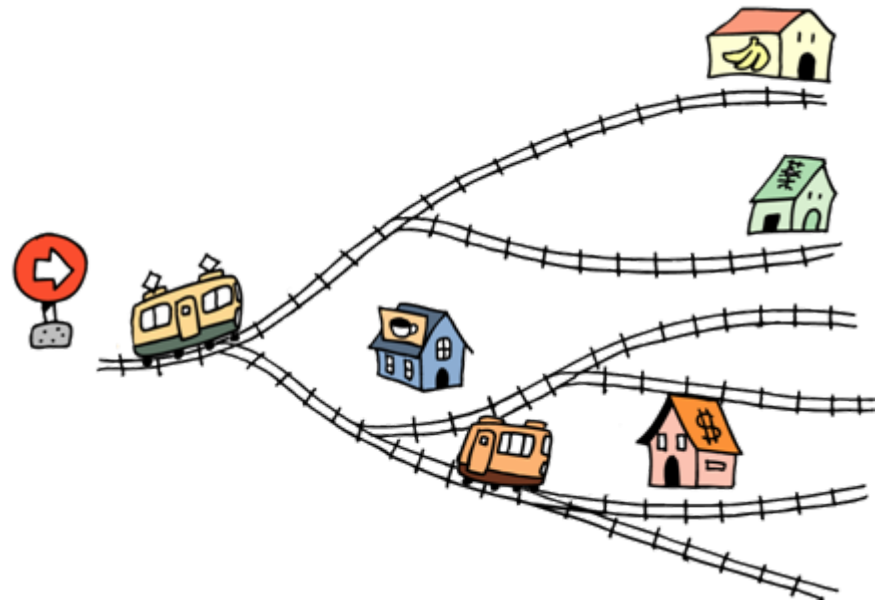
強制アクセス制御の概念の理解



強制アクセス制御の概念の理解



導入前

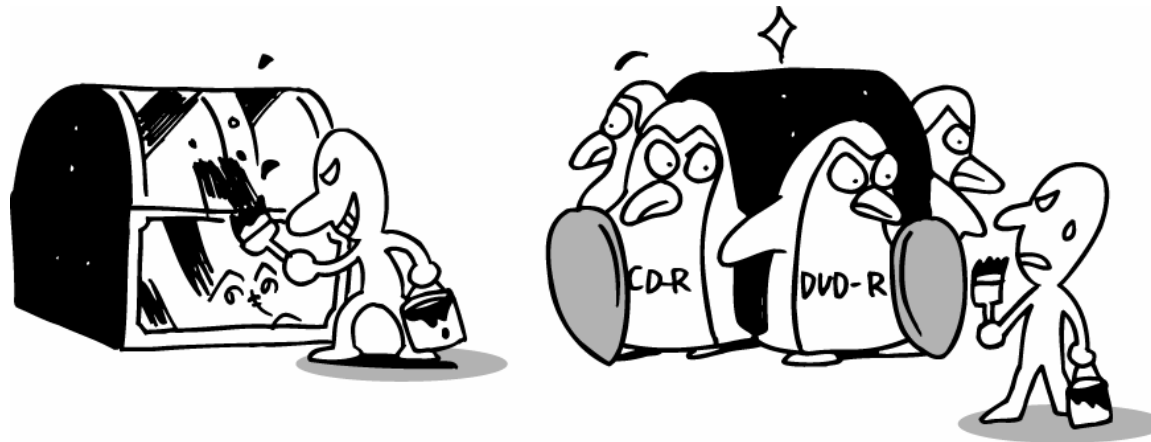


導入後

Stage 2

ファイルの改ざんを防止せよ!

2003.5~2003.10



隔離環境による改ざん防止

- chrootやjailによる隔離に注目
- 全てのプログラムをchroot環境で動作させることはできないか？
 - chroot環境で必要なファイルを把握するのが大変だ
 - プログラムは他のプログラムを呼び出すため、chroot環境から別のchroot環境のプログラムを呼び出せるようにカーネルを修正しなければならない
 - 実現を断念 orz



どのように改ざんを防止するか？

- 論理的な改ざん防止（読み込み専用モードでのマウント）では、読み書きモードで再マウントされたり、デバイスファイル経由で改ざんされる可能性がある。
- 物理的に改ざんできないようにしてしまえば、ポリシーなしで強固な改ざん防止が実現できないか？

読み込み専用化への壁(1)

- 物理的に読み込み専用にする前に、論理的に読み込み専用にはできなければいけない。
 - / を読み込み専用にすることを指す。
- / を丸ごと読み込み専用にしてログインができなくなる。
 - 書き込みが必要なファイルを見つけなければいけない。



読み込み専用化への壁(1)

- カーネル修正による書き込みアクセスの追跡実験
 - 書き込み失敗 (EROFSエラー) が発生した場合に、そのファイル名を表示するように修正した。
 - 読み書き可能でなければならないファイルを検出できるようになり、読み込み専用でマウントされるファイルシステム中に含めてはいけないファイルを容易に知ることができるようになった。

読み込み専用化への壁 (2)

- /dev が読み込み専用ではいけない
 - コンソールからログインができなくなる。
- /dev に tmpfs をマウントした瞬間にフリーズ
 - 「/devが見えない状態」を作ってはいけない。
- /dev に devfs をマウントして対処した。
 - tmpfs を適当な場所にマウントしてデバイスファイルを作成した上で、`mount --bind` により /dev へ反映するという方法でも可能。



