

# 単純化ポリシーによる SELinux の設定

中村 雄一\*  
和訳 山口 拓人

August 1, 2005

## Contents

1	この文書について	3
2	概要	3
3	単純化ポリシーの構成	3
4	注意	4
5	単純化ポリシーのサンプル	4
5.1	単純化ポリシーのデフォルト設定	4
5.2	単純化ポリシー用ディレクトリのコンテンツ	4
5.2.1	simplified_policy	4
5.2.2	Makefile	5
5.2.3	base_policy	5
5.2.4	macros	6
5.2.5	sepolicy	6
5.3	To load policy	6
6	単純化ポリシーコンパイラのオプション機能	7
7	例	7
7.1	vsftpd の設定	7
7.1.1	vsftpd 用ドメインの作成	8
7.1.2	ドメイン遷移テスト	8
7.1.3	vsftpd に関連したファイルの保護	9
7.1.4	vsftpd.t へのアクセス権限の付与	9
7.1.5	initrc.t へのアクセス権限の付与	11
7.1.6	再テスト	11

---

\*ジョージワシントン大学, 日立ソフトウェアエンジニアリング株式会社, ynakam@gwu.edu

<b>8</b>	<b>単純化ポリシー記述言語の仕様</b>	<b>11</b>
8.1	用語	11
8.1.1	Domain(ドメイン)/Role(ロール)/Global domain(globalドメイン)	11
8.2	デフォルト拒否の原則	12
8.3	単純化ポリシー記述言語による設定構造	12
8.4	セクションのシンタックス	12
8.5	ドメインとロールの宣言	12
8.5.1	ドメインの宣言	12
8.5.2	ロールの宣言	13
8.6	RBACの設定	13
8.6.1	user(ユーザ)	13
8.7	ドメイン遷移の設定	13
8.7.1	domain_trans	13
8.8	通常ファイルのアクセス制御設定	14
8.8.1	allow	14
8.8.2	deny	15
8.8.3	allowonly	15
8.8.4	denyonly	16
8.8.5	allow, allowonly, deny, denyonly の優先度	16
8.8.6	特殊ファイル	17
8.9	ネットワークのアクセス制御設定	17
8.9.1	allownet	17
8.10	プロセス間通信のアクセス制御設定	19
8.10.1	allowcom (ネットワークソケット)	19
8.10.2	allowcom (IPC)	19
8.10.3	allowcom(シグナル)	19
8.11	tty/pts デバイスのアクセス制御設定	20
8.11.1	allowtty	20
8.11.2	allowpts	20
8.12	/proc ファイルシステムのアクセス制御設定	21
8.12.1	allowproc	21
8.13	tmpfs のアクセス制御設定	21
8.13.1	allowtmpfs	21
8.14	他の管理上のアクセス権限のアクセス制御設定	22
8.14.1	allowadm	22
8.15	global とドメインにおける矛盾	23
<b>9</b>	<b>連絡先</b>	<b>23</b>
<b>10</b>	<b>TODO</b>	<b>23</b>

## 1 この文書について

この文書は、単純化ポリシー (バージョン 1.0.0) のリファレンスマニュアルです。インストールについては、インストール方法をお読み下さい。注：この文書は現在ツール自体が開発途中の段階であり、後に内容を変更することがあります。

## 2 概要

SELinux[1] は、粒度の高い強制アクセス制御を Linux 上に実装しています。しかし、アクセス制御の粒度の高さや、ポリシーが複雑になってきていることから、ポリシーの理解および設定することが非常に難しくなっています。「単純化ポリシー」は中間設定言語に書かれているポリシーです。中間設定言語はユーザからタイプラベルを隠すことによって、ポリシーの記述量を減らします。また、オブジェクトクラスとアクセスベクターの統合もしています。ユーザは SELinux システムを単純化ポリシーにより簡単に扱うことができます。例えば、`httpd_t` ドメインが `/var/www` 以下を読んだり、`tcp` ポート 80 番を使用したいとき、設定は次のようになります。

```
{
domain httpd_t;
allow /var/www r,s;
allownet -tcp -port 80;
```

