# SELinux Policy Editor(seedit) 管理ガイド（マニュアル) 2.0

中村　雄一*

October 27, 2006

## Contents



*himainu-ynakam@miomio.jp

この文書は、SELinux Policy Editor 2.0 のマニュアルです。基本概念や設定方法を紹介しています。インストール方法については、インストールガイドを参照してください。

# 1 背景：SELinux の難しさ

SELinux は既に多くのディストリビューションに取り込まれています。しかし、多くのユーザーは SELinux を無効にしてきました。SELinux は手に負えない、と感じるユーザーが多かったからです。SELinux の難しさはポリシーの設定にあります。ポリシーが難しい理由は以下が挙げられます。

- 多すぎるパーミッション
  SELinux のパーミッション定義は７００種類にも及びます。細かい設定ができる反面、必要な設定が増大します。
- ラベル管理が大変
  SELinux は、ファイルやポート番号などにタイプというラベルを付与し、アクセス制御を行います。ラベルは直感的に分かりにくい上，ファイルやポート番号のラベル管理の手間が生じます。
- 多すぎるマクロ
  SELinux の設定をする際には、「マクロ」を利用して、複数行の設定を一行での設定をまとめて行います。しかし、マクロの種類がどんどん増えています。例えば，BIND を設定するために使われているマクロだけでも、80 種類近くに及びます。一般的なシステム管理者には、設定内容の理解と設定の記述は極めて困難です。

# 2 SELinux Policy Editor(seedit) とは？

## 2.1 概略

SELinux Policy Editor(略称 seedit) は、SELinux を簡単にするツールです。seedit は、Simplified Policy(単純化ポリシー) と、GUI など Simplified Policy 周辺ツールから成り立つツールです。
最も重要な要素は Simplified Policy です。Simplified Policy とは、Simplified Policy Description Language(単純化ポリシ記述言語、以下 SPDL) で記述された SELinux のポリシのことです。SPDL は、SELinux の設定を大幅に簡略化します。SPDL は、セキュリティ上影響の少ないパーミッションを省略・統合することによってパーミッションの数を減らします。また、ラベルを隠蔽し、ファイル名、ポート番号を直接使って設定できるようにしています。同時に、マクロ地獄からも解放されます。
SPDL で記述されたポリシーが，SELinux の設定に変換され，設定が反映されます。
以下が、SPDL によって書かれた Simplified Policy の例です。Apache Web サーバーに「httpd_t」というドメイン（権限のこと）を割り当てて設定しています。

```
{
domain httpd_t;
program /usr/sbin/httpd;
...
allow /var/www/** r,s;
allownet -protocol tcp -port 80 server;
...
}
```

SPDL を使ったポリシーの意味は明快です。カスタマイズ、新規ポリシーの記述も簡単です。SPDL を生成するためのツールや GUI が用意されているからです。

### 2.2 seedit の歴史

SELinux Policy Editor の元となるバージョンは、日立ソフトウェアエンジニアリング (http://www.selinux.hitachi-sk.co.jp/) により開発され、2003 年 2 月に GPL にて公開されました。このバージョンをアップデートしたものが、バージョン 1.0 として 2005 年 7 月に公開されました。
現バージョン (2.0) は、中村が The George Washington University にて再設計し、日立ソフトの実装を一部使いながら，再実装、新規開発したものです。

### 2.3 seedit の限界

seedit は、到達できるセキュリティレベルに限界があります。Simplified Policy の、情報フロー分析可能性が証明されていません。「ポリシーは、形式的（数学的）に分析可能であるべきである」と考える人は、seedit を使ってはいけません。SELinux の strict ポリシーを頑張って使ってください。
Simplified Policy から生成される SELinux のポリシーを分析すれば、情報フロー分析を可能かもしれません。ラベル付けのルールも、情報フロー分析に気をつかってはいます。しかし、情報科学的に証明された物ではありません。

## 3 SELinux の背景知識

seedit を使う前に，SELinux に関する最低限の知識を押さえておく必要があります。

(1) TE(Type-Enforcement)
SELinux のアクセス制御メカニズムは TE(Type Enforcement) と呼ばれています。TE では、プロセスには、ドメインと呼ばれる権限が割り当てられます。例えば、Apache Web サーバー (/usr/sbin/httpd) には、httpd_t というドメインが割り当てられます。ポリシーと呼ばれる設定ファイルに、ドメインがどんなリソース（ファイルなど）にアクセスができるかのルールが記述されています。リソースを識別するために、タイプと呼ばれるラベルを使います (seedit では隠蔽されます)。全てのプロセスは、ポリシーに記述された通りのドメインを与えられ，その権限の範囲内だけで動作します。SELinux の鍵となるのは、どんなドメインを割り当てるか、ドメイン

にどんなアクセスを許可するか、というポリシーの設定となります。
なお、ドメインの割り当てられるタイミングは、実行ファイルの実行時になります。ポリシーファイルに、「実行ファイルを実行した場合、特定のドメインを割り当てる」と書いてある場合のみ、ドメインが割り当てられます。あるアプリケーションに割り当てるドメイン名を変更した場合は、アプリケーションの再起動が必要になるのに注意が必要です。ドメイン名を変えず、ドメインの持つ権限のみを変えた場合はアプリケーションの再起動は不要です。

(2) Enforcing/permissive モード
SELinux には、enforcing モード、permissive モードと呼ばれる 2 つのモードが用意されています。
Enforcing モードは通常のモードです。ポリシーに書かれたアクセス制御設定が有効な状態です。
Permissive モードは、テストモードとも言えるモードです。SELinux によって拒否されるアクセスがあったとしても，それは拒否されず，ログにアクセス拒否の事実が記述されるだけ、というモードです。 Permissive モードでは、アプリケーションは通常の Linux と同様に動きます。が、アプリケーションがどんなアクセスをするのか挙動を調べ、ポリシーを書くのに役立ちます。現在のモードを確認するには、*getenforce* コマンドを使います。enforcing/permissive モードの切り替えには、*setenforce* コマンドを使います。使い方は、後で紹介します。

(3) SELinux のアクセス拒否ログ
SELinux によってアクセスが拒否された場合、auditd サービスが動作している場合は、 /var/log/audit/audit.log にログアクセス拒否ログが出力されます。一方、auditd サービスが動作していない場合は、/var/log/messages にアクセス拒否ログが出ます。dmesg コマンドでも閲覧可能です。

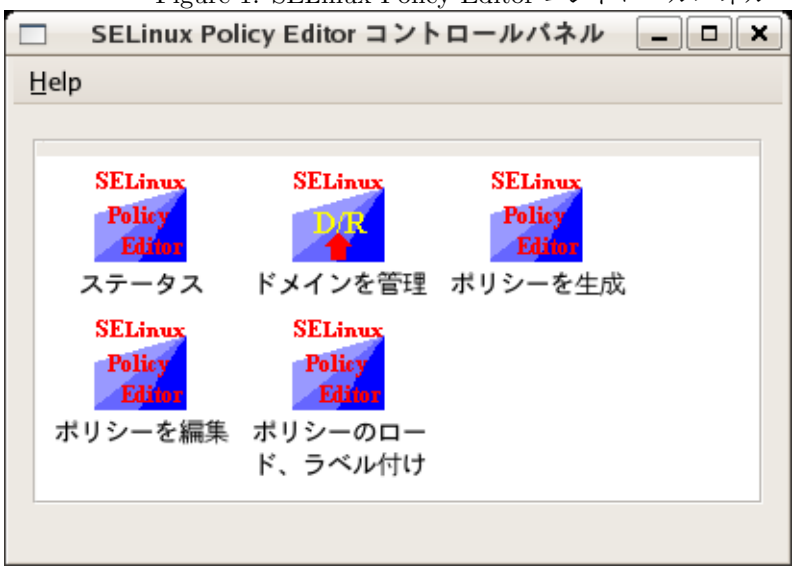
# 4 GUI の概要

seedit の GUI を使うことで、より簡単に SELinux を使うことができます。X Window System から、「seedit-gui」または「デスクトップ→管理→ SELinux Policy Editor」(Cent OS の場合は「アプリケーション」→「システム設定」→「SELinux Policy Editor」) または、コンソールから「seedit-gui」とすることで GUI が起動します。すると、*SELinux Policy Editor* コントロールパネルという画面が開きます。このコントロールパネルには、アイコンが 図 1 のように並んでいます (CentOS 4 では、アイコンの代わりにボタンが表示されます)。

アイコンをダブルクリック ( またはクリックしてリターン押下) すると、対応した管理ツールが起動します。それぞれのアイコンから設定できる設定内容は以下のようになります。

- ステータス
使い方は 5.4 節で。

Figure 1: SELinux Policy Editor コントロールパネル




  - SELinux のステータスを確認。SELinux のモード、アプリケーションに割り当てられたドメインを確認できます。
  - SELinux のモードを変更

- ドメインを管理
  - 新規ドメインを作成 (10.1 節)
  - ドメインの削除、無効化 (7.1 節)
- ポリシーを生成 (9 節)
  - SELinux のアクセスログからポリシーを生成
- ポリシーを編集 (8.3 節)
  - テキストベースのエディタからポリシーを編集。
- ポリシーのロード、ラベル付け
  - 手動でポリシーをロード
    GUI ツールの中では、必要な時に自動的にポリシーはロードされますが、手作業で行うこともできます。
  - 全ファイルのラベルを初期化
  - restorecon コマンドを実行

# 5　システムの状況を把握する

seedit をインストールすることで、システムに何が起こっているのかを把握しましょう。

## 5.1　Simplified policy はどこに?

Simplified policy は、/etc/seedit/policy 以下に「.sp」という拡張子のファイルとしてインストールされています（詳細は後ほど）。Simplified policy は、seedit-load コマンド（内部的に seedit-converter コマンドが実際の作業を行っています）によって、SELinux のポリシー (SELinux のバイナリポリシ、file_contexts ファイル) に変換されます。変換後のポリシーは/etc/selinux/seedit/policy, /etc/selinux/seedit/contexts/files にインストールされます。これらがカーネルに読み込まれていますが、/etc/selinux/seedit 以下のファイルを気にする必要は通常ありません。

## 5.2　デフォルトで用意されている Simplified Policy

デフォルトでインストールされている Simplified Policy は、「targeted ポリシー」相当のもので、設定が緩い部分があります。具体的に以下のようになってます。

- 選択されたデーモンプロセスだけが守られており、SELinux によって制限されないプロセスの存在を許容
  「SELinux によって制限されていないプロセス」には、「全てのアクセスを許可」するドメインが割り当てられます。アクセス制御は、普通の Linux のパーミッションチェックのみになります。

- RBAC(Role Base Access Control) は設定されていない

SPDL は、原理的には RBAC も設定可能ですし、strict ポリシー相当のものも記述可能です。RBAC については、「RBAC ガイド」を参照してください。

## 5.3　unconfined ドメイン

このように、デフォルトでは、SELinux で制限されていないプロセスが存在します。これらのプロセスには、「SELinux によって制限されないドメイン (以下 unconfined ドメインと呼びます)」が割り当てられています。unconfined ドメインは、全てのアクセスを許可するように設定されています。unconfined ドメインが割り当てられたアプリケーションは、SELinux によるアクセス制御を事実上受けなくなり、普通の Linux の上で動いているのと同様になります。
例えば、システム起動スクリプトには、initrc_t ドメインという unconfined ドメインが割り当てられています。
どのプロセスに unconfined ドメインが割り当てられているのかを把握することがセキュリティ上重要になってきます。

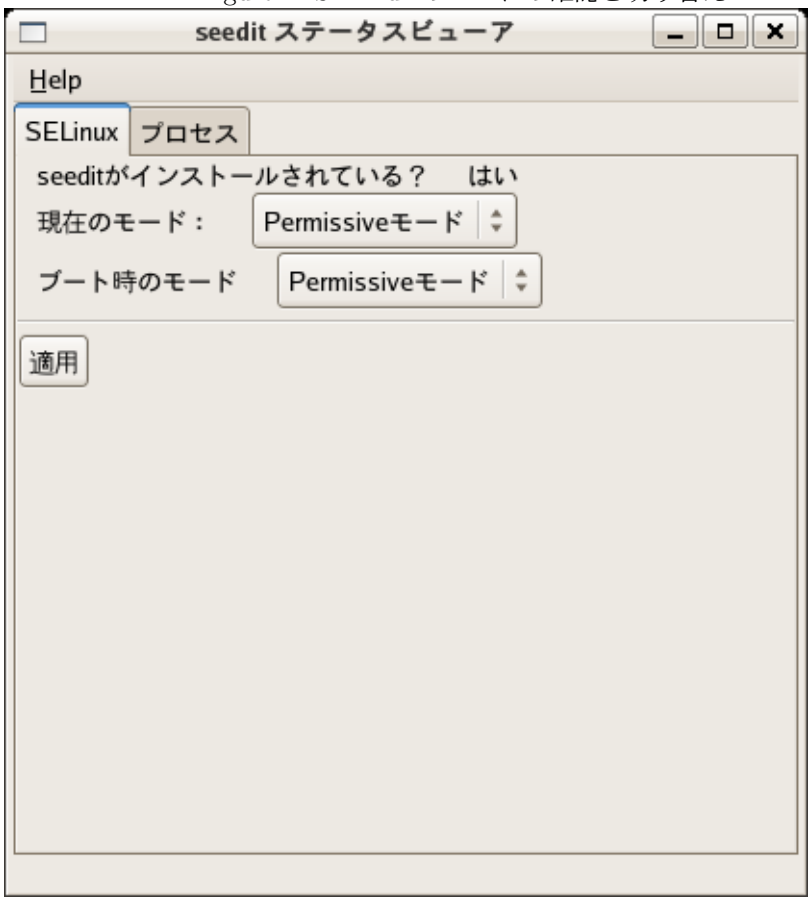
## 5.4　システムの状況把握 (GUI)

