Security-Enhanced Linuxの
アクセス制御ポリシ設定の簡易化

中村雄一　鮫島吉喜
日立ソフトウェアエンジニアリング (株)

目次

1. 背景･目的
2. SELinuxの機能
3. SELinux設定の問題点
4. SELinux設定ツール
5. 結論
6. 今後の課題

背景

侵入による不正アクセス多発

侵入：セキュリティホールを利用しプログラムを
乗っ取り破壊活動すること

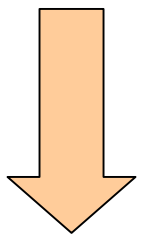
アプリケーションレベルの対策（パッチ当て等）ではいたちごっことなる

OSレベルで
セキュリティ強化

Security-Enhanced Linux(SELinux)
に着目

目的

SELinuxのアクセス制御ポリシ設定は
非常に難解



- 設定が難解な原因を明らかにする
- 設定を簡略化する方法を提案する

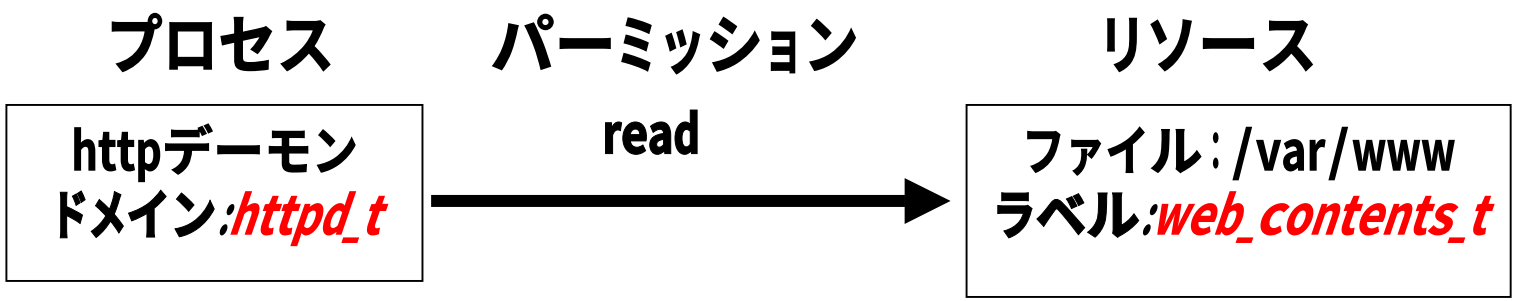
SELinuxの概要

- ＮＳＡで開発のLinuxカーネルパッチ
  - オープンソースで配布:http://www.nsa.gov/selinux/
- カーネルレベルの細かいアクセス制御機能
- 既存Linuxとバイナリ互換

SELinuxのアクセス制御機能

- 必須アクセス制御(MAC)
  - セキュリティ管理者だけがアクセス制御設定可
- ラベルベースのプロセス毎のアクセス制御

  ドメイン：プロセスの持つ権限名
  ラベル：リソースの識別名
  ドメインとラベル間でパーミッションを定義し設定



プロセスが必要最小限の権限で動作

SELinuxのアクセス制御ポリシ設定言語

- allow文：ドメインにラベルへのパーミッション付与

  例：**allow** httpd_t web_contents_t file:{ read };

  ドメイン　ラベル　リソース種別　パーミッション

- マクロ

  例：can_network(httpd_t)

  …
  allow httpd_t port_t:tcp_socket name_bind;
  allow httpd_t port_t:udp_socket name_bind;
  …