単純化ポリシーは元々日立ソフトウェアエンジニアリング株式会社 [4] によって SELinux Policy Editor[2][3] の一部の機能として開発されてきたものです。現在 SELinux Policy Editor は SELinux Policy Editor Project[5] においてメンテナンスされています。また、本ツールは Fedora Core 4 と 3 に対応しています。単純化ポリシーは現在利用されている SELinux には影響を与えません。デフォルトの SELinux に簡単に戻ることができます。気軽に試してみてください。

## 3 単純化ポリシーの構成

単純化ポリシーは単純化ポリシーコンパイラとサンプルポリシーの2つが主な構成要素です。さらに、オプションとして GUI(SELinux Policy Editor) も構成されています。

1. 単純化ポリシーコンパイラ  
単純化ポリシーのコンパイラです。単純化ポリシーコンパイラは単純化ポリシーを読み込み、`m4` マクロと `checkpolicy` により SELinux ポリシーを理解しやすいものにして生成してくれます。
2. サンプルポリシー  
サンプルの単純化ポリシーです。
3. GUI(SELinux Policy Editor)(オプション)  
Web ベースの GUI で、単純化ポリシーを編集します。GUI を使用することで、SELinux をより簡単なものにします。また、GUI を使用した単純化ポリシーのことを *SELinux Policy Editor* と呼んでいます。

## 4 注意

- シンタックスが次のバージョンで変更されるかもしれません。単純化ポリシーのセキュリティについて再検討中であり、その結果変更はあるかもしれません。
- いくつかのパーミッションがサポートされていません。5.2.3を参照して下さい。
- いくつかのファイルシステムがサポートされていません。サポートされていないものにつきましては、`unlabeled_t`として取り扱っております。サポートされていないファイルシステムにアクセスするときは、`allowadm unlabeled`を利用して下さい。

## 5 単純化ポリシーのサンプル

この章では単純化ポリシーのサンプルについての説明をしていきます。お急ぎでしたら 5.1、5.2.1そして 5.3をお読み下さい。

### 5.1 単純化ポリシーのデフォルト設定

- バージョン 1.0.0においてサポートされているサービス  
サンプルポリシーにおいてサポートされているサービスは「`auditd`、`syslogd`、`httpd`、`webmin`、`iptables`、`network`」です。これらはサービス名`_t`ドメインとして動作します。
- RBAC 設定  
`sysadm_r`と`user_r`、`sysadm_r`というロールはあらゆることができます。それに対して、`user_r`ができることは限られています。デフォルトでは、`root`ユーザのみが`sysadm_r`としてログインできます。

### 5.2 単純化ポリシー用ディレクトリのコンテンツ

単純化ポリシーのサンプルは、「`/etc/selinux/seedit/src/policy/`」にあります。いくつかのディレクトリとファイルが配置されています。

#### 5.2.1 `simplified_policy`

これは最も重要な機能です。単純化ポリシーのサンプルは、このディレクトリに記録されています。単純化ポリシーのシンタックスの詳細については、8を御覧下さい。

単純化ポリシーのサンプルは、`global`とドメイン名`.te`において記述されています。

- `global`  
すべてのドメインに共通して使われる設定です。ただし、いくつかのアクセス権限はSELinuxを簡単なものにするためにデフォルトで許可されていることに注意して下さい。(例えば、`tmpfs`の使用や`tty`デバイスアクセスです。)

- ドメイン名.a  
ドメインの設定について説明します。例えば、`httpd_t.a` ファイルにおいては、`httpd_t` ドメインのための設定が記述されています。
- all  
global と全ての.a ファイルが連結されたものです。単純化ポリシコンパイラはこのファイルを読みます。このファイルは自動的に生成されるものであり、編集してはいけません。