システムの状況を把握するために、 ステータス をコントロールパネルから選択します。*seedit* ステータスビューア というウィンドウが開きます。

### 5.4.1 モードの確認と切り替え

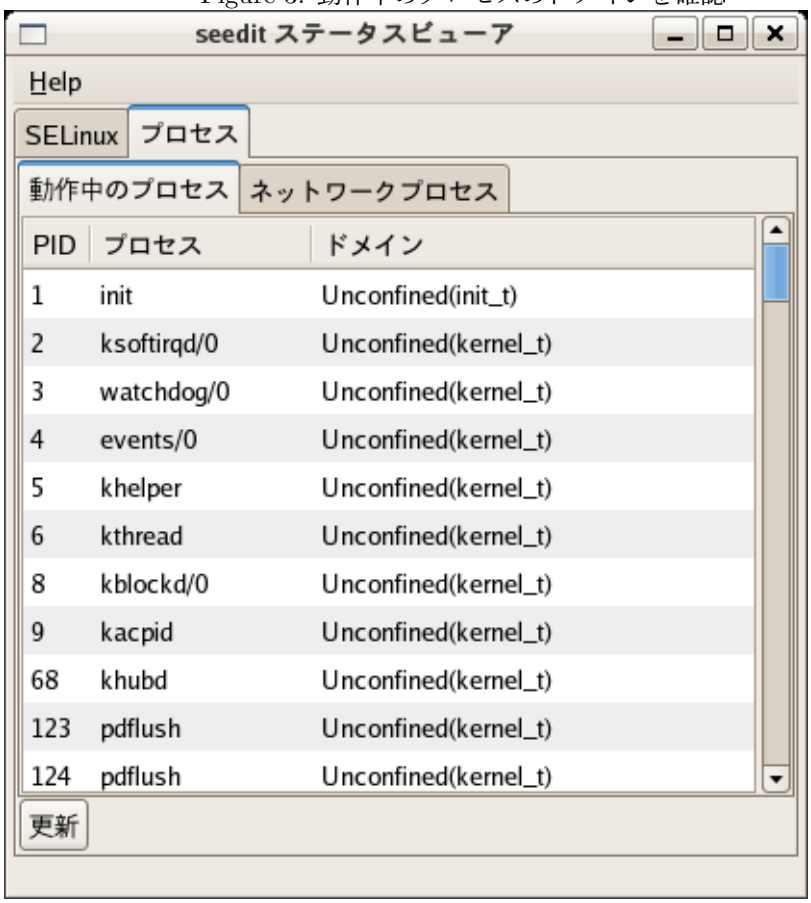
*SELinux* タブから、SELinux のモードを確認したり切り替えたりできます (図 2)。

Figure 2: SELinux のモードの確認と切り替え




*seedit*がインストールされている？ はい, という表示は、seedit が無事にインストールされていることを示しています。
現在のモードより、現在のモードが permissive モードであることが分かります。現在のモードの変更も、ここで行えます。例えば、Enforcing モードに変えるには、 *Enforcing* モードを選択し、 適用 ボタンを押します。
ブート時のモード は、システム起動時のモードです。 *Permissive* モードと表示されている時は、再起動時、システムが Permissive モードで起動します。実運用時には、両方とも Enforcing モードに切り替えるべきです。

Figure 3: 動作中のプロセスのドメインを確認




### 5.4.2 動作中のプロセスのドメインを確認

プロセスタブ → 動作中のプロセス, より、動作中のプロセスのドメインを確認できます。図は 3 実行例です。

Unconfined ドメインで動作しているプロセスには「Unconfined」と表示されています。例えば、bash は Unconfined ドメインで動作しています。一方、httpd には、httpd_t ドメインという通常のドメインが付与されていることが分かります。

PID, プロセス, ドメイン、をクリックすることで、それぞれの項目で結果をソートすることができます。

更新 ボタンを押すと、表示が更新されます。

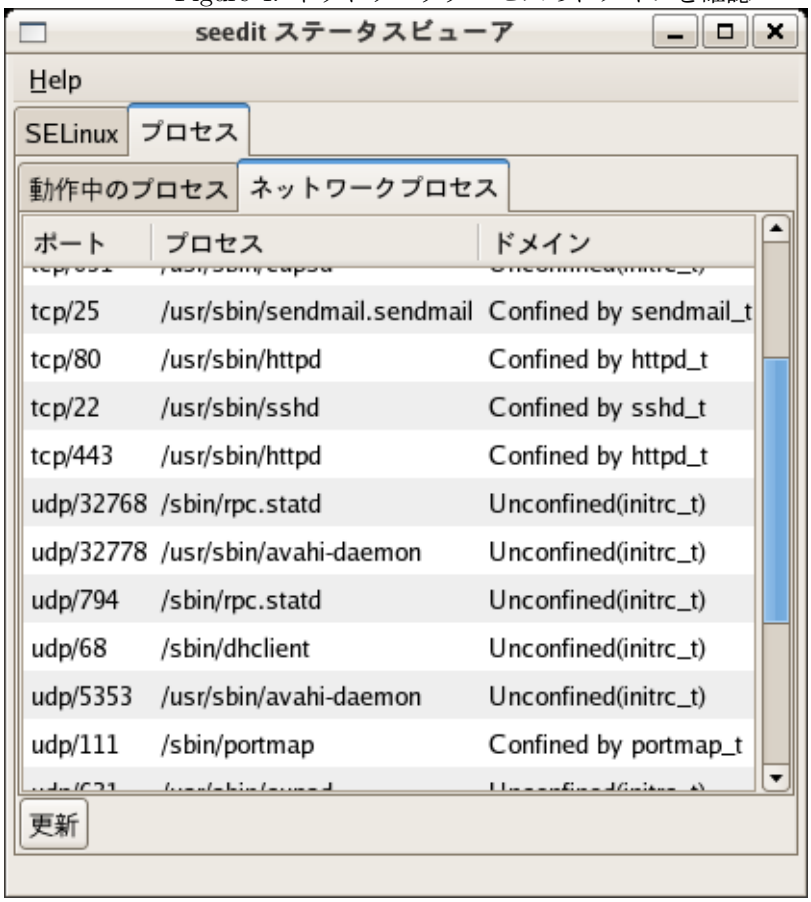
### 5.4.3 ネットワークプロセスのドメインを確認

ネットワークプロセス (ネットワーク接続を外部から受け付けるプロセス) は、攻撃者が侵入する際の入口として使われます。ここをしっかりと守ることが、外部からの不正侵入による被害を無くすために重要になります。以下のような状況が理想です。

- 全てのネットワークプロセスに適切なドメインが割り当てられている
- Unconfined ドメインが割り当てられた（SELinux によって制限されない) プロセスがある場合
  以下のような選択肢があります。
  - 適切なドメインを割り当てる設定をする
    設定方法は、後ほど紹介します。
  - SELinux 以外の対策を強化する
    ファイアウォール (iptables) によって、接続できるアドレスを制限したり、パッチ当ての優先度を上げるなどする
  - そのようなプロセスを立ち上げない（サービスを止める)
  - リスクを許容する

ネットワークプロセスのドメインは、プロセス → ネットワークプロセスより確認できます。実行例を図 4 に示します。avahi-daemon, rpc.statd などに Unconfined ドメインが割り当てられていることが分かります。これらのサービスは使わないならば、停止すべきでしょう。使うならば、ドメインを割り当てるべきです。

Figure 4: ネットワークプロセスのドメインを確認



## 5.5 システムの状況把握 (seedit-unconfined コマンド )

コマンドラインからもシステムの状況を把握可能です。seedit-unconfined コマンドを使います。seedit-unconfined コマンドは、root ユーザーになってから使います。

### 5.5.1 動作中のプロセスのポリシー適用状況を確認

「seedit-unconfined -e」にて、動作中のプロセスのポリシー適用状況を確認できます。以下に実行例を示します。

```
$ su -
# seedit-unconfined -e
Current SELinux mode: permissive ----(1)
PID     Comm    Domain
1       init    Unconfined(init_t) ---(2)
...
1853    sshd    Confined by sshd_t ---(3)
```

- (1) は、現在の SELinux のモードを示しています。「permissive モードである」と言っています。permissive モードでは、SELinux のアクセス制限がかからないことに今一度注意しましょう。
- (2) は、init プロセスは、SELinux に制限されていないことを言っています。そして、SELinux に制限されていないドメイン init_t が割り当てられています (init_t ドメインは、全てのアクセスが許可されていることを意味します。)。
- (3) は、sshd に、「sshd_t ドメイン」が割り当てられているという意味です。sshd_t ドメインは、sshd に必要最小限のアクセス許可を与えるように設定されています (設定内容は後で示す方法で確認できます)。

ちなみに、ps -eZ コマンドでも、動作中のプロセスのドメインを確認可能です。しかし、どのドメインが unconfined ドメインかを知ることはできません。SELinux の制限がかかってないドメイン一覧は、/etc/selinux/seedit/policy/unconfined_domains に記述されていますので、このファイルの内容と照合する必要があります。

### 5.5.2 ネットワークプロセスのドメインを確認

ネットワークプロセス (外部からネットワーク接続を待ち受けているプロセス) の状況を seedit-unconfined -n コマンドで確認できます (AppArmor の unconfined‘コマンドみたいなものです)。

さて、実際に確認してみましょう。seedit-unconfined -n の実行例を以下に示します。

```
#seedit-unconfined -n
Current SELinux mode: permissive ----(1)
/usr/sbin/smbd  Unconfined(initrc_t) -- (2)
/usr/sbin/sendmail.sendmail     Confined by sendmail_t --(3)
...
```

ネットワークプロセス一覧が表示され、ドメインの適用状況が表示されます

- (1) は現在の SELinux のモードです。
- (2) は、smbd は制限されていないことを言っています。
- sendmail は、sendmail_t ドメインが割り当てられており、sendmail_t は、最小限の権限を持つよう設定されてます。

この場合、smbd に対して何らかの対処をしないとセキュリティを保てません。

### 5.5.3 Enforcing/Permissive モードを切り換える

インストール直後は、permissive モードですが、以下のコマンドで Enforcing モードに切り換え可能です。

```
# setenforce 1
# getenforce
enforcing
```

ただし、これだと再起動時にまた Permissive モードに戻ってしまいます。ブート時から Enforcing モードにするには、/etc/selinux/config を次のようにします。実運用の際には、このように Enforcing モードにすることを強く薦めます。

```
SELinux=permissive
-->
SELINUX=enforcing
```

## 6 次に何をすればいいの？

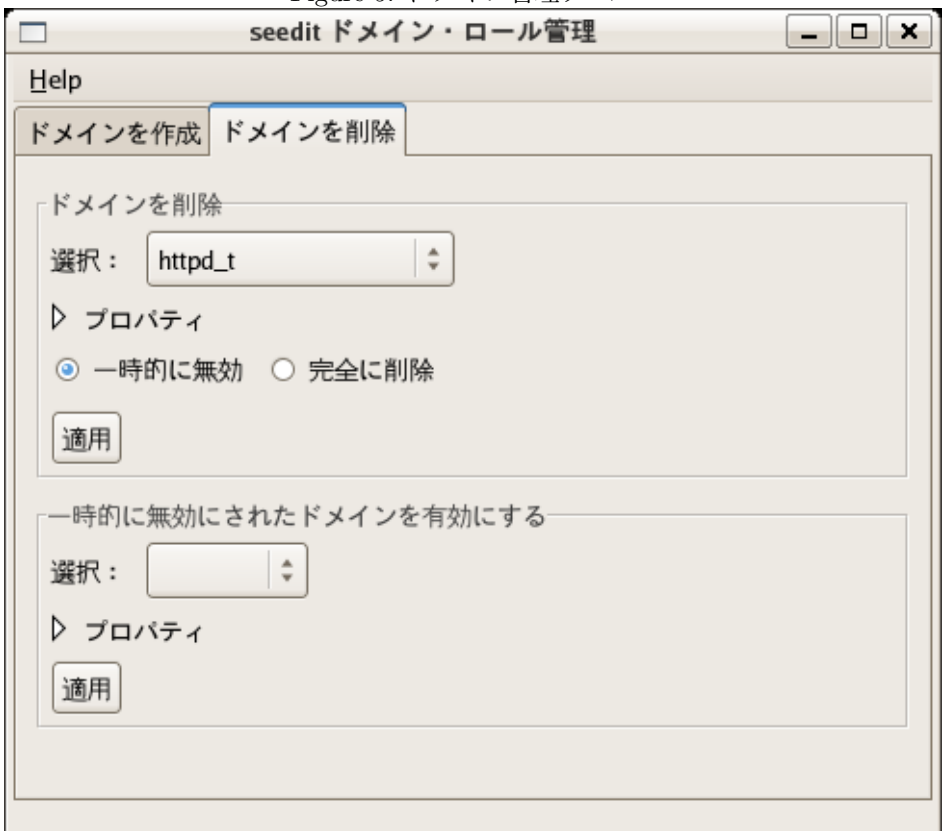
さて、SELinux のステータスを把握したところで、次に何をすればいいのでしょう。以下のようにまとめることができます。