  ネットワーク利用許可の13個のallow文に展開

- リソースとラベルの関連付け

  例：/var/www(|/.*)　system_u:object_r:web_contents_t

SELinuxの設定の問題点（概要）

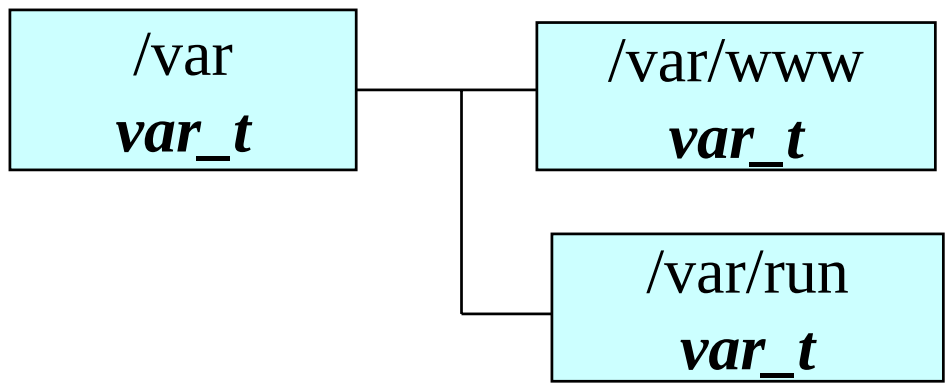
- リソースのラベル付けが煩雑
- 設定項目が膨大
- 設定の確認が困難

SELinuxの設定の問題点（1）

- **リソースのラベル付けが煩雑**
  - ラベル付けの競合
  - ラベル名の重複

ラベル付けの競合の例：

・ファイル・ラベルの対応



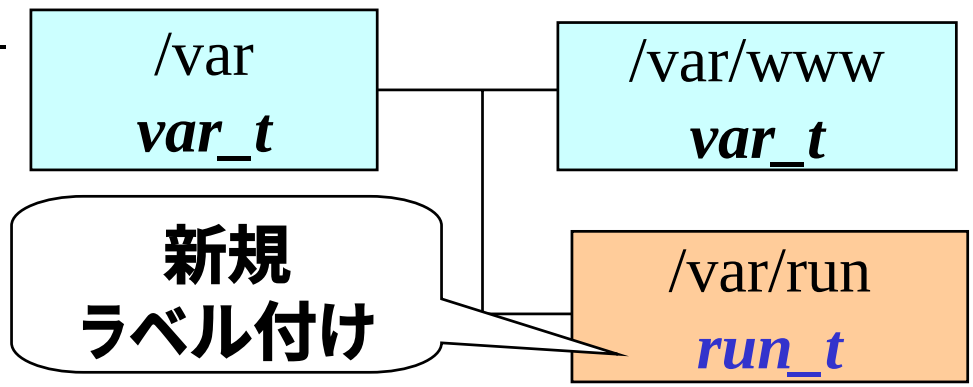
・some_tは/var以下をread可：**allow** *some_t var_t* file:{*read*}

SELinuxの設定の問題点（1）

- リソースのラベル付けが煩雑
  - ラベル付けの競合
  - ラベル名の重複

ラベル付けの競合の例：

・ファイル・ラベルの対応



・some_tは/var以下をread可：**allow** *some_t var_t* file:{*read*}

新規ラベル付けしたファイルにsome_tはアクセス不可になり設定の追加が必要

SELinuxの設定の問題点(2・3)