### 5.2.2 Makefile

Makefile は単純化ポリシをコンパイルし、カーネルにポリシをロードします。5.3 を御覧ください。

### 5.2.3 base\_policy

このディレクトリにあるファイルは SELinux のポリシを生成するために単純化ポリシコンパイラによって使用されます。

- default.te  
このファイルは有用です。なぜなら、単純化ポリシコンパイラによって生成されるポリシに、このファイルの記述が含まれているからです。ファイルには、単純化ポリシにおいてデフォルトで許可されている allow 文が記述されています。従って、このファイルを見れば、どのアクセスベクターやオブジェクトクラスが単純化ポリシにサポートされていないのかがわかります。例を以下に示します。

```
allow global self:capability ~{ net_raw net_bind_service
net_admin sys_boot sys_module sys_rawio sys_ptrace sys_chroot };
```

生成された SELinux ポリシにおいて、global 属性が使われています。global 属性はすべてのドメインに属しています。よって、すべてのドメインは net\_raw net\_bind\_service net\_admin sys\_boot など、以上に示された 8 つ以外に関してはケイバビリティを使用することを許されています。つまり、以上のアクセスベクターは単純化ポリシでサポートされていないことを意味します。

さらに、このファイルにオリジナルの SELinux のルールを記述できます。auditallow ルールを書くのは良いアイデアだと思います。ただ、allow ルールはこのファイルに書いてはいけません。なぜなら生成されたポリシのセキュリティを侵害しうるからです。

- attribute.te  
このファイルには、属性が記述されています。属性は単純化ポリシコンパイラによって生成されるポリシにおいて使用されています。このファイルは編集してはいけません。

- `types.te`  
このファイルは、タイプが記述されています。タイプは単純化ポリシコンパイラによって生成されるポリシにおいて使用されています。以下のファイルはオリジナルバージョンの SELinux のポリシと同じ内容です。これらのファイルは編集してはいけません。

- `genfs_contexts`  
unlabeled\_t というタイプが付与されているファイルシステムはサポートしていません。
- `security_classes`
- `access_vectors`
- `initial_sid_contexts`
- `fs_use`
- `initial_sids`

#### 5.2.4 macros

SELinux のポリシを生成するマクロがおかれています。単純化ポリシコンパイラはマクロを含んだポリシを生成します。make コマンドを実行すると、マクロを含んだポリシが生成された後、m4 コマンドによって処理され、`policy.conf` が生成されます。この `policy.conf` は `checkpolicy` コマンドによってコンパイル可能なものです。

#### 5.2.5 sepolicy

生成された SELinux のポリシは、このディレクトリに書き込まれます。

- `test.conf`  
マクロを含んだポリシーです。
- `policy.conf` マクロを展開した後のポリシーです。checkpolicy でコンパイル可能です。
- `file_contexts`  
`file_contexts` ファイルです。

### 5.3 To load policy

`simplified_policy` にあるポリシに手を加えるなら、変更したポリシをカーネルにロードしなければなりません。Makefile にはいくつかのターゲットがあります。まず最初に、`/etc/selinux/seedit/src/policy` に移動して下さい。

- `make reload`  
このコマンドは `simplified_policy` ディレクトリにある単純化ポリシを SELinux のポリシに変更します。  
(1) `policy.conf` と `file_contexts` が `sepolicy` ディレクトリに生成されます。  
(2) バイナリポリシは生成された `policy.conf` から `checkpolicy` によって作成

されます。

(3) バイナリポリシーは/etc/seedit/policyにインストールされ、file\_contextsは/etc/seedit/contexts/policyにインストールされます。そしてbase\_policy/contextsの内容は/etc/seedit/contextsにインストールされます。

(4) バイナリポリシーはload\_policyによってロードされます。

- **make relabel**  
*make reload* コマンドが実行された後、*fixfiles restore* コマンドが実行されます。
- **make diffrelabel**  
このコマンドは非常に有用です。*make relabel* コマンドはすべてのファイルを再度ラベル付けします。そのため長い時間を要します。それに対して、*make diffrelabel* コマンドはラベルが変更されたファイルのみを再度ラベル付けします。ポリシーをロードする時には *make diffrelabel* コマンドを使用することを推奨します。