(1) アプリケーションの SELinux 保護を無効に
アプリケーションが、SELinux のアクセス拒否のため動作しなかった場合、もっとも簡単な解決方法は、そのアプリケーションに対してのみ、SELinux のアクセス制御が働かないようにすることです。セキュリティはもちろん落ちますが、SELinux 全部を無効にするよりはマシといえます。 7 で紹介します。

(2) ポリシーを修正する
アプリケーションが、SELinux のアクセス拒否のため動作しなかった場合、本来は、ポリシーを編集すべきです。 9 で紹介します。

(3) ドメインを新たに設定する
SELinux によって動作が制限されていないアプリケーションの安全性を高めるには、自分でドメインを設定し、アプリケーションにドメインを割り当てる必要があります。 10.1 で紹介します。

Figure 5: ドメイン管理ツール




# 7　アプリケーションに対する SELinux 保護を無効に

アプリケーションに対する SELinux 保護を無効にする方法は 2 つあります。boolean パラメータを使う方法と、ポリシーファイルを移動する方法です。ここで「アプリケーションに対する SELinux 保護を無効にする」と言っていますが、前述の unconfined ドメインを割り当てることにより、SELinux のアクセス制御が全部許可されるようにする、という意味です。GUI およびコマンドから作業することができます。

## 7.1　GUI(ドメイン管理ツール)

GUI から作業するには、 コントロールパネルからドメインを管理を選択します。*seedit* ドメイン・ロール管理というウィンドウが開きます (図 5)。ドメインを削除タブを選択します。

### 7.1.1 方法１：一時的に無効に

アプリケーションの SELinux 保護を無効にする最も簡単な方法は、以下のステップでドメインを一時的に無効にすることです。

(1) 無効にしたいドメインを選択

(2) 一時的に無効 ラジオボタンを選択

(3) 適用　ボタンを押す

Apache Web サーバに unconfine ドメインを割り当てたい場合を見てみます。Apache には httpd_t ドメインが割り当てられていますので、 httpd_t を選択し、適用ボタンを押します。Apache を再起動し、 ステータス GUI から確認すると、ドメインとして *Unconfined(initrc_t)* が表示されます。
もう一度動作を制限するには、 一時的に無効にされたドメインを有効にするからドメインを選択し、　 適用 ボタンを押します。
ちなみに、これらの操作は内部的には SELinux の boolean を使っています。詳細を知りたい人はコマンドラインから操作してみるといいでしょう。

### 7.1.2 方法 2：ドメインを削除

アプリケーションの SELinux 保護を無効にするもう一つの方法は、次の手順のようにドメインの設定ファイルを除去することです。

(1) 無効にしたいドメインを選択

(2) 完全に削除 を選択

(3) 適用ボタンを押す

しかし、元に戻したい場合は、手作業でやらなくてはなりません。

(1) cd /etc/seedit/policy

(2) mv /etc/seedit/policy/extras/ドメインの名前.sp /etc/seedit/policy/ドメインの名前.sp

(3) seedit-load

(4) ドメインを削除したサービスを再起動

この方法の良い点は，変換後の SELinux のポリシーのサイズが小さくなることぐらいです。

## 7.2 コマンド ライン

### 7.2.1 方法１：boolean パラメータを使う

SELinux の boolean パラメータ (条件変数とも呼ばれます) を知っているのならば、簡単にできます。例えば、Apache の場合、httpd_t ドメインが割り当てられていますので、httpd_disable_trans を on にして、Apache を再起動するだけです。

再起動するのは、ドメインの割り当ては実行ファイルの実行のタイミングで起こるためです。httpd_t ドメインというドメインが割り当てられてるのを、unconfined なドメイン（この場合 initrc_t) を割り当てるようになります。割り当てたいドメイン名が変わるので、Apache を再起動する必要があります。

実行例:

```
# setsebool -P httpd_disable_trans 1
# /etc/init.d/httpd restart
# seedit-unconfined -e
Current SELinux mode: enforcing
PID     Comm    Domain
1111     httpd    Unconfined(initrc_t)
```

元に戻したいときは、boolean を off にします。

```
Example:
# setsebool -P httpd_disable_trans 0
# /etc/init.d/httpd restart
# seedit-unconfined -e
Current SELinux mode: enforcing
PID     Comm    Domain
1111     httpd   Confined by httpd_t domain
```

### 7.2.2 方法２：ポリシーファイルを移動する

ドメインが設定されているファイルは、/etc/seedit/policy/ドメイン名.sp. というファイルです。/etc/seedit/policy ディレクトリ以外にある設定は無効になります。この振舞を利用し、ファイルを別のディレクトリに移動して、設定を反映し直し、アプリを再起動すれば OK です。

実行例:
Apache の SELinux 保護を無効に

```
# cd /etc/seedit/policy
# mkdir unused
# mv httpd_t.sp unused
# seedit-load
# /etc/init.d/httpd restart
# seedit-unconfined -e
Current SELinux mode: enforcing
PID     Comm    Domain
1111     httpd    Unconfined(initrc_t)
```

元に戻したい場合は、/etc/seedit/policy ディレクトリに再度移動し、設定を反映します。

```
Example:
# cd /etc/seedit/policy
# mv unused/httpd_t.sp .
```

Figure 6: Simplified Policy の例：Apache Web サーバのためのポリシ

```
 1  {
 2  domain httpd_t;
 3  program /usr/sbin/httpd;
 4  include common-relaxed.sp;
 5  include daemon.sp;
 6  include nameservice.sp;
 7  allow /var/www/** r,s;
 8  allow /var/log/httpd/** r,a,s;
 9  allow /etc s;
...<snip>..
10  allownet -protocol tcp -port 80,443 server;
11  allowpriv netlink;
12  }
```

```
# seedit-load
# /etc/init.d/httpd restart
# seedit-unconfined -e
...
```

# 8 Simplified Policy の基礎知識

Simplified Policy を扱う前に、基本的な知識をおさえておきましょう。

## 8.1 どこに？

Simplified Policy は、/etc/seedit/policy ディレクトリに配置されています。このディレクトリの下には、ドメイン名.sp というファイルが配置されています。

## 8.2 書式概要

Simplified Policy は、Simplified Policy Description Language(SPDL) という書式で書かれています。詳細は、別のドキュメント (Specification of SPDL) に書かれていますが、全てを理解する必要はありません。書式を知らなくとも、Simplified Policy を生成するツールがあるからです。
が、SPDL の概要を知っておくことは、どんな設定がされているのか知る上で重要です。ここでは、SPDL の概要を具体例と共に見ていきます。 具体例としては、図 6 の Apache 用ポリシを使います。

### 8.2.1 アプリケーションにドメインを付与する

2 行目と 3 行目は、アプリケーションにドメインを付与する設定です。2 行目は、ドメインの命名です。「httpd_t」というドメインを命名しています。以下、{} 内

に記述される設定は、httpd_t ドメインに対するものになります。デフォルトでは、ドメインは何もアクセス権限を与えられません。明示的にドメインに権限を与える設定を記述していくことで、設定を行っていきます。
３行目は、実際にアプリケーションにドメインを付与する設定です。アプリケーションの実行ファイル (/usr/sbin/httpd) を指定し、ドメインを付与します。これにより、/usr/sbin/httpd が実行されると同時にドメイン httpd_t が割り当てられるようになります。

- 上級者向けメモ
  ドメインを付与するために、SELinux のドメイン遷移が使われています。2 行目と 3 行目では、unconfined ドメインが、/usr/sbin/httpd を実行すると、/usr/sbin/httpd に httpd_t ドメインが割り当てられる、という設定がされます。SELinux のポリシーの書式で書くと次のようになります。

  `domain_auto_trans(unconfined_domain, /usr/sbin/httpd のタイプ, httpd_t)`
  `* unconfined_domain` は、`unconfined` ドメインに付与される属性

  unconfined ドメインじゃないドメインから /usr/sbin/httpd が実行された場合は、/usr/sbin/httpd にドメインが割り当てられないことに注意が必要です。

### 8.2.2 典型的な設定を使いまわす:include 文

4,5,6 行目で、一般的なアプリケーションで共通して使う設定を挿入しています。include 書式を使うと、他のファイルに記述された設定を挿入することができます。実際に、どんな設定が挿入されるかは、/etc/seedit/policy/include ディレクトリ以下を見れば分かります。例えば、 *include include/nameservice.sp;* では、/etc/seedit/policy/include/nameservice.sp に記述された設定が挿入されます。/etc/hosts への読み込み権限などが許可されます。

### 8.2.3 ファイルへのアクセスを許可する:allow 文

7 から 10 行目では、ファイルへのアクセスを許可しています。allow という書式を使って、ファイル名とパーミッションが記述されています。ファイル名については、次のような一括指定記法が使えます。

ディレクトリ名`/*` :ディレクトリ以下のファイル全て。サブディレクトリは含まない。
ディレクトリ名`/**`: ディレクトリ以下のファイルをサブディレクトリも含め全て。
例えば、`/etc/*`とした場合は、`/etc` 直下のファイルが指定され、
`/etc/sysconfig/network` など、サブディレクトリのファイルは含まれません。
`/etc/**`とすると、サブディレクトリにあるファイルも含まれます。

~ から始まるファイル名は、ホームディレクトリ（/root 以外）を表します。

`~/public_html/**`

は、各ユーザのホームディレクトリの下にある public_html ディレクトリ以下のファイル全てを表します。

パーミッションとしては、以下のパーミッションを使うことができます。

- 基本パーミッション
  - s
    Search の略です。ファイルツリーをサーチする、という意図で作られました。ディレクトリにあるファイル一覧を取得する権限、および、カレントディレクトリに設定する権限が設定されます。ファイルに対してこのパーミッションを設定しても何も意味はありません。
  - r
    Read の略です。ファイルを読み込む権限が設定されます。
  - x
    Execute の略です。ファイルを実行する権限が設定されます。
  - w
    Write の略です。ファイルを上書き、追記する権限や、ファイル・ディレクトリを生成消去する権限が設定されます。
- 詳細設定パーミッション
  w パーミッションは、多くの権限が設定されます。本当に必要最小限の権限を設定するために、w を分割した５つのパーミッションを利用できます。
  - a
    Append の略です。ファイルを追記オープンする権限が設定されます。
  - o
    Overwrite の略です。ファイルを上書き保存する権限が設定されます。
  - c
    Create の略です。ファイルやディレクトリを新規作成する権限が設定されます。
  - e
    Erase の略です。ファイルやディレクトリを消去する権限が設定されます。
  - t
    Setattr の略です。ファイルやディレクトリの属性を変更する権限が設定されます。属性とは、ファイルの所有者, 最終更新時刻など，ファイルに関する情報のことです。ファイルのセキュリティ属性 (SELinux のラベル) の変更は許可されません。

さて、これで、7-9 行目の設定の意味が理解できます。

- 7 行目: http_t ドメインが、/var/www 以下のファイル一覧取得可能、ファイル (サブディレクトリにあるファイル含む) の読み込みを可能です
- 8 行目: http_t は、/var/log/httpd 以下 (サブディレクトリ含む) のファイル一覧取得可能、およびファイルを読み込み，追記可能です。

- 9 行目 httpd_t は、/etc にあるファイル一覧のみを取得可能です。/etc 以下のファイルに対しては何もできません。/etc 以下のファイルにアクセスさせたい場合は、例えば、/etc/*や/etc/**などと記述する必要があります。