SAKURA Linux 誕生

- ついにroot fsを読み込み専用で動作させることに成功
 - もちろん、物理的にも読み込み専用
- マウント制御機能を追加
 - root fsが読み込み専用でも自由にtmpfsをマウントできては意味がないため必須
 - カーネルのマウント処理の内部で引数をチェックし、ポリシーで許可されていないものを全て拒否する



task_structの拡張実験

- `execve()` の実行を放棄する権限を追加
 - 必要なくなった時点でこの権限を放棄することで、シェルコード対策ができる。
- 「自発的権限放棄」という概念
 - プロセスが必要としない権限をプロセス自らが放棄することを徹底できれば、ポリシーを必要としない「強制アクセス制御」が可能となる。



task_structの拡張実験

- chroot済みか否かを記憶する変数を追加
 - chrootしたプロセスがアクセスしたファイルだけを追跡できるようにした。
 - chroot環境で必要なファイルを簡単に知ることが可能になった。
 - chroot環境構築が簡単になり、ApacheとTomcat用のchroot環境の自動構築に成功

2003.10 LC2003

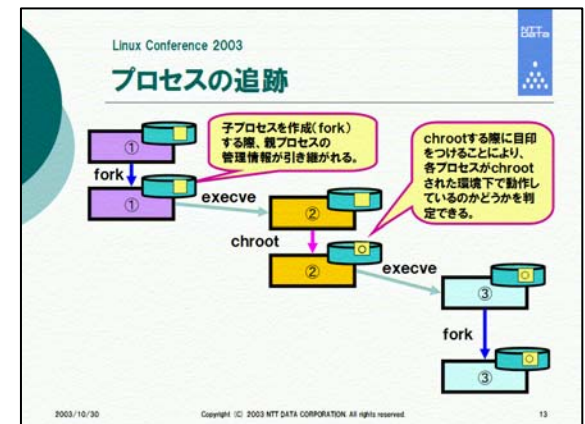
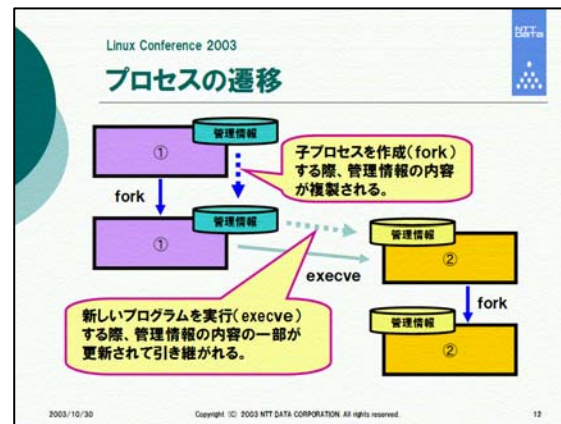
- Linux Conference 2003で、「読み込み専用マウントによる改ざん防止Linuxサーバの構築」として論文を発表
 - 物理的な改ざん防止
 - マウントの制限
 - chroot環境の自動構築
 - 自発的な権限放棄
- これらの機能は現在もTOMOYOに引き継がれている



Linux
CONFERENCE
Linux Conference 2003

「読み込み専用マウントによる
改ざん防止Linuxサーバの構築」

平成15年10月30日
株式会社NTTデータ
技術開発本部
原田季栄 haradats@nttdata.co.jp



Stage 3

そのアクセスを記録せよ!

2003.7~2003.10





アクセス制御の難しさ

- 「物理的改ざん防止」は最強だと思いつつも、情報漏洩や参照には無力なので、なんとかしたいと考えた
- そうすると、ポリシーの定義と管理は不可欠
- でも、SELinuxは難しすぎる
- SELinux用のポリシー作成の手助けはできないだろうか？
- カーネルの構造を調べるところから開始



アクセス制御の難しさ

- カーネル修正によるアクセス制御の実験
 - 特定ファイル（特定iノード番号を持つファイル）の実行を禁止するようexecve()を直接修正
 - しかし、対象のファイルをコピーして実行されたら意味がない（iノードが変わってしまうため）
 - そもそも実行を禁止するだけでは不十分、読み書きも制御できなければ意味が無い

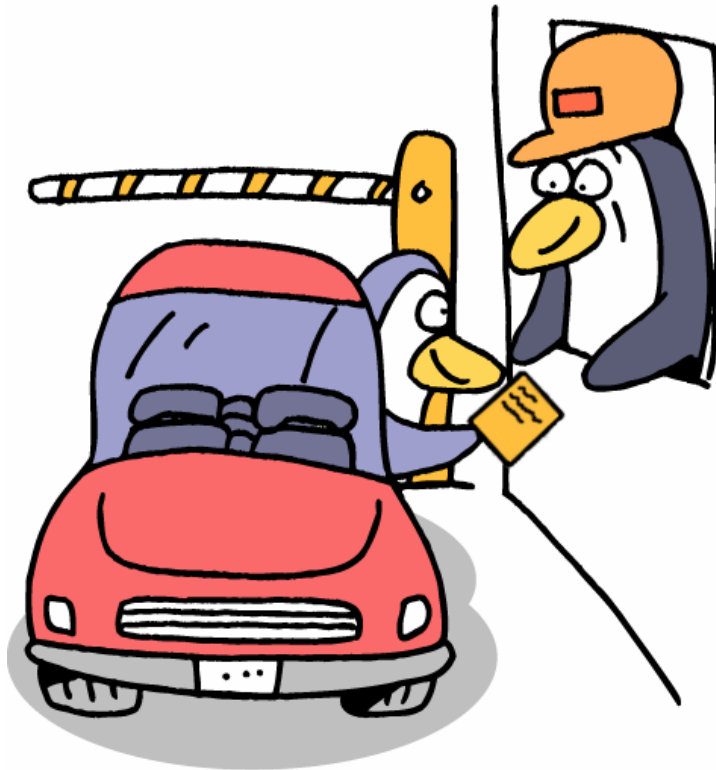
アクセスの追跡と記録

- task_structの拡張実験
 - プログラムの実行履歴を追加
 - fork () /execve () の仕組みを活用
 - プログラム毎のファイルアクセスが追跡可能に