## 6 単純化ポリシーコンパイラのオプション機能

単純化ポリシーコンパイラ (*converter* コマンド) は *make* コマンドにより実行されます。ですので、*converter* コマンドを直接使う必要はありません。以下、参考のために示します。

Usage : *converter -i infile -b base policy dir -o policyfile -f file\_context*

- *infile*  
単純化ポリシーファイルです。*make* コマンド使用時に読み込まれるのは all です。all については 5.2 を御覧下さい。
- *base policy dir*  
base\_policy のディレクトリです。
- *policyfile*  
生成されたポリシーファイルの名前です。
- *file\_contexts*  
生成された file\_contexts ファイルの名前です。

## 7 例

### 7.1 vsftpd の設定

デーモンの設定例として、単純化ポリシーを用いて vsftpd を設定してみましょう。ここでは、Anonymous ftp アクセスができるまでのポリシーを設定することを目標にします。

デフォルトでは、vsftpd ドメインは *initrc.t* です。*initrc.t* は etc/rc.d 以下の起動スクリプト用ドメインです。vsftpd は、/etc/rc.d/init.d/vsftpd (ドメインは *initrc.t*)

によって実行され、ドメインを継承します。  
しかし、以上の動作は安全ではありません。なぜなら、`initrc_t`は多くのアクセス権限を持っているからです。(`/etc/selinux/seedit/src/policy/simplified_policy/initrc.t.a`を御覧ください)。

以下、カレントディレクトリは`/etc/selinux/seedit/src/policy`であり、`permissive`モードに切替えていることを前提とします。

```
login: root
....
# newrole -r sysadm_r
# id -Z
root:sysadm_r:sysadm_t
# cd /etc/selinux/seedit/src/policy
# setenforce 0
```

また、シンタックスの詳細については、8を御覧ください。

### 7.1.1 vsftpd 用ドメインの作成

それでは vsftpd に `vsftpd_t` ドメインを与えてみましょう。

1. 設定ファイルの作成  
`simplified_policy/vsftpd.t.a` を作成します。
2. ドメイン遷移の設定  
`simplified_policy/vsftpd.t.a` に以下に示す内容を記述して下さい。

```
# simplified_policy/vsftpd.t.a
{
domain vsftpd_t;
domain_trans initrc_t /usr/sbin/vsftpd;
}
```

2行目で `vsftpd_t` ドメインを定義しています。3行目でドメイン遷移を設定しています。ここで、親ドメインは `initrc_t` であり、エントリポイントは `/usr/sbin/vsftpd` です。

### 7.1.2 ドメイン遷移テスト

設定を編集したら、`make` コマンドを実行しなければなりません。これは設定の変更をカーネルに伝えるためです (5.3を参照して下さい)。この場合、以下の様にタイプします。

```
# make diffrelabel
```

多くの場合、`make diffrelabel` で十分です。  
そして、vsftpd を再起動させて vsftpd のドメインをチェックします。



```
# /etc/init.d/vsftpd restart
# ps -eZ
...
root:system_r:vsftpd_t          13621 pts/1    00:00:00 vsftpd
...
```

以上のようにすると vsftpd ドメインが *vsftpd\_t* になったことを確認できます。ドメイン遷移の設定がうまくいったことが分かりました。

### 7.1.3 vsftpd に関連したファイルの保護

vsftpd に関連したファイルの保護を行います。vsftpd ドメインに関連したファイルを保護したいなら、global において *deny* するのが一番良い方法です。この方法を用いて */etc/vsftpd* と */var/ftp* を保護してみましょう。simplified\_policy/global に以下に示す文を加えて下さい。{ と } の間に書き加えなければならぬことに注意して下さい。

```
# In simplified_policy/global
deny /etc/vsftpd;
deny /var/ftp;
```