#### 8.2.4 ネットワークアクセス制御を設定:allownet

ネットワークに関連するアクセス制御も可能です。ポート番号を使ってサーバーとして振る舞う権限、クライアントとしてポートに接続する権限を設定可能です[1]。10 行目では、httpd_t ドメインが TCP80,443 ポートを使ってサーバーとして振る舞う権限をあたえられています。もし、MySQL サーバー (TCP 3306) に接続したいなら、以下のように設定します。allownet -protocol tcp -port 3306 client;
ポート番号の指定としては、 -1023, 1024-, * という表記が可能です。-1023 は全ての Wellknown ポート (他ドメインで使っているポートは除く) です。1024-は、1024 以上のポート番号 (他ドメインで使っているポートは除く)、*は全てのポート番号を意味します。

#### 8.2.5 他の特権を設定する:allowpriv

ファイルやネットワークに関連しない操作も、SELinux によって制限されています。*allowpriv* 特権名*;*. という書式で、設定できます。例えば、11 行目では、netlink ソケット (カーネルと通信するために使われる) の利用許可を与えてます。

#### 8.2.6 Unconfinedドメインにする

allowpriv all; と設定することで、そのドメインが unconfined ドメインになります。

```
{
domain httpd_t;
program /usr/sbin/httpd;
allowpriv all;
}
```

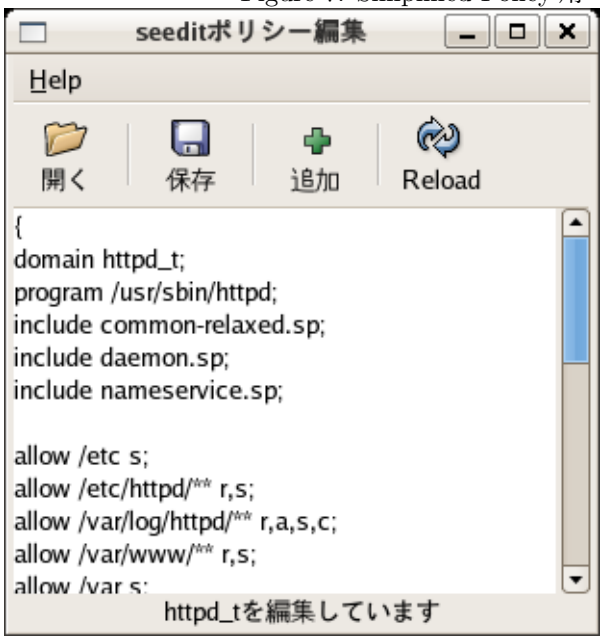
このようにすると、httpd_t は、unconfined ドメインとして扱われます。
/etc/selinux/seedit/policy/unconfined_domains にも、httpd_t が追加されてます。

### 8.3 GUI エディタ

コントロールパネルから ポリシーを編集 を選択すると、エディタが開きます。開く を選択し、ドメインを選択します。例えば、httpd_t ドメインを選択すると、図 7 のような画面になります。テキストエディタのように、ポリシーを編集できます。
保存 ボタンを押すと、編集内容を保存し、設定を反映します。
*Reload* ボタンを押すと、再度ファイルからの設定内容を読み出します。これは、他のツール（ポリシ生成ツール等）で、設定内容が変更された場合に便利です。

[1] ネットワークインターフェースや IP アドレスの利用制御も可能ですが、これらは common-relaxed.sp でデフォルト許可されています

Figure 7: Simplified Policy 用エディタ




追加 ボタンを押すと、ポリシーを挿入するためのウィンドウが開きます (図 8, 9)。
ファイル タブから、ファイルアクセス制御設定を挿入できます。図 8 の例では、
追加ボタンを押すと、

```
allow /var/www/** r,s;
```

という設定が末尾に追加されます。
ネットワークタブからは、ネットワークに関する設定を挿入できます。図 9 では、
追加ボタンを押すと

```
allownet -protocol tcp -port 80 server;
```

が挿入されます。

Figure 8: ファイルアクセス制御設定を追加

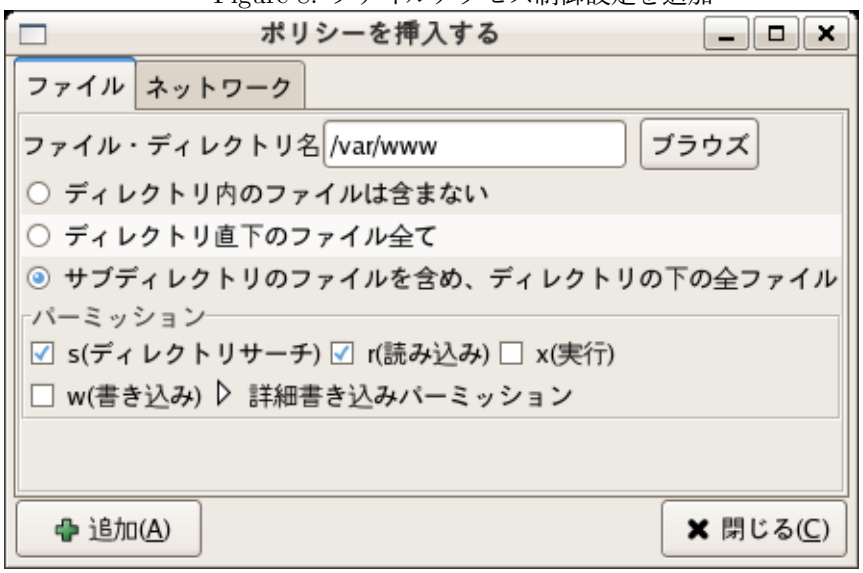
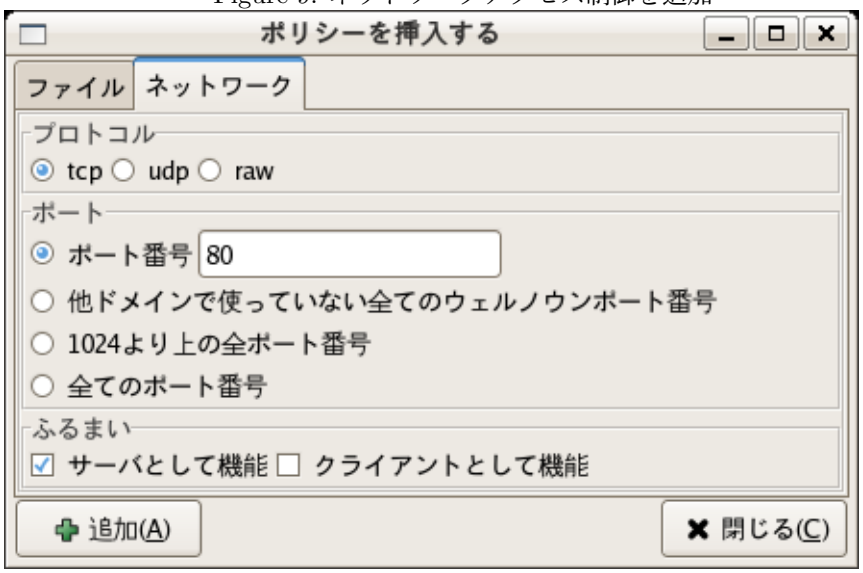



Figure 9: ネットワークアクセス制御を追加



# 9 ポリシーを追加する

## 9.1 Permissive モードで問題切り分け

SELinux のアクセス拒否のためにアプリケーションが動作しない、と判明したらポリシーを追加する必要があります。
ポリシーを追加する前に、permissive モードで動作確認を行います。もし、permissive モードでアプリケーションが動作したならば、SELinux が原因でアプリケーションが動作しないことになります。この場合は、ポリシーを追加してやる必要があります。SELinux Policy Editor では、このような場合にポリシーを生成するためのツールが GUI およびコマンドラインで用意されています。より具体的な例は、新規ドメイン作成とともに、10 章にて紹介します。

## 9.2 ポリシ生成の仕組み

ポリシの生成は SELinux のログを基にして行われます。 Permissive モードでの動作確認の間に, SELinux のログが取得されます。Permissive モードとは、アクセスが SELinux によって拒否される場合、ログを取るだけのモードでした。(SELinux が原因で動作しない) アプリケーションを動作させるには、拒否されるアクセスを許可してやる必要があります。
以下は、アクセス拒否ログの例です。このログを例に、ポリシ生成ツールがどうやってポリシを生成するのかを見ていきます。

```
----
time->Wed Apr 26 18:34:32 2006
1: type=PATH msg=audit(1146090872.442:29): item=0
name="/etc/vsftpd/vsftpd.conf" flags=101  inode=584775 dev=03:03
mode=0100600 ouid=0 ogid=0 rdev=00:00
type=CWD msg=audit(1146090872.442:29):
cwd="/etc/selinux/seedit/src/policy/simplified_policy"
2: type=SYSCALL msg=audit(1146090872.442:29): arch=40000003
syscall=5 success=yes exit=3 a0=bfb04c52 a1=8800 a2=0 a3=8800
items=1 pid=13151 auid=4294967295 uid=0 gid=0 euid=0 suid=0
fsuid=0 egid=0 sgid=0 fsgid=0 comm="vsftpd" exe="/usr/sbin/vsftpd"
3: type=AVC msg=audit(1146090872.442:29): avc:  denied  { read }
for  pid=13151 comm="vsftpd" name="vsftpd.conf" dev=hda3
ino=584775 scontext=user_u:system_r:ftpd_t
tcontext=system_u:object_r:default_t tclass=file
----
```

３行目は SELinux のアクセス拒否ログで，「 ftpd_t ドメインが、vsftpd.conf という名前のファイルを読もうとして拒否された」 という意味です．３行目から、以下のような設定を追加する必要があることが分かります。

```
allow vsftpd.conf r;
```

これだけでは、vsftpd.conf のフルパスを記述してないので、アクセスを許可できてません。しかし、３行目のログにはフルパス情報は書かれていません。 フル

Figure 10: ポリシ生成ツール


パスを得るために、1 行目のログを利用します。ここに、vsftpd.conf のフルパスは/etc/vsftpd/vsftpd.conf と書かれています。1 行目と 3 行目を組み合わせることで、以下を追加すればいいことが分かります。

```
allow /etc/vsftpd/vsftpd.conf r;
```

auditd サービスを起動していない場合、1 行目のログは取得されません。つまり、フルパス情報がログに含まれないことを意味します。auditd が動いてない場合は、locate コマンドを使ってフルパスを推測しますが、失敗することも多いです。ですので、SELinux Policy Editor を使うときは auditd を使うことをお薦めします。

## 9.3 GUI(ポリシ生成ツール)

### 9.3.1 ツールを起動

コントロールパネルより ポリシーを生成をクリックします。図 10 のような画面が開きます。

これは、ポリシ自動生成のための設定画面です。通常はデフォルトのままでかまいません。「ポリシーの生成」ボタンを押すことで、ポリシーが生成されます。なお、ここで設定可能な項目を参考のために解説します。

- 入力
  これは、SELinux のアクセスログが記録される場所を設定します。auditd が動いている時は、audit.log を選びます。そうでない場合は、dmesg を選択します。自分でファイル名を指定することもできます。
- セキュリティの高い設定を生成
  これがチェックされると、ポリシ生成ツールは、よりセキュリティの高い設

定を生成しようとします。現バージョンでは、詳細書き込みパーミッション (a,o,c,e,t) を使う設定を生成します。これをチェックしない場合は、w パーミッションしか使いません。

- 全てのログを読み込み
  これをチェックすると、選択したログから全てのメッセージを入力として読み込みます。これをチェックしない場合は、最後の設定反映後 (load_policy が許可された後) のログのみを読み込みます。

- Skip search log
  これをチェックすると，dir:search パーミッションのアクセス拒否ログをスキップします。このパーミッションは余計な設定生成につながりやすいからです。

#### 9.3.2 結果を確認し、設定を実際に追加

設定を生成　ボタンを押すと、ポリシーが生成されます。　しばらく時間がかかります。設定生成が完了すると、「結果」タブに結果が表示されます。
図 11 は、生成された結果の例です。

最初の行では、 *allownet -protocol tcp -port 1024- server;* という設定を vsftpd_t ドメインに追加するかを聞いています。対応するアクセス拒否ログも一緒に表示されています。このポリシーを追加したい場合は、チェックボックスにチェックを入れます。

まとめて指定 ボタンは便利な機能です。ディレクトリ単位でまとめてアクセス許可をすることができます。
*allow /etc/vsftpd/vsftpd.conf r,s;* を選択し、「まとめて指定」ボタンを押していくと、以下のようにファイル名が変わっていきます。

```
/etc/vsftpd/vsftpd.conf ->
/etc/vsftpd/* ->
/etc/vsftpd/** ->
/etc/* ->
/etc/** ->
/* ->
/** ->
```

まとめて指定を戻すボタンを押すと、上と逆の順番でファイル名が変わっていきます。
チェックされたポリシーは、 以下の設定が保存されますの部分に表示されます (図 12)。設定を追加する前に、権限を与えすぎでないか今一度見直します。大丈夫そうならば、 セーブし、設定を適用ボタンを押します。これらのポリシーがファイルに記述されるだけでなく、設定内容も SELinux 側に反映されます。