アクセスの追跡と記録

- プロセス起動履歴に基づくアクセス記録



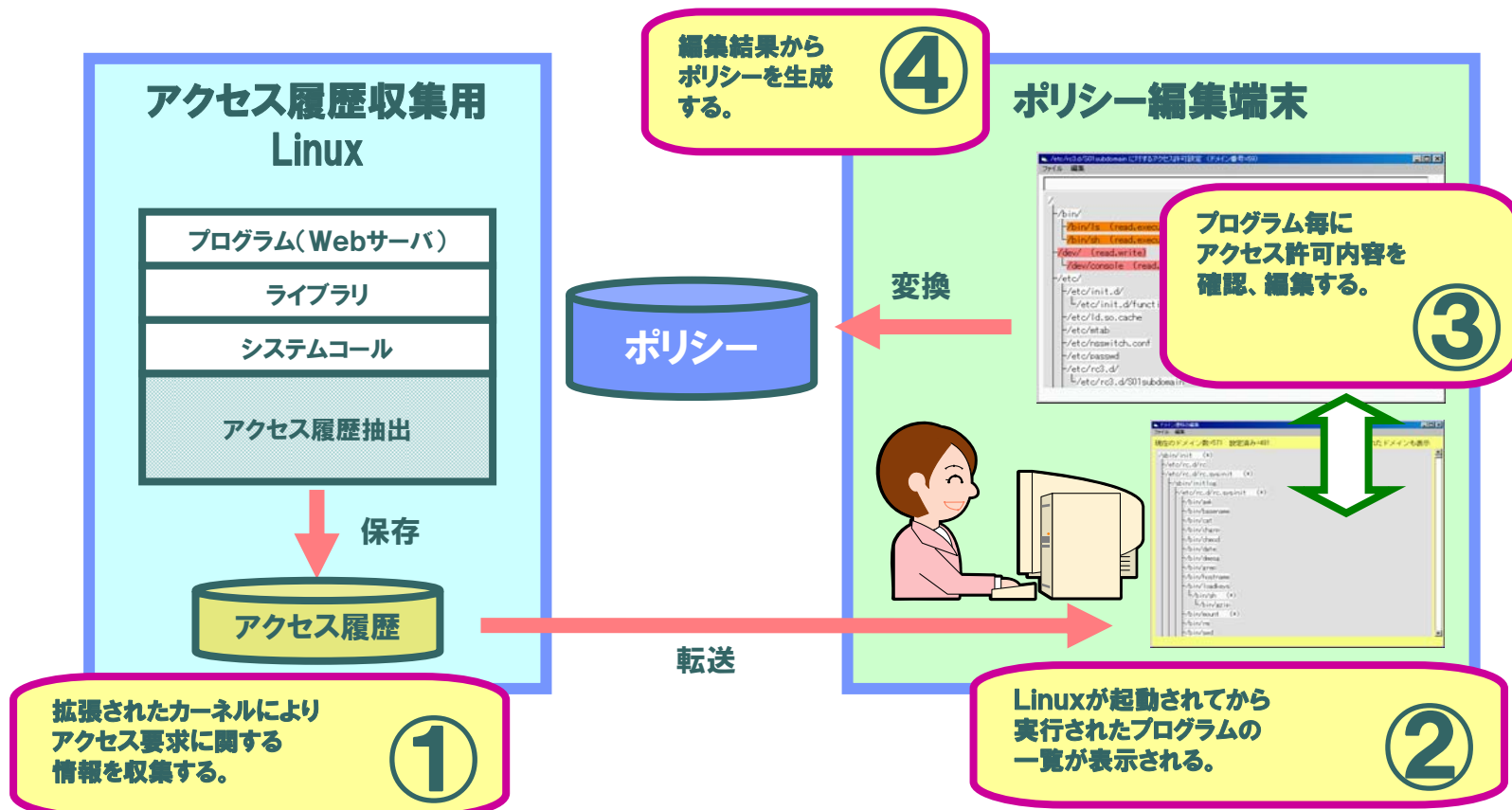
ちょっと走行記録を拝見・・・。

え〜と、あなたは
/sbin/init を経由して
/etc/rc.d/rc を経由してきた
/bin/egrep
さんですね。

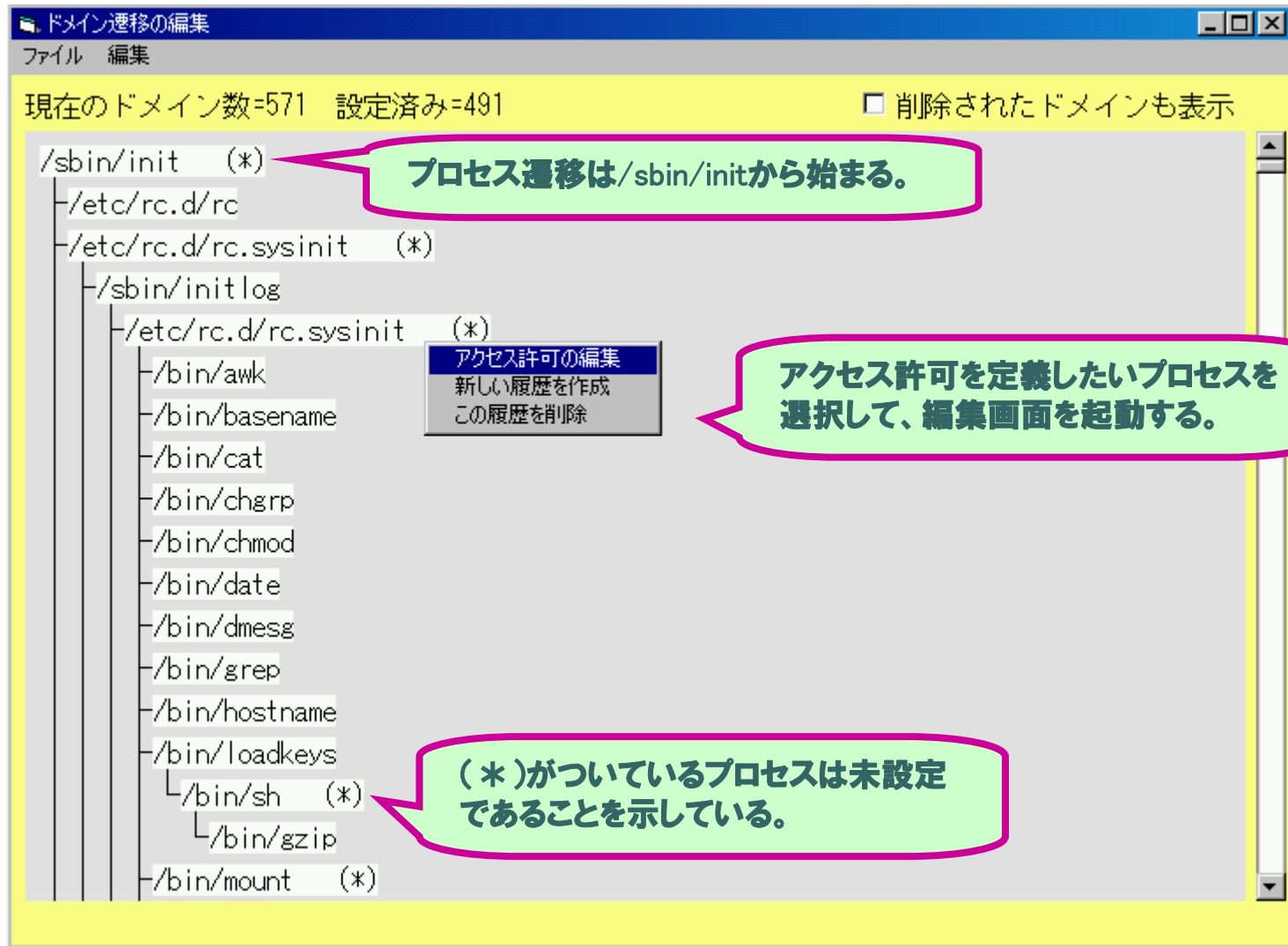
では、この場所を記録に追加しました
ので、通過下さい。

ポリシーの生成と編集

- 収集したアクセス情報をファイルとして保存、Windows PC上でVBで書いたプログラムで編集
- SubDomain風のポリシーを生成可能



ポリシーの生成と編集



ドメイン遷移の編集
ファイル 編集

現在のドメイン数=571 設定済み=491 削除されたドメインも表示

- /sbin/init (*)
 - /etc/rc.d/rc
 - /etc/rc.d/rc.sysinit (*)
 - /sbin/initlog
 - /etc/rc.d/rc.sysinit (*)
 - /bin/awk
 - /bin/basename
 - /bin/cat
 - /bin/chgrp
 - /bin/chmod
 - /bin/date
 - /bin/dmesg
 - /bin/grep
 - /bin/hostname
 - /bin/loadkeys
 - /bin/sh (*)
 - /bin/gzip
 - /bin/mount (*)

プロセス遷移は/sbin/initから始まる。

アクセス許可を定義したいプロセスを選択して、編集画面を起動する。

(*)がついているプロセスは未設定であることを示している。

アクセス許可の編集
新しい履歴を作成
この履歴を削除

ポリシーの生成と編集