そして以下のコマンドを実行して下さい。

```
# make diffrelabel
```

その結果、他のドメインが */etc/vsftpd* や */var/ftp* にアクセスしようとするなら、明示的にアクセスを許可しなければなりません。

例: *httpd\_t* が */etc/vsftpd* を読みたい場合、*allow /etc/vsftpd r;* という文が *httpd\_t* に記述されていなければなりません。 *allow /etc r;* では、 */etc/vsftpd* はアクセス許可されてません。重要なファイルを区別するには *deny* は有用です。

### 7.1.4 vsftpd\_t へのアクセス権限の付与

vsftpd\_t 用のデフォルトのアクセス権限は、simplified\_policy/global から継承されています。ただ、それだけでは十分でなく、設定を追加しなければなりません。最も良い方法は、permissive モードで vsftpd をテストして SELinux のログを参照し、どのようなアクセス権限が必要であるのかを見極めることです。(この文章では過程については割愛しています。) 以下、vsftpd\_t 用ポリシです。

```
# simplified_policy/vsftpd_t.a
1 {
2  domain vsftpd_t;
3  domain_trans initrc_t /usr/sbin/vsftpd;
4  # access to files related to vsftpd
5  allow /etc/vsftpd r,s;
6  allow /var/ftp r,s;
7  allowonly /var/log r,w,s;
8  # allow to communicate with syslog
9  allow dev_log_t r,w,s;
```

```

10 allowcom -unix syslogd_t;
11 # allow to use tcp 20 and 21
12 allownet;
13 allownet -connect;
14 allownet -tcp -port 20;
15 allownet -tcp -port 21;
16 #
17 allowadm chroot;
18 }

```

以上を記述した後に、以下のコマンドを実行して下さい。

```
# make diffrelabel
```

上の設定ファイルの中身を見てみましょう。

- 5行目から7行目  
これらは vsftpd に関連するファイルにアクセスするための設定です。5行目と6行目で、vsftpd 設定ファイルと ftp の public ディレクトリを読むアクセス権限を与えています。7行目に注目して下さい。

```
allowonly /var/log r,w,s;
```

ここで、/var/log/xferlog に書き込み権限を与えたい場合、本来は以下の様にするのが最適です。

```
allow /var/log/xferlog r,w,s;
```

しかし、/var/log/xferlog は管理者によって消去されるかもしれません。ファイルが再度生成された場合ラベル情報が失われてしまいます。<sup>1</sup>よって、/var/log/xferlog のアクセスを制御できません。そのため、*allowonly /var/log r,w,s* としているのです。vsftpd\_t は /var/log 以下にあるすべてのファイルに書き込み権限を与えています。しかし、子ディレクトリにあるファイルには書き込み権限を与えていません。これは、*allow /var/log r,w,s;* よりは良いです。(この文は子ディレクトリも含めて /var/log 以下の全てのファイルに書き込みアクセス権限を与えることを意味しています)。/tmp、/var/run も同様にファイル単位でのアクセス制御ができません。これらのディレクトリではファイルが消去、再生成されるため、SELinux のラベル情報が失われる可能性があるからです。

SELinux について十分な知識があるなら *allow file exclusive label;* が使えます。これは SELinux のファイルのタイプ遷移を設定します。消去、再生成されるファイルのアクセス制御設定ができます。詳細については 8.8.1 を御覧下さい。

---

<sup>1</sup>これは SELinux の実装上の問題です。ファイル単位でアクセス制御する場合、ファイルの i ノード番号が変わると親ディレクトリと同じアクセス制御設定になってしまいます。この場合にファイル単位でアクセス制御するにはファイルタイプ遷移を使う必要があります。

- 9、10 行目  
syslogd で通信するためにあります。syslogd を用いて通信する場合は、この 2 行を必ず入れて下さい。
- 12 行目から 15 行目  
tcp20 番ポートと 21 番ポートを使って通信します。