### 9.4 コマンド ライン (audit2spdl コマンド )

audit2spdl というコマンドを使うことでも、ポリシー生成を行うことができます。使いかたは、以下のようになります。auditd サービスが起動していない場合

Figure 11: ポリシ生成結果

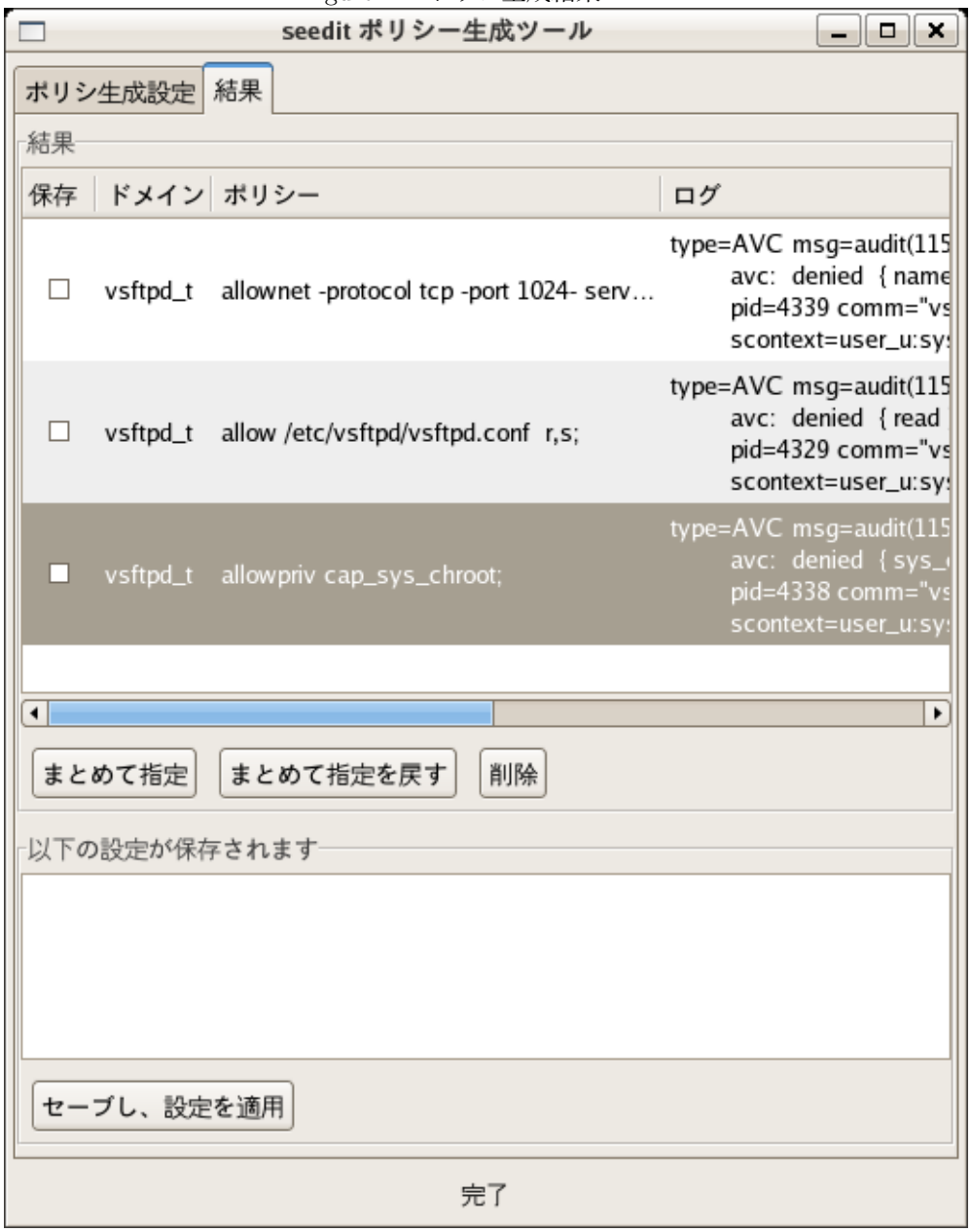
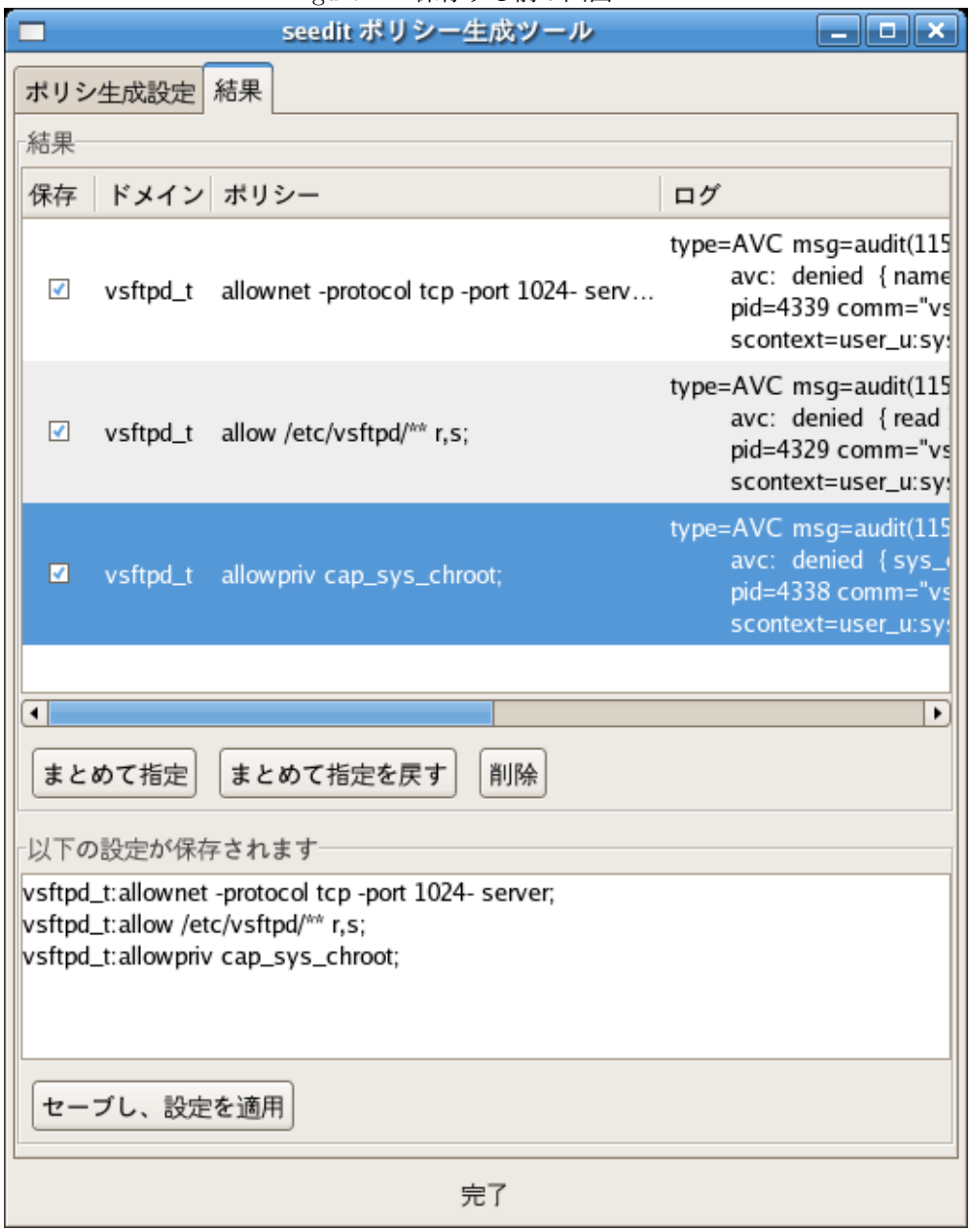


Figure 12: 保存する前の画面



(Fedora Core5 のデフォルト),

```
# audit2spdl -dl
```

When auditd is running(Fedora Core4 default)

```
# audit2spdl -al
```

You can read log by specifying filename,

```
# audit2spdl -l -i /var/log/messages
```

audit2spdl コマンドは、SELinux のアクセス拒否ログを、Simplified Policy に変換するものです。これにより、アプリケーションを動作させるために必要な権限を簡単に追加できます。以下が、出力例です。

```
#audit2spdl -al
.... It takes some time...
-------------------------
#SELinux deny log:
audit(1146162965.963:16): avc:  denied  { read } for  pid=6653
comm="vsftpd" name="vsftpd.conf" dev=hda3 ino=584775
scontext=user_u:system_r:ftpd_t
tcontext=system_u:object_r:default_t tclass=file
#Suggested configuration
File ftpd_t.sp:
allow /etc/vsftpd/vsftpd.conf  r;
-------------------------
...
```

これは、vsftpd.conf に read アクセス拒否をされたログが出ていますが、このアクセスを許可するには、

```
allow /etc/vsftpd/vsftpd.conf  r;
```

という設定を ftpd_t.sp ファイルに追加する必要がある、と言っています。

### 9.4.1 audit2spdl を効率よく使うために

auditd サービスを有効にしておいたほうが、効率よく作業ができます。auditd サービスはより詳細なログを取ってくれるからです。SELinux のアクセス拒否ログには、フルパス情報が含まれていません。例えば/etc/vsftpd.conf にアクセス拒否された場合、「vsftpd.conf」にアクセス拒否されたとしか記録されません。auditd サービスを使うと、フルパスも記録されます。アクセス拒否ログから、ポリシーを生成するには，フルパス情報が不可欠です。auditd サービスを有効にすることで、フルパス情報を得られ、正しいポリシーが生成できるわけです。
Fedora Core5 の場合、以下で auditd を有効にできます。

```
#yum install audit
#chkconfig auditd on
#/etc/init.d/auditd start
```

auditd を有効にしたら、通常は「audit2spdl -al」で事足ります。audit2spdl -dl は意味がなくなるので注意が必要です。
auditd を使わない場合、audit2spdl は、フルパス情報を「locate コマンド」と組み合わせて推測します。ただし、locate コマンドを使う前に定期的に「updatedb」コマンドを走らせる必要があります（デフォルトでは cron ジョブで走っています）。

### 9.4.2 提示されたポリシーを追加して設定反映

上の例では、必要なポリシーが提示されただけで、設定が反映されていません。/etc/seedit/policy/vsftpd.sp. を開き、「allow /etc/vsftpd/vsftpd.conf r;」という行を の間に記述します。次のようになります。

```
{
domain vsftpd\_t
program /usr/sbin/program;
allow ....
<ここに追加！>
}
```

ポリシーを追加したら、その変更内容を反映する必要があります。「seedit-load」とタイプします。

```
#seedit-load
seedit-load: Success
```

このコマンドは、Simplified Policy を SELinux のポリシの形式に変換し、/etc/selinux/seedit 以下に生成されたポリシーをインストールします。そして、そのポリシーをカーネルに読み込ませ、必要次第でファイルのラベル付けもします。
seedit-load コマンドの詳細な進行具合を「-v」オプションを付与することで見ることができます。

```
# seedit-load -v
mkdir -p ./sepolicy;
m4 -s ./simplified_policy/*.sp >./simplified_policy/all.sp;
/usr/bin/seedit-converter -i ./simplified_policy/all.sp -o
./sepolicy -b ./base_policy -I ./simplified_policy/include ;
.................

cp /etc/selinux/seedit/contexts/files/file_contexts.all
/etc/selinux/seedit/contexts/files/file_contexts.all.old
seedit-load: Success
```

なお、上の例では、

```
allow /etc/vsftpd/* r;.
```

という設定を追加してもよいです。これは、ファイル名が、audit2spdl に提示されたものとは異なります。が、システム管理者は、/etc/vsftpd ディレクトリが vsftpd の設定ファイルであることを知っていますので、このディレクトリ全体の読み込み権限を与えてしまったほうが効率がよいです。

### 9.4.3 audit2spdl についての諸注意

(1) セキュリティ上最適とは限らない
audit2spdl によって提示される設定は、最適なものとは限りません。提示された設定を追加する前に、注意深く見直す必要があります。例えば、audit2spdl のデフォルトは、詳細設定パーミッション (o,a,c,e,t)) を提示しません。代わりに w を提示しますので、設定が粗くなります。ちなみに、詳細設定パーミッションを提示するには、s オプションを例えば audit2spdl -als のように使います。

(2) 設定の提示に失敗することがある
ログからファイルのフルパスを得ることを失敗すると、以下のようなメッセージが現れます。

```
#Failed to generate, because failed to obtain fullpath.
```

SELinux のログにはフルパス情報は含まれてませんが、audit2spdl はフルパスを得るため様々な作業を行っています。それにも関らず失敗することもあります。auditd サービスを起動しておくことで、フルパスを得る可能性を高めることができます。