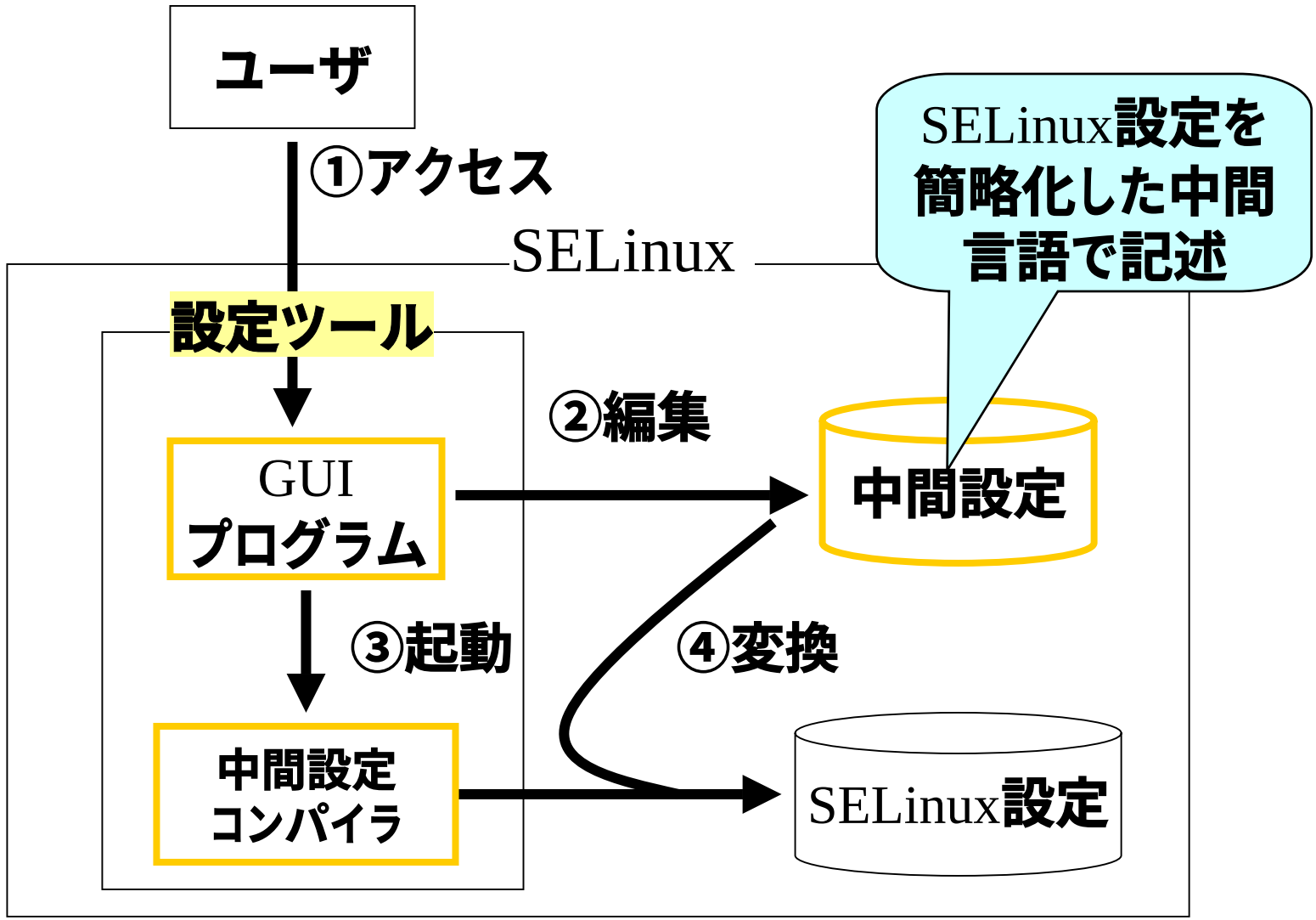
- **設定項目が膨大**
  - リソース種別、パーミッション、マクロが膨大

  例:ファイル種別7種、ファイルパーミッション17種
  マクロ150個以上

  →設定ファイル数200以上、行数10000行以上に
- **設定の確認が困難**
  - テキストベース
  - リソースにアクセス可能なドメインが分からない

設定状況の把握・設定の編集が困難

SELinux設定ツールの構成



中間設定コンパイラ：中間設定言語

- リソースを直接指定

例：

```
domain httpd_t;
allow /etc/httpd r;
allownet -tcp -port 80;
```

httpd_tドメインは
/etc/httpdを読み込み可
TCP80番ポートを使用可

- 設定項目の絞込み

ーリソース種別とパーミッションのうち使用頻度の低いものを統合

例：

従来の設定言語	中間設定言語
7種類のファイル種別 file dir lnk_file chr_file blk_file sock_file fifo_file	「ファイル」に統合
4種類のパーミッション read getattr ioctl lock	「r（読み込み）」に統合

中間設定コンパイラ：変換処理

1．リソース名からラベルを生成

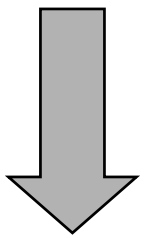
2．SELinux設定言語の出力

- allow文の出力
- リソースとラベルの関連付けを出力

中間設定コンパイラ：変換処理 (例)

変換前

```
domain one_t;      domain two_t;
allow /var r;      allow /var/www r;
```



/var以下：var_t
/var/www以下：var_www_t
というラベル生成

変換後

```
allow one_t var_t file:r_file_perms;
allow one_t var_www_t file:r_file_perms;
allow two_t var_www_t file:r_dir_perms;
…その他6つのファイル種別にも同じallow文
/var/(/.*) system_u:object_r:var_t
/var/www(/.*) system_u:object_r:var_www_t
```

子ディレクトリに対するラベルのallow文も出力、ラベルの競合解消

GUIの機能

設定状況を視覚的･多角的に表示

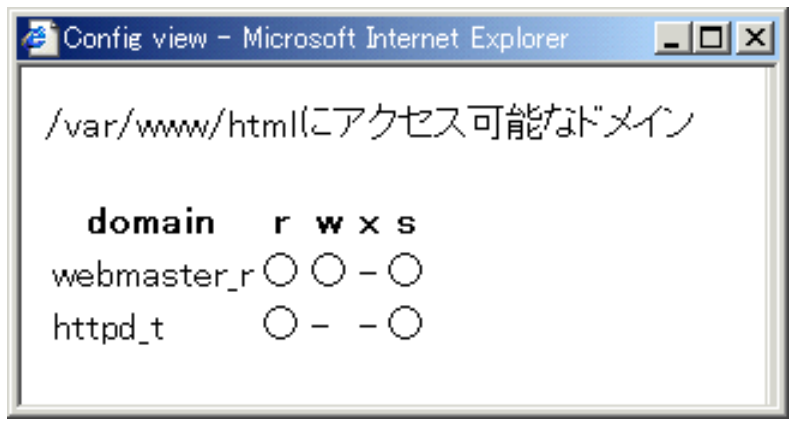
ドメインから見た権限

現在のドメイン :httpd_t :現在のディレクトリ :/var/www

	r	w	x	s	子ディレクトリに適用	
cgi-bin	☑	☐	☑	☑	○ no	◉ yes
html	☑	☐	☐	☑	○ no	◉ yes
icons	☑	☐	☐	☑	○ no	◉ yes

適用

リソースから見た権限



結論

- SELinuxの設定の問題点を洗い出した
  - リソースのラベル付けが煩雑
  - 設定項目が膨大
  - 設定の確認困難
- SELinux設定ツールを試作した
  - リソースを直接指定（中間設定）
  - 設定項目の絞込み（中間設定）
  - 設定を視覚的・多角的に確認（GUI）

SELinuxの設定の問題点を解決できた

今後の課題

- 設定の脆弱性監査
  - 現状は目視。自動的に脆弱性を監査するツールが必要
- 操作性向上
  - 同様の設定を多くする場合に操作が煩雑。テンプレート機能が必要