### 7.1.5 initrc\_t へのアクセス権限の付与

initrc\_t は起動スクリプト (/etc/init.d/vsftpd) 用のタイプです。initrc\_t は /etc/vsftpd へ読み込み権限を必要とします。しかし、このファイルへのアクセスは global において拒否されています。よって、明示的に allow しなければなりません。

```
#add to simplified_policy/initrc_t.a
allow /etc/vsftpd r,s;
```

以上を書き加えたら、以下のコマンドを実行して下さい。

```
# make diffrelabel
```

### 7.1.6 再テスト

permissive モードにおいてテストをして、アクセスログを参照して下さい。deny が出力されていないければ、次に enforcing モードでも正常に動作するか確かめて下さい。

## 8 単純化ポリシー記述言語の仕様

単純化ポリシーは、単純化ポリシー記述言語によって記述されています。ここでは、単純化ポリシー記述言語の仕様を示します。

### 8.1 用語

#### 8.1.1 Domain(ドメイン)/Role(ロール)/Global domain(globalドメイン)

- Domain(ドメイン)  
ドメインは SELinux におけるドメインと同じものです。ドメイン遷移によってプロセスに付与されています。
- Role(ロール)  
単純化ポリシー記述言語におけるロールは単純化されています。ロールはユーザシェル用ドメインと同一視されています。単純化ポリシー記述言語において、ロール用アクセス権限を記述します。事実、ロールのユーザシェルに対するアクセス権限を与えています。例えば、*sysadm\_r* にアクセス権限を与える時、アクセス権限は *sysadm\_t* に対して与えられます (*sysadm\_r* のユーザシェル用ドメイン)。  
生成された SELinux のポリシーでは、すべてのロールがすべてのタイプを使うことができるようになっています。単純化ポリシー記述言語に *role::x:types:y* に対応するシンタックスはありません。

- *global domain*(global ドメイン)  
*global* という名前のドメインは特別なドメインです。global ドメインに記述された設定は、すべてのドメインによって継承されます。例えば、global ドメインにおいて/etcに読み込み権限を与えると、*httpd.t*、*sendmail.t* などすべてのドメインが/etcを読み込めるようになります。

## 8.2 デフォルト 拒否の原則

デフォルトでは、設定を記述しない限り、ドメインとロールはすべてのパーミッションを拒否されています。

## 8.3 単純化ポリシー記述言語による設定構造

設定はセクションによって構成されています。各々のセクションにおいて、ドメインとロール用のアクセス制御が記述されています。セクションは{に始まり、}に終わります。

## 8.4 セクションのシンタックス

```
{ (セクションの始まり)
domain/role (ドメインもしくはロールを宣言します。1つのセクションにつき、
ドメインもしくはロールを1つ宣言できます。)
users (ロールによって使用されます。)
domain.trans (ドメイン遷移を設定します。)
allow/deny (ファイルのアクセス制御を記述します。)
allow:xxx (ファイル以外のリソースのアクセス制御を記述します。)
}
```

## 8.5 ドメインとロールの宣言

### 8.5.1 ドメインの宣言

1. シンタックス  
`domain domainname ;`
2. 意味  
ドメインを宣言します。セクションの中では、この文によって宣言されたドメインに対して、設定がなされます。*domainname*が*global*であった場合は、そのセクションにおける設定は、他のすべてのドメインによって継承されます。
3. 制約  
ドメインの名前は必ず*.t*で終わらなければなりません (*global*は除きます)。この文は1つのセクションにつき1度しか使用できません。

### 8.5.2 ロールの宣言

1. シンタックス  
role *rolename* ;
2. 意味  
ロールを宣言します。 *rolename* は以下で示す *user* 文を使用することによって *user* に関連付けられます。
3. 制約  
*rolename* は *\_r* で終わらなければなりません。

## 8.6 RBAC の設定

### 8.6.1 user(ユーザ)

1. シンタックス  
user *user name*;
2. 意味  
ユーザがどのロールを使用できるか宣言します。
3. 例  
{  
role *user\_r*;  
user *root*;  
user *ynakam*;  
....  
以上の文は、 *root* と *ynakam* が *user\_t* を使用できるということを意味しています。
4. 制約  
これはロールが宣言されているセクションでのみ使用されます。