`/etc/sysconfig/network-scripts/ifup` に対するアクセス許可設定 (ドメイン番号=96)

ファイル 編集

```

/dev/console 読み書き
/
/bin/
  /bin/basename (read,execute)
  /bin/grep (read,execute)
  /bin/ipcalc (read,execute)
  /bin/sed (read,execute)
  /bin/sleep (read,execute)
/dev/ (read,write)
  /dev/console (read,write)
  /dev/null (read,write)
/etc/
  /etc/init.d/
    /etc/init.d/functions
  /etc/ld.so.cache
  /etc/mtab
  /etc/nsswitch.conf
  /etc/passwd

```

読み込みのみ (read)
読み書き (read-write)
実行可能 (read,execute)
検出のみ (scan)
追記のみ (read,append)
全て許可 (ALL)
全て禁止 (DENY)

背景が白となっているのは、「読み込み」のみを許可。

この画面は `/etc/sysconfig/network-scripts/ifup` に関するアクセス許可を編集するためのものである。

自動学習されたアクセス許可を変更したい場合は、該当するアクセス対象を選択してメニューから変更する。

2003.10 NSF2003

- Network Security Forum 2003で、「プロセス実行履歴に基づくアクセスポリシー自動生成システム」として論文を発表
- プロセスの実行履歴を元に「ドメイン」を自動的に定義し、ドメイン毎にアクセスを追跡する仕組みが完成