## 9.5 ポリシ生成の諸注意

- chroot しているアプリケーションではフルパスがうまく生成されないことも
例えば、vsftpd は、/var/ftp に chroot します。/var/ftp 以下のファイルにアクセスし、ポリシ生成をした時、allow /pub r,s; のようなポリシが生成されますが、ファイル名が正しくありません。実際は、/var/ftp/pub と記述しなくてはいけません。

# 10 新たにド メインを作成する

seedit でのポリシーの書き方を覚える最も良い方法は、新たなド メインを作成する設定を試すことです。vsftp FTP サーバに ftpd_t ド メインを付与する実例を元に、見ていきましょう。
新たなド メインを作成、設定する場合の手順は一般的に次のようになります。

(1) テンプレートとなる設定を作成

(2) ド メインが正しく付与されることを確認

(3) テスト動作とポリシーの追加

vsftpd をインストールしていない場合は、以下でインストールします。

```
#yum install vsftpd
```

## 10.1 GUIから新規ドメインを設定

vsftp_t ドメインを作成し、Anonymous FTP サーバとして動作するのに必要な設定をします。

### 10.1.1 テンプレートを作成

ドメイン管理 GUI を使うことで、新規ドメインを設定できます。コントロールパネルから、ドメインを管理を選択します。
vsftpd に vsftpd_t ドメインを割り当てます。図 13 のように設定します。
まず、実行ファイルの名前を指定します。「/usr/sbin/vsftpd」を入力します。
ドメイン名として、「vsftpd_t」を入力します。
あとは、いくつかの質問に答えます。デーモン用のドメインを作成するには、「デーモンプログラムですか？」の「はい」を忘れないようにします。

テンプレートを作成ボタンを押すと、作成されたテンプレートに設定が生成されます。

### 10.1.2 アクセスすると分かっている設定を追加

アプリケーションについての知識がある場合、あらかじめどんなアクセス権限が必要か分かっていることも多いです (分かってない場合は、後からポリシ生成ツールで追加できます)。その場合、この画面から追加可能です。

今回は、Anonymous FTP サーバを構築しますので、公開ディレクトリ（/var/ftp 以下）への読み込み権限が必要ですし、21 番ポートを使う権限が必要です。
これらの設定は「ポリシーを追加」ボタンを使って追加できます。図 14 では、/var/ftp ディレクトリ以下への読み込みアクセスを設定してます。図 15 では,TCP21 番ポートを使ってサーバーとして振る舞うことを許可しています。追加 ボタンを押すと、

```
allow  /var/ftp/** r,s;
allownet -protocol tcp -port 21 server;
```

のような設定が追加されます。追加したい設定を全て追加したら、「セーブし、設定を適用」ボタンを押します。

Figure 13: 新規ドメイン作成画面

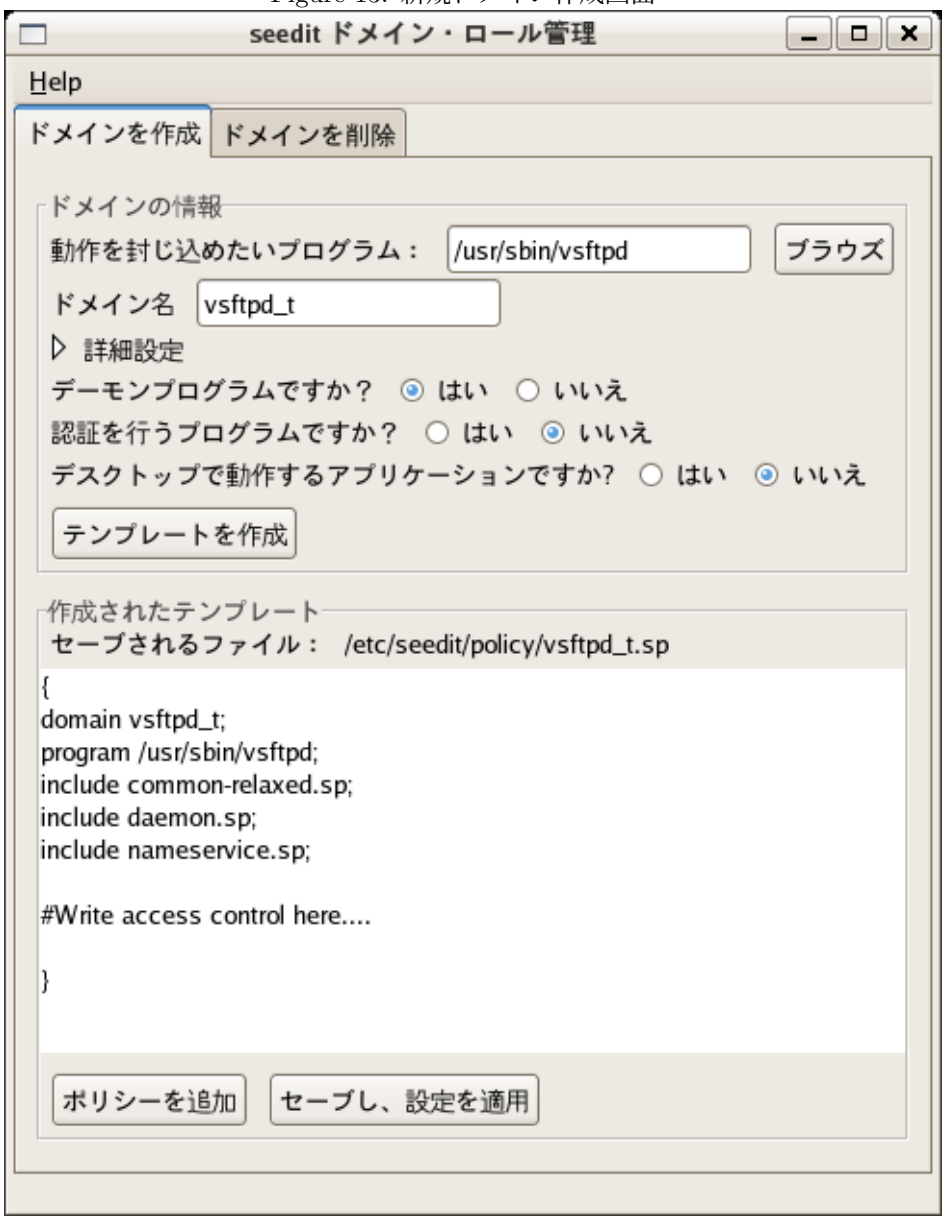


Figure 14: ファイルアクセス制御の設定追加

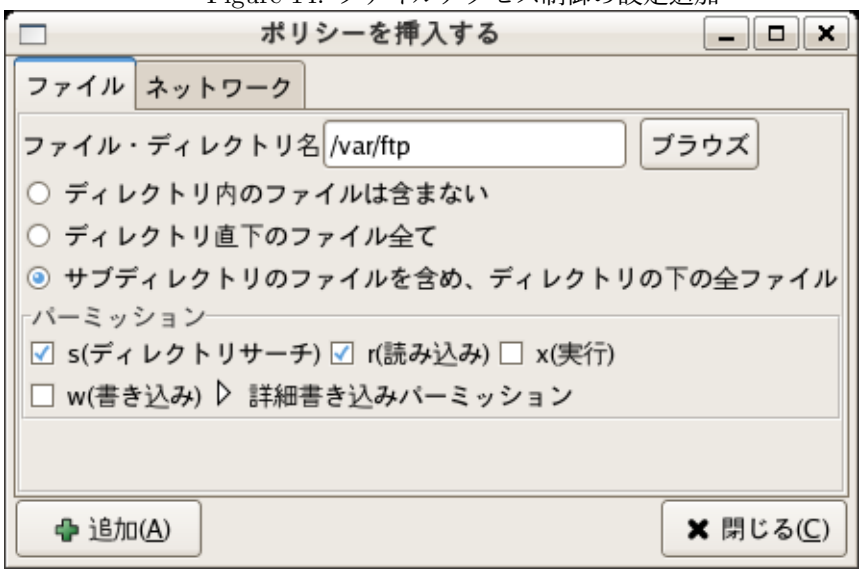
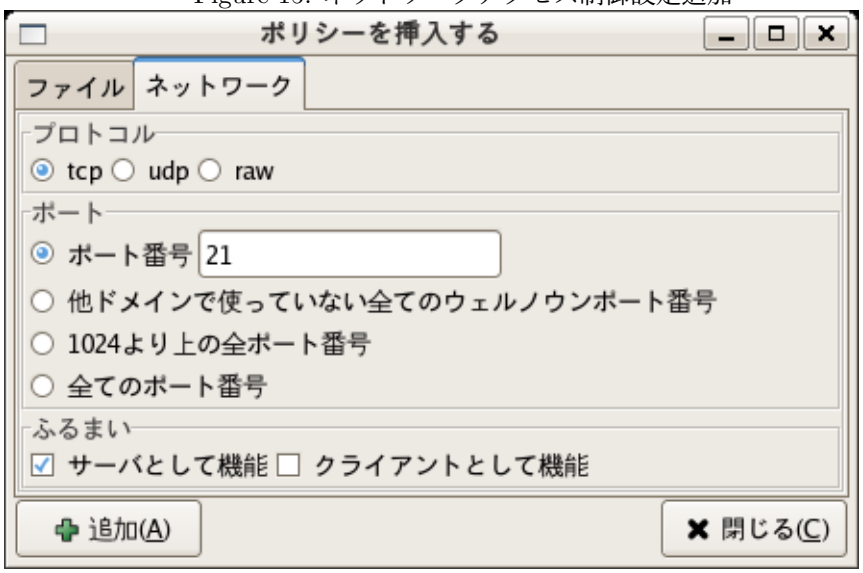



Figure 15: ネットワークアクセス制御設定追加



### 10.1.3 ドメインの確認

Permissive モードに切り替え、vsftpd を起動し、そのドメインが「vsftpd_t」であることをステータスツールなどで確認します。

### 10.1.4 テスト動作とポリシーの追加

vsftpd を permissive モードでテスト動作させます。テスト動作を通じて、必要な権限を洗い出すためです。

以下のコマンドでは, Anonymous ftp でファイル転送できることを確認してます。

```
#echo "test">/var/ftp/pub/test.txt
```
(テスト用のファイル作成)

```
$ ftp localhost
Name (localhost:ynakam): Anonymous
Password: <なんでも OK>
> cd /var/ftp/pub
> get test.txt
```

ここで、ログを見てみると（auserch -m AVC で見ることが出来ます)、様々なアクセスログが出ていることが分かります。ポリシー生成ツールでこれらを許可する設定を追加していきます。ポリシーを生成 をコントロールパネルから選択すると、図 16 のような画面が開きます。通常は、デフォルトの設定のまま「ポリシーを生成」ボタンを押します。図 17 のように結果画面が開きます。追加したい設定についてはチェックボタンを押します。ディレクトリ丸ごとアクセス許可をしたい場合は「まとめて指定」を使います。今回の場合は、/etc/vsftpd ディレクトリにまとめて設定を許可したほうが効率がいいので、/etc/vsftpd/vsftpd.conf のところで、「まとめて指定」ボタンを使います。ファイル名は、 /etc/vsftpd/vsftpd.conf → /etc/vsftpd/* → /etc/vsftpd/**のように変わっていきます

さて、今回の例では図 18 のような設定が追加されることになります。 セーブし設定を適用 を押して設定を反映します。
再度 vsftpd の動作確認をして、ポリシ生成ツールを使います。何も設定が生成されなくなったら、次は Enforcing モードでテストします。もし動作すれば、設定はおしまいです。そうでなければ、生成ツールでポリシを追加し、再度テストします。テスト、ポリシの生成をアプリケーションが Enforcing モードで動作するまで繰り返します。