## 8.7 ドメイン遷移の設定

### 8.7.1 domain\_trans

1. シンタックス  
domain\_trans *parentdomain filename-of-entrypoint*;
2. 意味  
これはドメインがどのようにプロセスと関連するかを宣言します。
3. 例  
{  
domain *httpd\_t*;  
domain\_trans *initrc\_t /sbin/httpd*;  
....  
以上の文は、プロセス (ドメイン: *initrc\_t*) が */sbin/httpd* を実行したとき、*/sbin/httpd* が *httpd\_t* ドメインで走行することを意味しています。

## 8.8 通常ファイルのアクセス制御設定

### 8.8.1 allow

#### 1. シンタックス

- (a) `allow filename | label [r],[w],[x],[s];`
- (b) `allow directoryname exclusive label;`

#### 2. 意味

- (a) ファイルへのアクセス権限を設定する
- (b) これは SELinux の `file_type_auto_trans` 相当の機能です。 `directoryname` 以下に作成されるファイルは `label` によって識別されます。そのようなファイルにアクセスを許可するには、 `allow label [r],[w],[x],[s]` を使います。消去、再生成されるファイル (例: `/etc/mtab`) を保護したいのならば、この機能を使用します。 `label` は SELinux においてタイプと同じようなものです。  
単純化ポリシコンパイラが `label` と名付けられたタイプを有するファイル (ファイル名は A と仮定) を見付けた時、以下の文が、生成される `file_contexts` に含まれます。

```
A system_u:object_r:label
```

ファイルが設定時に存在せず、消去、再生成されるファイルを保護したい場合、有用な方法です。例えば、 `/var/run`、 `/tmp`、 `/var/log` 以下のファイルが考えられます。

#### 3. パーMISSIONの意味<sup>2</sup>

- r: 読み込みと属性の閲覧
- w: 書き込み
- x: 実行
- s: ディレクトリの場合、ディレクトリの内容を参照  
ファイルの場合、属性の閲覧

#### 4. 例

```
{  
domain httpd_t;  
...  
allow /var/www r,s;  
....  
httpd_t は /var/www 以下の全てのファイルとディレクトリの読み込み権限  
を持つ。
```

---

<sup>2</sup>[6] において問題が指摘されており、パーMISSION種別については再検討する予定です。

### 8.8.2 deny

1. シンタックス  
`deny filename;`
2. 意味  
通常のドメインでは、これは `allow` をキャンセルするために使われます。  
globalドメインでは、明示的にアクセスを拒否するために使います。以下の例を御覧下さい。
3. 例
  - (a) 例 1  

```
{
domain httpd_t;
...
allow /var r,s;
deny /var/named; ...
```

`httpd_t` は `/var` の読み込み権限があるが、`/var/named` を読み込む権限は拒否されている。
  - (b) 例 2  

```
{
domain global;
deny /etc/shadow
...
{
domain httpd_t;
...
allow /etc r,s;
...

```

`httpd_t` は `/etc` 以下を読み込む権限があるが、`/etc/shadow` を読み込む権限は拒否されている。`/etc/shadow` へのアクセスは、`global` において拒否されているからである。重要なファイルがあれば、`global` において `deny` 文を記述すると良いです。

### 8.8.3 allowonly

1. シンタックス  
`allowonly directory name [r],[w],[x],[s];`
2. 意味  
`allow` においてアクセス権限はすべてのサブディレクトリに継承されます。それに対して、`allowonly` では、ディレクトリ直下のファイルへのアクセスは許可されているが、サブディレクトリに対しては許可されていません。
3. 例  

```
{
domain httpd_t;
```

```
...
allowonly /etc r,s;
```

```
....
```

httpd\_t は/etc 以下の読み込み権限を持っているが、/etc/httpd のようなサブディレクトリへのアクセス権限はない。

#### 8.8.4 denyonly

1. シンタックス  
denyonly *directory name*;
2. 意味  
deny アクセスがディレクトリ直下のファイルに設定されているが、サブディレクトリには設定されていない。