Network Security Forum 2003

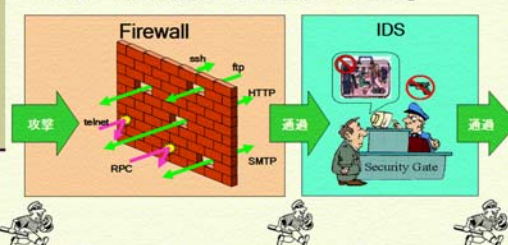
**「プロセス実行履歴に基づく
アクセスポリシー自動生成システム」**

平成15年10月22日
株式会社NTTデータ
技術開発本部
原田季栄 haradats@nttdata.co.jp

Network Security Forum 2003

ファイアウォールとIDS

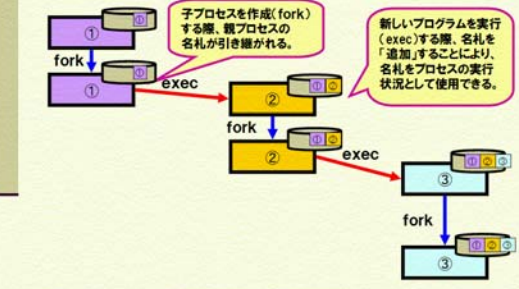
- ファイアウォール…不必要な通信の遮断
- IDS…既知の攻撃手法との照合による「検知」



Copyright © 2003 NTT DATA CORPORATION. All rights reserved.

Network Security Forum 2003

プロセス遷移の追跡方法



子プロセスを作成 (fork) する際、親プロセスの名札が引き継がれる。

新しいプログラムを実行 (exec) する際、名札を「追加」することにより、名札をプロセスの実行状況として使用できる。

Copyright © 2003 NTT DATA CORPORATION. All rights reserved.

SELinuxの壁

- SELinux用ポリシー生成の困難さに直面する。
 - アクセスを許可すべきパス名からラベルへの変換ができない。
 - ハードリンクに対しては1個しかラベルを付けられない
 - iノード番号は変化する可能性があるのに、iノードに対してラベルを割り当てなければいけない。
 - ラベル名の割り当て方が一意ではない。
 - ドメイン名に使える文字が制限されている。
 - 例えば ‘/’ を含めることができない。
 - URLエンコーディング等を施すと可読性が低下する。
- 結論：
 - ラベルに基づく強制アクセス制御は難しい。

Stage 4

使いになせるアクセス制御を実現せよ!

2003.11~2004.11





ポリシーを生成できない理由

- ラベル名の割り当て方が一意ではないから
 - アクセス可否の判断をiノードで行おうとしているから
- ラベル名から状態遷移を把握できないから
 - 命名規則が解りにくい
- ラベルに基づくアクセス制御は難しい



ラベルを用いないアクセス制御

- アクセス可否の判断はパス名で行うべき
 - プログラムはパス名でアクセスしている。
 - iノードの状態について利用者に意識させたくない
- task_structを中心に考える
 - ドメイン遷移をツリー状に制限
 - fork () /execve () の仕組みを最大限に活かせる
 - 状態遷移のループにより生じる混乱を防止
 - ドメイン遷移の把握を容易に



ラベルを用いないアクセス制御

- d エントリから絶対パス名の導出に成功
 - i ノード番号では無く、パス名での制御が可能に
- パス名のワイルドカード指定に対応
 - /proc/PID 等にも対処
- 強制アクセス制御を有効にしたままでのシャットダウン処理に成功
 - 強制アクセス制御を無効にしないとシャットダウン不可能なのではと心配していた。

2004.2 NET&COM講演と展示



2004.2 NET&COM講演と展示

- プロジェクト発足後最初の講演
- 一般の人に理解してもらうためにアニメーションを制作した（これ以降の講演でも活用）
 - バッファオーバーフロー
 - ファイアウォールとIDSの限界
 - 強制アクセス制御
 - システムI/Fの互換性について
- OSセキュリティ強化の全体像と現状および課題について紹介



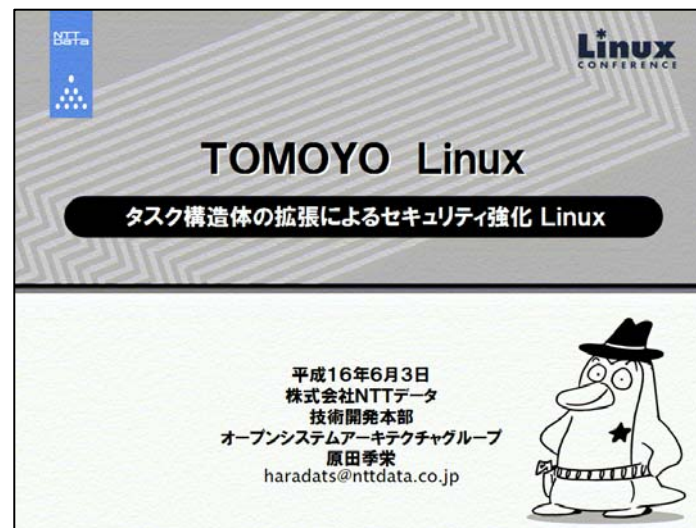
2004.5 ビジネスシヨウ2004講演

- 「何故セキュリティ強化が必要か」を実感してもらうために脆弱性の実演（デモ）を行った。
 - samba
 - wu_ftpd
 - ローカル権限昇格
- 自社の取り組みについても紹介