Figure 16: Policy Generate tool

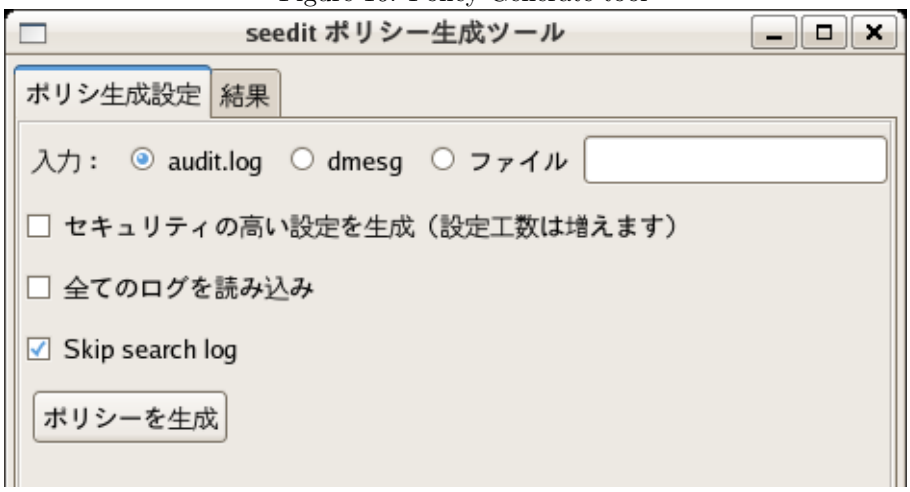


Figure 17: Policy Generate result

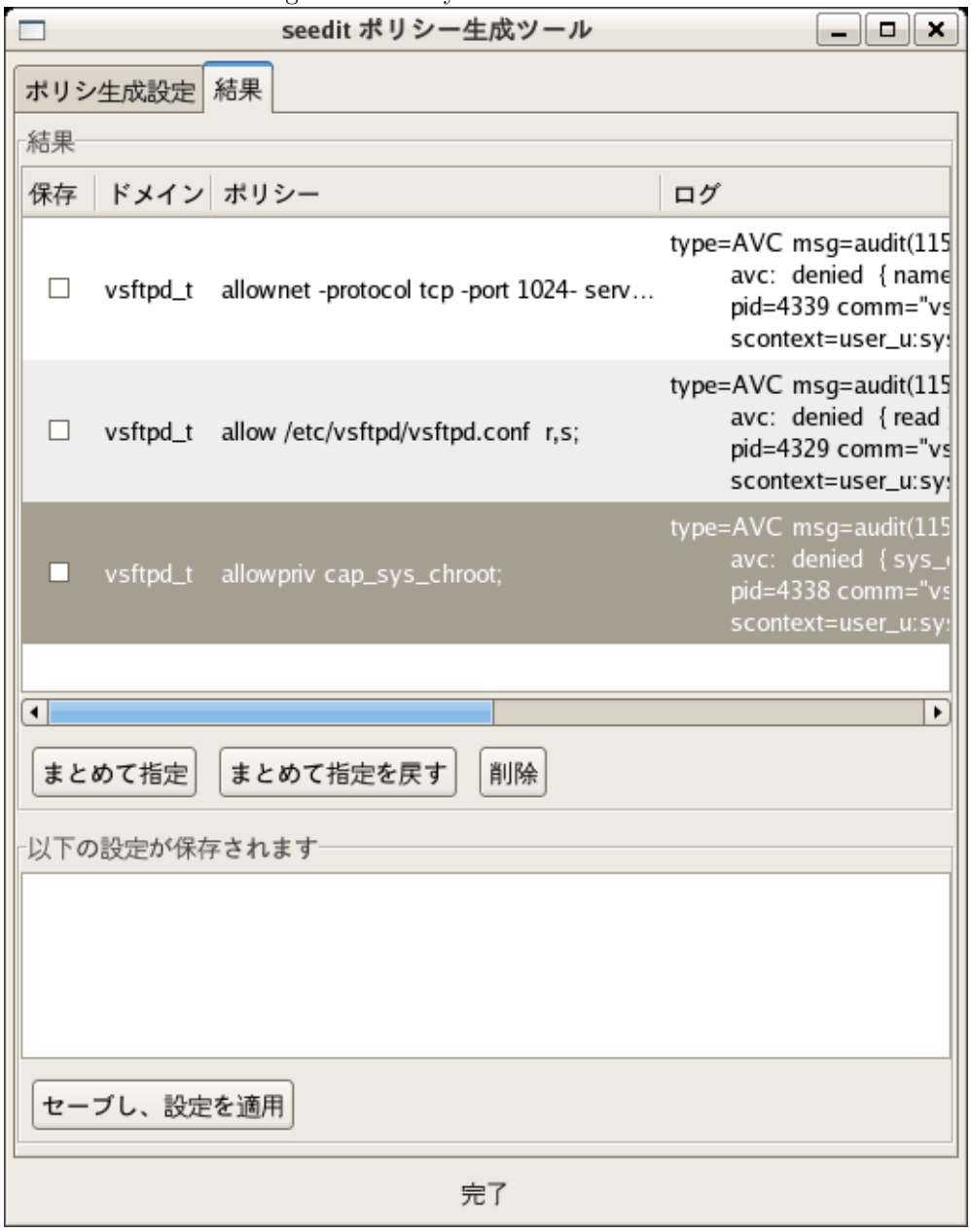
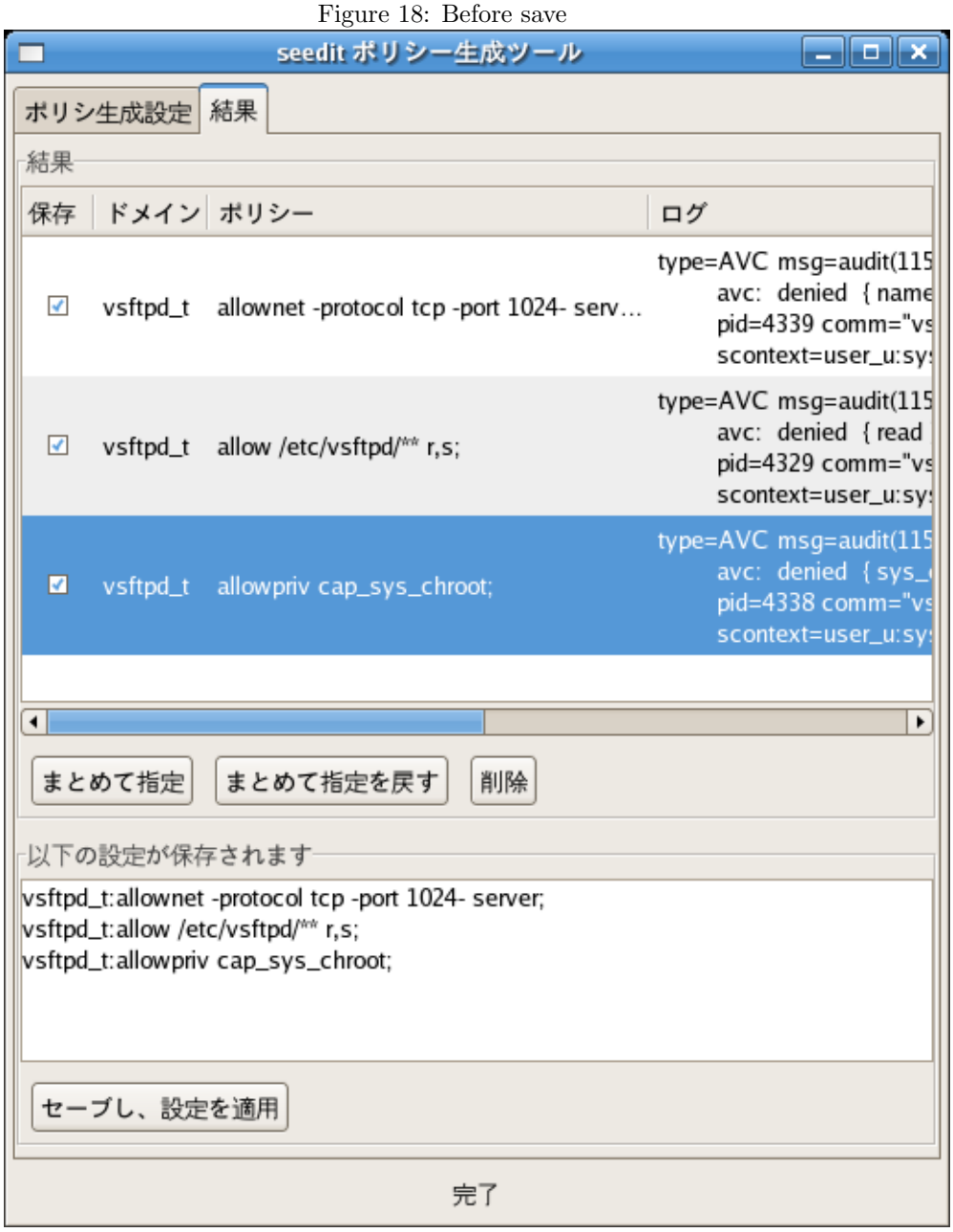


Figure 18: Before save



## 10.2 コマンド ラインからド メイン作成

### 10.2.1 テンプレート となる設定を作成

seedit-template コマンド を使うことで、テンプレートとなる設定を作成できます。書式は以下です。

```
seedit-template -d <ド メインの名前> -e <実行ファ イル名> -o <出力>
```

今回は、ド メインの名前は「ftpd_t」で、実行ファイルは「/usr/sbin/vsftpd」ですので、以下のように実行し ます。コマンド と実行結果を示し ます。

```
# seedit-template -d vsftpd_t -e /usr/sbin/vsftpd
{
domain ftpd_t;
program /usr/sbin/vsftpd;
include common-relaxed.sp;
include daemon.sp;
include nameservice.sp;
}
```

テンプレートとなる設定が生成されています。/usr/sbin/vsftpd に ftpd_t を付与しています。*include* から始まる行で、一般的なデーモンプログラムに必要な権限が与えられます。この設定を、/etc/seedit/policy/ftpd_t.sp に保存し ます。ファイル名は、必ず「ド メイン名.sp」である必要があります。さもないと、設定反映時にエラーが出ます。

### 10.2.2 ド メインが正しく付与されることを確認

設定の反映には、次のコマンド を使います。

```
#seedit-load
```

なお「-v」をつけると詳細な経過を見ることができます。
また、permissive モード に切り替える必要があります。ftpd_t にアクセス許可する設定をしていませんので、vsftpd が起動しようとしても enforcing モード ではアクセス拒否のため失敗するからです。

```
#setenforce 0
#getenforce
Permissive
```

vftpd を起動して、seedit-unconfined コマンド で vsftpd のド メインを確認し ます。

```
# /etc/init.d/vsftpd restart
# seedit-unconfined -e
10530   vsftpd  Confined by ftpd_t
```

上のように vsftpd に ftpd_t ド メインが付与されていることと思います。

### 10.2.3 テスト動作とポリシーの追加

さて、次は permissive モードでアプリケーションをテスト動作させます。そして、audit2spdl でどんな権限が足りないのかを洗い出します。ここでは、vsftpd が Anonymous FTP サーバとして動作するのに必要な権限を洗い出すことにします。
テストのために、以下のように ftp サーバに Anonymous としてログインします。

```
# touch /var/ftp/pub/test.txt
↑　テスト用のファイルを作成
$ ftp localhost
Name (localhost:ynakam): Anonymous
Password: <anything is OK>
#以下、pub/test.txt をダウンロードしている
ftp> ls
ftp > cd pub
ftp > get test.txt
ftp > quit
```

ログを閲覧します。以下、auditd サービスが起動しているという前提で話を進めます。auditd サービスが起動していない場合は、audit2spdl -al を-dl に置き換えて読んでください。

```
#auserch -m AVC
↑/var/log/audit/audit.log のログを閲覧するコマンドです。
```

様々なアクセス拒否が出ていると思います。
audit2spdl を使ってみましょう。

```
#audit2spdl -al
-------------------------
#SELinux deny log:
audit(1146179470.043:86): avc:  denied  { search } for
pid=10904 comm="vsftpd" name="vsftpd" dev=hda3 ino=584772
scontext=user_u:system_r:ftpd_t
tcontext=system_u:object_r:etc_t tclass=dir
#Suggested configuration
File ftpd_t.sp:
allow /etc/vsftpd  s;
-------------------------
-------------------------

#SELinux deny log:
type=AVC msg=audit(1148486747.277:30): avc:  denied
{ append } for  pid=11761 comm="vsftpd" name="xferlog"
dev=hda3 ino=163412 scontext=user_u:system_r:ftpd_t
tcontext=system_u:object_r:var_log_t tclass=file
```

```
#Suggested configuration
File ftpd_t.sp:
allow /var/log/xferlog  w,r,s;
-------------------------
.....
```

様々な設定を追加しなければならないことが分かると思います。なお、以下のように「#」から始まる行が提示されることがあります。

```
-------------------------
#SELinux deny log:
type=AVC msg=audit(1148486754.718:36): avc:  denied  { lock } for
pid=11763 comm="vsftpd" name="test.txt" dev=hda3 ino=163311
scontext=user_u:system_r:ftpd_t tcontext=system_u:object_r:default_t
 tclass=file
#Suggested configuration
File ftpd_t.sp:
#Failed to generate, because failed to obtain fullpath.
#allow test.txt  r,s;
-------------------------
```

これは、audit2spdl が test.txt のフルパスを発見することに失敗したため、先頭に#がついています。しかし、test.txt のフルパスは、/var/ftp/pub/test.txt であることを読者は知っているので、

```
allow /var/ftp/pub/test.txt  r,s;
```