2004.6 LC2004

- Linux Conference 2004で、「TOMOYO Linux-タスク構造体の拡張によるセキュリティ強化 Linux」を発表
- “TOMOYO Linux”という名前のデビュー
- 発表の中でポリシー学習のデモを実施
- 説明資料として、当日説明用にイラストを取り入れ紙芝居風にしたものと技術的解説を加えたものの2種を作成







機能の向上

- ポリシーの動的「追加」に対応
 - /procインタフェースを使用
 - /procの使い方が解るまでは、readlink()インタフェースを使っていた。
- ポリシーエディタ(CUI)開発開始
- ケイパビリティへの対応を開始
 - POSIX項目をそのまま利用
- カーネル2.6.9への対応を開始

2004.10 CEATEC講演

- CEATEC 2004で「Linuxセキュリティ強化エッセンシャル」として講演
- 機能面ではあまり進歩がないが、プレゼンテーション面ではかつてないレベルまでこだわった

	Linuxセキュリティ強化エッセンシャル
	<small>株式会社NTTデータ オープンソース開発センター シニアスペシャリスト 原田孝栄 <haradats@nttdata.co.jp></small>
	<ul style="list-style-type: none"> • 1時間で理解する「Linuxセキュリティ強化」のエッセンス <ul style="list-style-type: none"> - 既存OSが直面しているセキュリティ上の脅威とは - 既存のツールでは守れないのか - OSのセキュリティ強化が必要な理由 - Linuxの「セキュリティ」は十分か? - セキュリティ強化OS、高信頼OS、「基準」 - 「強制アクセス制御」とは - SELinuxについて - 残された課題 - NTTデータの取り組み
	<small>Copyright©2004 NTT DATA Corporation</small>

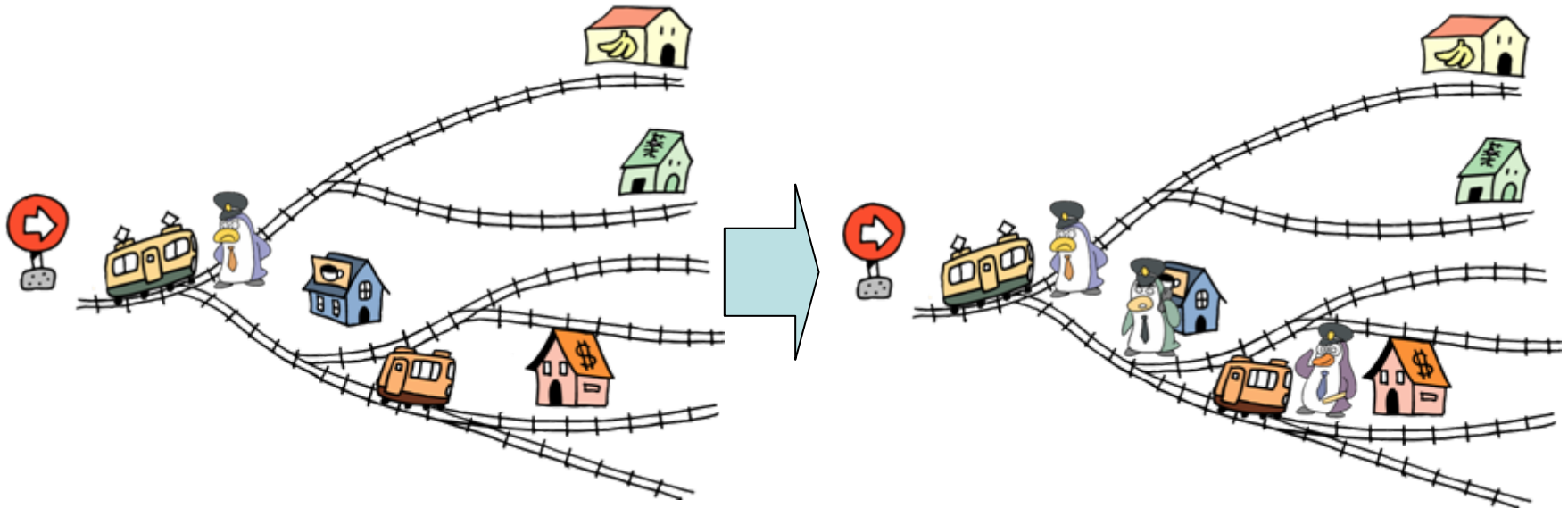
	考えられる脅威
	<small>KEYWORDS:</small>
	<ul style="list-style-type: none"> • 「セキュリティ強化」とは考えられる脅威への対策、予防策です。 • 脅威について正しく理解することが必要です。 • 「これで完璧」はありません。
	<small>Copyright©2004 NTT DATA Corporation</small>

	SELinuxの運用
	<small>KEYWORDS: バッファオーバーフロー、任意アクセス制御(DAC)、最小権限、強制アクセス制御(MAC)、セキュリティ強化OS、高信頼OS、TCSEC、ISO/IEC 15408、ポリシー、SELinux</small>
	<ul style="list-style-type: none"> • SELinux等の実装ではポリシーは、システムコール(カーネルに対するインタフェース)単位で記述します。
	<small>Copyright©2004 NTT DATA Corporation</small>

不正なログインの防止

- 追加認証機構 (CERBERUS) の開発
 - (SELinuxのように) sshdの改造はできない
 - 正規の手順で侵入されては意味が無い
 - ドメイン遷移を応用してログイン認証を強化
 - 絶対に突破されないログイン認証を実現
- サーバ管理業務の一部委託も可能 (YUE)
 - RBACではないがロール相当の機能を実現
 - 全員がrootとしてログインしてもらっても大丈夫

不正なログインの防止



標準の認証(1段)

強化された認証(多段)

ポリシーエディタ(CUI)

```
<<< Domain Transition Editor >>> 382 domains
Commands = 'Q'uit 'D'elete 'R'efresh
<kernel>
- <kernel>
  0:
  1: /sbin/init
  2: /etc/rc.d/rc
  3: /bin/egrep
  4: /bin/grep
  5: /bin/grep
  6: /etc/rc.d/init.d/FreeWnn
  7: /bin/grep
  8: /bin/rm
  9: /bin/touch
 10: /bin/usleep
 11: /sbin/consoletype
 12: /sbin/initlog
 13: /sbin/killall5
 14: /usr/bin/jserv ( -> 378 )
 15: /etc/rc.d/init.d/anacron
 16: /bin/nice
 17: /sbin/initlog
 18: /usr/sbin/anacron ( -> 371 )
 19: /bin/rm
 20: /bin/touch
 21: /bin/usleep
```

ドメイン遷移の編集

```
<<< ACL Uviewer >>> 27 acls
Commands = 'Q'uit 'D'elete
<kernel> /sbin/init /etc/rc.d/rc /etc/rc.d/init.d/xinetd
-
  0: 1 /bin/rm
  1: 1 /bin/touch
  2: 1 /bin/usleep
  3: 6 /dev/console
  4: 2 /dev/null
  5: 6 /dev/tty
  6: 4 /etc/mtab
  7: 4 /etc/passwd
  8: 4 /etc/passwd
  9: 4 /etc/rc.d/init.d/functions
 10: 4 /etc/rc.d/init.d/xinetd
 11: 4 /etc/sysconfig/l18n
 12: 4 /etc/sysconfig/init
 13: 4 /etc/sysconfig/network
 14: 4 /etc/sysconfig/xinetd
 15: 4 /lib/i686/libc-2.3.2.so
 16: 4 /lib/libdl-2.3.2.so
 17: 4 /lib/libnss_files-2.3.2.so
 18: 4 /lib/libtermcap.so.2.8.8
 19: 4 /proc/meminfo
 20: 1 /sbin/consoletype
 21: 1 /sbin/initlog
```

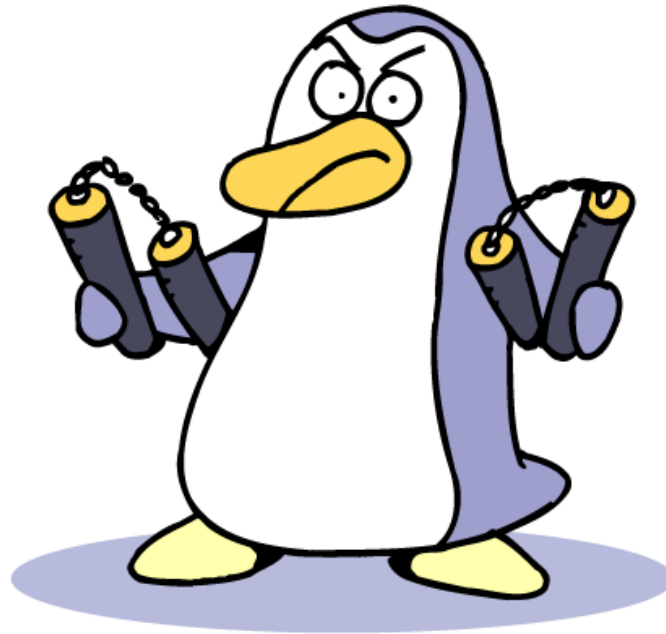
ドメインに対するアクセス許可の編集

デバイスファイルの改ざん防止

- アクセス制限付き/devファイルシステムの開発 (SYAORAN)
 - SAKURAでは/devを読み込み専用できない
 - TOMOYOではファイルの種別を考慮していない
 - SELinuxでも偽デバイスファイルを作成される可能性がある。
 - 例：/dev/sdaの属性を持った/dev/sdbというファイル
 - ファイル名とデバイスファイルの属性の対応を強制できるファイルシステムを開発

2004.11 セキュスタ2004

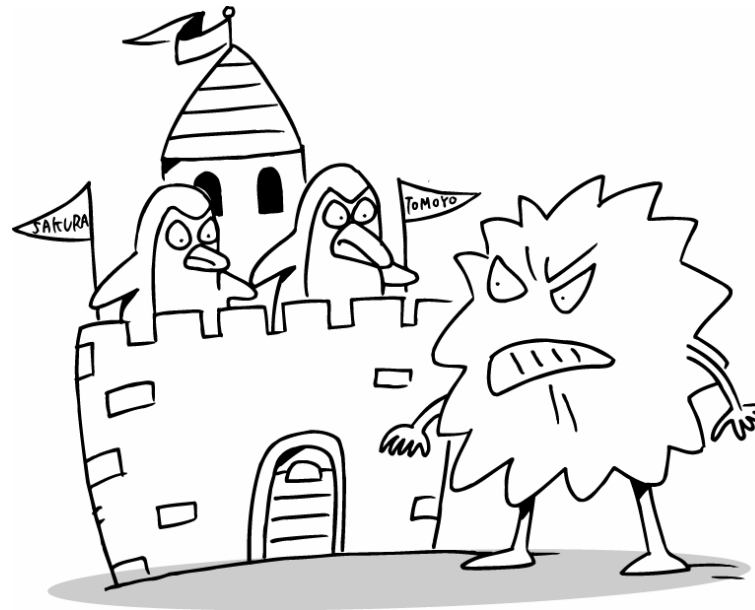
- セキュリティ・スタジアム 2004
 - 防御側システムとして参加
 - 改めてOSセキュリティ強化の必要性を認識させられた