を追加すればいいことが分かります。筆者の環境では、以下のような設定が提示されました。

```
allow /etc/vsftpd  s;
allow /var/log/xferlog  w,r,s;
allow /var/ftp/pub  s;
allow /var/ftp/pub/test.txt  r,s;
allownet -protocol tcp -port 21 server;
allow /etc/vsftpd/vsftpd.conf  r,s;
allowpriv cap_sys_chroot;
allow /var/ftp  s;
allow /var/log/xferlog  r,s;
```

これをそのまま追加してもいいのですが、冗長な設定があります。例えば、/var/ftp、/etc/vsftpd はディレクトリ全体を読み込み許可すれば冗長な設定を削れます。冗長な設定を削ると、ftpd_t.sp は結局以下のようになります。

```
{
domain ftpd_t;
program /usr/sbin/vsftpd;
include common-relaxed.sp;
```

```
include daemon.sp;
include nameservice.sp;
# added by audit2spdl suggestion
allow /etc/vsftpd/**  r,s;
allow /var/ftp/**  r,s;
allow /var/log/xferlog  r,w,s;
allownet -protocol tcp -port 21 server;
allowpriv cap_sys_chroot;
}
```

さて、seedit-load コマンドで設定を反映し、vsftpd を再起動して、再度動作テストをしてみましょう。audit2spdl -al すると、さらに以下のような設定が提示されるかもしれません。

```
allownet -protocol tcp -port 6353 server;
```

これは、vsftpd が TCP 6553 ポートを使ってサーバーとして振る舞おうとしていることを言っています。が、ポート番号は毎回変わります。なので、以下のように 1024 以上のポートを使えるよう設定します。

```
allownet -protocol tcp -port 1024- server;
```

設定を反映し、さらにテスト動作を、アクセス拒否がでなくなるまで繰り返します。最後に enforcing モードにして、テストをして正しく動作することを確かめます。

```
#setenforce 1
```

さて、ここまでの加方法で、概ね設定できるかと思います。ログインユーザのセキュリティを高めるための機能として、SPDL は RBAC もサポートしています。RBAC については、「RBAC ガイド」を参照してください。

# 11　その他の注意点

(1) ファイルの移動
ファイル A を別のディレクトリ B に移動した場合，ディレクトリ B にアクセスできるドメインが、移動後の A にアクセスできなくなることがあります。ファイル移動時、「ファイル A に対するアクセス権限」を引き継ぐからです（詳しくは後述)。このようなトラブルを解決するには restorecon コマンドを使う必要があります。

* 具体例：ホームページのアップロードの場合
Apache が移動したファイルにアクセスできないことがあります

```
# pwd
/root/homepage/index.html
# mv index.html /var/www/html
# restorecon /var/www/html/index.html
```

ここで、最後の restorecon コマンドを忘れると、Apache は/var/www 以下にアクセスできるように設定されてたとしても、index.html にアクセスできません。なぜなら、httpd_t は、/root/index.html にアクセスできませんが、これがファイル移動後も継承されているからです。これを修正するには restorecon コマンドをする必要があります。これでも問題が解決できない場合は特殊な場合です。audit2spdl で設定を追加する必要があります。

(2) ファイルの新規作成
新規作成されたファイルのアクセス権限は、ディレクトリから引き継がれます。次の例を見てください。以下の設定が存在したとします。

```
domain foo_t;
allow /foo/bar/** r,s;
allow /foo/bar/test.txt r,w,s;
```

foo_t は、/foo/bar 以下に読み込み可能で、/foo/bar/test.txt に書き込み可能です。
/foo/bar/test.txt が、設定時に存在しなかったとし、設定後に test.txt に新規作成されたとします。foo_t は、/foo/bar/test.txt に読み込みアクセスしかできません。/foo/bar のアクセス権を引き継ぐからです。この状況を直すには、

```
restorecon -R /foo/bar
```

をする必要があります。

(3) restorecon はなぜ必要?
SELinux は、ファイルなどのリソースを「タイプ」を呼ばれるラベルで識別しています。タイプの名前は、ファイル名の「/」を「_」に置換した名前になっています。mv コマンドをするときは、ラベルは保存されます。ファイルが新規作成された時は、タイプはディレクトリのものを引き継ぎます。ファイルとタイプの関連付けを restorecon コマンドを使うことで直す必要があるのです。

(4) cron ジョブ
Cron ジョブは、unconfined なドメインで動きます。cron job のアクセス制御を行いたい場合は、system_crond_t.sp を編集し、*allowpriv all;* を削る必要があります。ただし、cron ジョブを正しくアクセス制御するのは難しいことに注意が必要です。

(5) 消去、生成が繰り返されるファイル
消去、生成が繰り返されるファイルについては、アクセス制御が思ったようにいかない場合があります。次の例を見てください。

```
domain foo_t;
allow /foo/bar/** r,s;
allow /foo/bar/test.txt r,w,s;
```

こう設定したとすると、test.txt が消去され、再度 test.txt が生成された場合、test.txt のアクセス権限は所属ディレクトリのものを継承しますから、foo_t は、test.txt を読み込みしかできません。restorecon コマンドでこれを直すことができるのですが、また test.txt が消去、生成されると、さらに restorecon をしなければなりません。
ディレクトリに対するアクセス権限と違うファイル、かつそれが消去、生成が繰り返される場合は、アクセス制御がうまく設定できないことになります。
これを解決するには以下のような方法があります。

(a) ファイル毎のアクセス制御をあきらめ、ディレクトリ単位のアクセス制御をする
つまり、allow /foo/var/** r,s,*w.* のように、/foo/var 全体に書き込み権限を与えてやれば問題ありません。しかし、書き込み可能な範囲が広がってしまいます。

(b) allowtmp を利用
このような、生成消去を繰り返すファイルのアクセス制御をするために、SPDL には allowtmp という設定要素が用意されています。今回の場合は次のように設定することになります。

```
allowtmp /foo/bar -name auto  r,w,s;
```

allowtmp を使うことで、ファイルを「ラベル」を使って識別できるようになります。上の文にて、foo_t が、/foo/bar 以下に作成したファイルは、foo_foo_bar_t という名前のラベル (-name auto で命名がされます。foo_t + /foo/bar = foo_foo_bar_t という規則です) が付与され、そして、foo_t は、このラベルに対し、r,w,s アクセス可能、ということになります。
このようにすることによって、test.txt が /foo/bar ディレクトリに消去、生成を繰り返されたとすると、test.txt には、foo_foo_bar_t というラベルが常に付与されます。そして、このラベルを使って SPDL から設定することができます。
例えば、他のドメインから、このラベルが付いたファイルにアクセスしたい場合は、

```
allow foo_foo_bar_t r;
```

のように書きます。ちなみに、allowtmp では、SELinux のファイルタイプ遷移が使われてます。
デフォルトで用意されているポリシーでは、allowtmp は、/etc/mtab のアクセス制御や、/tmp,/var/tmp 以下の一時ファイルのアクセス制御に使われています。

(6) /dev ディレクトリ以外のデバイスを使う
デバイスファイルは、システムに対して致命的な影響を及ぼすため、SPDL でも特に注意深く取り扱われています。デフォルトで用意されているポリシでは、デバイスは、/dev ディレクトリに存在すると設定されています。つ

まり、/dev ディレクトリ以外のデバイスに対して、allow でアクセス許可を設定しても、無視されてしまいます。
/dev ディレクトリ以外のデバイスを使いたい場合は、allowdev 文を使う必要があります。例えば、/var/chroot/dev/null にアクセスしたい場合は、/var/chroot/dev/null にアクセスする設定を記述する前に、

```
allowdev -root /var/chroot/dev;
```

のように、デバイスの格納されるディレクトリを指定しておく必要があります。

(7) シンボリックリンクを含むファイル名
シンボリックリンクを含むファイル名は無視されます。例えば、
allow /etc/init.d/httpd r;
は無視されます。init.d は、 rc.d/init.d のシンボリックリンクだからです。

(8) ハード リンクの扱い
Linux システムでは、ハード リンクを使うことで、ファイルの中身を複数のファイル名で参照することができます。ハード リンクは、デフォルトではほとんど使われていないため、以下の内容を気にする場面はほとんど現れませんが、セキュリティ上知っておいたほうがいいでしょう。
SPDL では、ハード リンクは以下のルールで処理されます。
ファイルの中身が複数のハード リンクで参照される場合、元々存在したファイル名を記述する必要がある。それ以外のファイル名が指定された場合は無視される。
例えば、/etc/shadow と /var/chroot/etc/shadow がハード リンクされていたとします。/etc/shadow が元々存在していたとすると、/etc/shadow (と /var/chroot/etc/shadow) の中身を見るためには、allow /etc/shadow r と記述する必要があります。allow /var/chroot/etc/shadow r と記述しても無視されます。
ハード リンクが複数存在する場合、どのファイル名を「元々存在するファイル名」とするかの基準が気になるところです。以下の基準で「元々存在するファイル名」が判定されます。以下で出てくる例では、/etc/shadow, /var/shadow がハード リンクされたファイルだと仮定します。

(a) 全ポリシ中で、一つのファイル名に対する設定しか書かれていない場合、そのファイル名が「元々存在するファイル名」になります
例: allow /etc/shadow r がある場所で記述されているとします。そして、/var/shadow を使った設定はどこにも記述されていないとします。この場合は、/etc/shadow が、元々存在するファイル名として取り扱われます。

(b) 複数のファイル名に対する設定が記述されていた場合、名前が一番若いものが、「元々存在するファイル名」になります。
例: allow /etc/shadow r,allow /var/shadow r; という設定が記述されていたとします。この場合、「/etc/shadow」が、「元々存在するファイ

ル名」になります。なぜなら、/etc/shadow のほうが名前が若いからです。

(c) ハードリンクされたファイル名を使った設定がどこにも記述されてないときは、所属ディレクトリ名を比較して、所属ディレクトリ名が最も大きいものが「元々存在するファイル名」となります。
例: /etc/shadow, /var/shadow を使った設定がどこにも記述されていない場合、/var/shadow が「元々あるファイル名」となります。なぜなら、/var/ > /etc だからです。

しかし、どのハードリンク名が「元々存在するファイル名」か分からない場合は、全ての名前を使う手もあります。例えば、

```
allow /etc/shadow r;
allow /var/shadow r;
```

のように記述した場合、どちらかの設定は無視されるだけで、無害です。
以上のハードリンクの取り扱いは、パス名ベースのセキュリティの「抜け穴」を防ぐために必要なものです。この取り扱いがなかったとすると、例えば、/etc/shadow のハードリンクが、なんらかの手で/var/www/html/shadow に作られてしまうと、Web サーバーから/etc/shadow の中身を覗けてしまうことになります。これを防ぐために、ハードリンクされたファイルの中身にアクセスするには、「一つのファイル名」しか使えないようにする必要があるわけです。パス名ベースのセキュリティの問題点については、
http://securityblog.org/brindle/2006/04/19
に詳しいです。

# 12 Tips

(1) Web アプリケーション (CGI) を安全に設定する

デフォルトでは、CGI は、httpd_t ドメイン (Apache Web サーバのドメイン) で動作します。これは、サブプログラムのドメインは、親プログラムのドメインと同じになるという SELinux の仕様によるものです。CGI のドメインを変えたい場合は、「domain_trans」というものを使って設定を記述する必要があります。具体例を解説します。
CGI スクリプトが、/var/www/cgi-bin にあると仮定し、cgi_t ドメインで走らせたいとします。cgi_t.sp を以下のように作ります。

```
{
domain cgi_t;
domain_trans  httpd_t /var/www/cgi-bin/**;
include common-relaxed.sp;
##### allowxxx will be here...
}
```

program 文は、unconfined ドメインから起動したプログラムにドメインを割り当てるものでした。domain_trans 文は任意のドメインから起動されたプログラムにドメインを割り当てるものです。

```
domain cgi_t;
domain_trans  httpd_t /var/www/cgi-bin/**;
```

は、httpd_t ドメインで動いているプログラムが、/var/www/cgi-bin 以下のプログラムを実行した場合、cgi_t ドメインを割り当てる、という意味になります。
なお、PHP については、httpd_t ドメイン以外で動作させることはできません。PHP は、通常の実行とは異なる方法で処理されているからです。

(2) deny
*deny* という設定要素を使うことで、重要なファイルをブラックリストに登録し、設定ミスの防止に利用することができます。以下の例を見てみましょう。

```
{
domain foo_t;
deny /etc/shadow;
allow /etc/** r,s;
}
```

foo_t が、/etc 以下の全ファイルに読み込みできるように設定されていますが、/etc/shadow にアクセスすることができません。/etc/shadow にアクセスするには、 *allow /etc/shadow* と明示的に記述する必要があります。なお、デフォルトでは include/common-relaxed.sp にいくつかのファイルが deny が登録されています。

# 13 質問をしたい

質問があれば、気軽に 中村 himainu-ynakam@miomio.jp までメールを下さい。
匿名で質問をしたい場合は、SELinux フォーラム (http://intrajp.no-ip.com/xoops/) を活用して下さい。