Stage 5

TOMOYO Linux誕生

2004.12~2005.11



構成の見直し

- ドメイン割り当てを動的に行うよう方式を変更
 - メモリの節約に成功
- SAKURA用のカーネルパッチとTOMOYO用のカーネルパッチを統合して、1つのパッチに集約した。
- 「無条件読み込み可」に対応
 - 共有ライブラリファイル等

機能の拡張

- ケイパビリティチェックをLinux標準項目からTOMOYO独自の項目に変更
 - 2.6 では CAP_SYS_ADMIN を全てのプロセスがチェックしてしまうので、自動的にアクセス許可を追加できるTOMOYOには嬉しい仕様。
 - CAP_SYS_ADMINをチェックできないなら、POSIXに従う意味が薄れると判断。
- ポリシーファイルの別名保存に対応
 - 日付を含めることでバージョン管理が可能に



使い勝手の向上

- 信頼済みドメインに対応
 - メンテナンスが容易に
- ドメイン遷移例外に対応
 - カーネルモジュールの動的呼び出しとデーモンの再起動に対応
- 起動オプションによる機能選択に対応
- カーネル2.4用と2.6用を統合した。

2005.6 LC2005

- Linux Conference 2005で、「使いこなせて安全なLinuxを目指して」として論文を発表
- それまでの取り組み経過の全体像を整理した「インデックス」的な論文
- Frank Mayer氏の講演を聞いて感激した後だったため、SELinuxについて一歩踏み込んで整理している

Linux CONFERENCE

使いこなせて安全なLinuxを目指して
[当日説明用資料]



平成17年6月2日
株式会社NTTデータ
オープンソース開発センター
技術開発担当
原田孝崇
haradats@nttdata.co.jp

Linux CONFERENCE

SELinuxにおけるドメインの考え方

ドメインは階層構造を持たずにフラット。
プログラムの実行をトリガーとして他のドメインに移る。
ドメインは、ユーザ定義。



2005/6/2 Copyright © 2005 NTT DATA CORPORATION 複製厳禁・無断転載禁止 9

Linux CONFERENCE

TOMOYO Linuxの場合(ポリシー例)


「主体」となるプロセスのドメイン→プロセス実行履歴毎

```
<kernel> /usr/sbin/sshd /bin/bash
1 /usr/bin/passwd
<kernel> /usr/sbin/sshd /bin/bash /usr/bin/passwd
6 /etc/shadow
```

「客体」となる
オブジェクトのパス名

認めるアクセス許可内容
読み込み: 4
書き込み: 2
実行: 1

ドメインの包含関係より
このドメインに移れるのは
" <kernel> /usr/sbin/sshd /bin/bash " だけ。
他のドメインからこのドメインに
遷移してしまう心配は無用。



2005/6/2 Copyright © 2005 NTT DATA CORPORATION 複製厳禁・無断転載禁止 16



使い勝手の向上

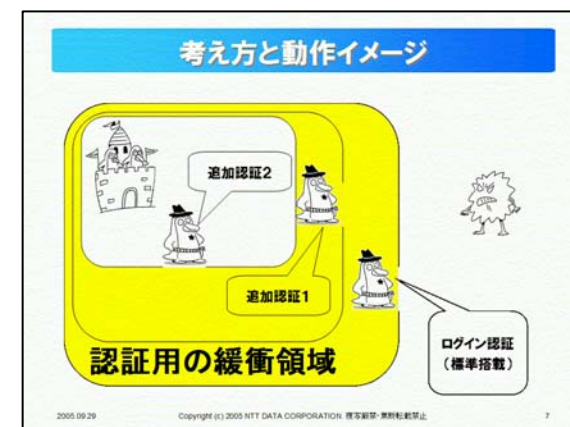
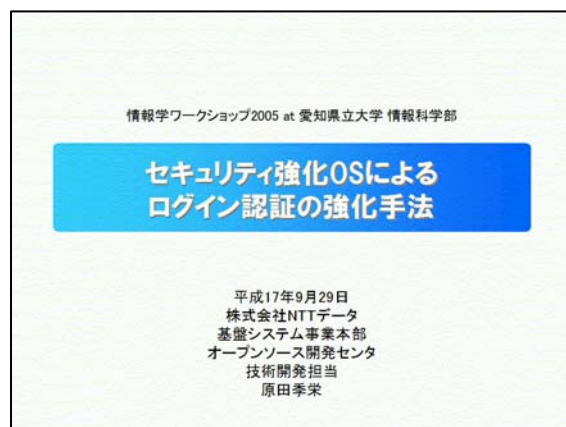
- アクセス許可ログ、アクセス拒否ログの取得に対応
- プロセスを再起動させずにポリシーの追加と削除に成功
- TCPおよびUDPのポート番号の制御に対応

使い勝手の向上

- ポリシーファイルの構成を見直し
 - 機能が増えてきたため、ポリシーファイルが10個を超えてしまった。
 - SAKURA用ポリシー、TOMOYO用ポリシー、TOMOYO例外用ポリシーの3つに統合した。
- 各種パラメータの動的変更に対応
 - 機能毎に有効・無効を切り替えできるようになった。
- カーネルコンフィグメニューに対応
 - 必要とする機能だけを選択できるようになった。

2005.9 WiNF2005

- 情報学ワークショップ2005で、「セキュリティ強化OSによるログイン認証の強化手法」として論文を発表



2005.11

TOMOYO Linux公開へ!

<https://sourceforge.jp/projects/tomoyo/>