#### 8.8.5 allow, allowonly, deny, denyonly の優先度

1. 同じディレクトリに対して設定された場合、global ドメインにおいて *allow(deny), allowonly(denyonly)* は、通常ドメインにおける *allow(deny), allowonly(denyonly)* によって上書きされる。
  - 例 1)  
global:allow /usr/ r;  
a\_t:domain:allowonly /usr/ w;  
a\_t は /usr 以下に書き込み権限を持つ。
2. 子ディレクトリに allow もしくは deny が存在した場合、親ディレクトリの allow を上書きする。
  - 例)  
a\_t の中:  
allow /usr r;  
allow /usr/local w;  
a\_t は can read under サブディレクトリを含めて /usr 以下を読み込む権限を持っている。しかし、/usr/local 以下については書き込み権限を持っている (読み込み権限は持っていない)。
3. 同じドメインかつ同じディレクトリにおける allow もしくは deny OR 演算が取られます。
  - 例 1)  
{  
domain httpd\_t;  
allow /var/www r;  
allow /var/www w;  
httpd\_t は /var/www 以下に読み込み (r) 権限と書き込み (w) 権限を持つ。



- 例 2)
 

```
{
domain httpd_t;
allow /var/www r;
deny /var/www;
httpd_t は/var/www にはアクセスできない。
```

4. global ドメインに関する補足  
global ドメインにおいて allow/deny をキャンセルするためには、明示的に allow/deny を記さなければなりません。

- 例 1)
 

```
In global: deny /etc/shadow;
a_t ドメインに/etc/shadow を読ませたいなら、allow a_t /etc/shadow
r; を記さなければなりません。
```

- 例 2)
 

```
In global:allow /usr/local r;
In a_t: allow /usr w;
a_t は/usr/local 以下に書き込むことができません。/usr/local に書き
込みたいなら、a_t ドメインの中で allow /usr/local w; を記さなけれ
ばなりません。
```

- 例 3)
 

```
global: allowonly /usr/local r;
a_t: allow /usr w;
a_t は allowonly により /usr/local を読み込みます。
```

5. ドメインがどのファイルにアクセスできるのか知るには GUI が有用です。

### 8.8.6 特殊ファイル

以下のファイルへのアクセスは特殊になっています。

1. /dev/tty\* /dev/pts /dev/ptmx  
これらのファイルに allow 文を書いても何も起こりません。allowtty や allowpts によってアクセス制御がかけられているからです。
2. /proc, /sys, /selinux, /dev/tmpfs  
これらのファイルを allow しても何も起こりません。なぜなら、xattr をサポートしていないファイルシステムによって、これらのファイルはマウントされているからです。/proc と /sys については allowproc を参照して下さい。/dev/tmpfs については allowtmpfs を、/selinux については allowadm を参照して下さい。

## 8.9 ネットワークのアクセス制御設定

### 8.9.1 allownet

1. シンタックス

- (a) allownet;
- (b) allownet -connect;
- (c) allownet -raw;
- (d) allownet -netlink;
- (e) allownet -wellknown;
- (f) allownet (-tcp|-udp) -port *port number*;
- (g) allownet (-tcp|-udp) -allport;

## 2. 意味

これらはネットワークの取り扱いに関連しています。

- (a) tcp/ip ネットワークの使用を許可します。これは tcp、udp ソケット、1024 番以上のポートの使用を含んでいます。ネットワーク接続を開始することは許可されていないことに注意して下さい。ネットワーク接続を許可するには、allownet -connect を使用して下さい。ウエルノウンポートの使用は許可されていません。
- (b) ネットワーク接続を許可します。SELinux における *name\_connect* と *connect* パーミッションを使用することを意味します。
- (c) raw ソケットの使用を許可します。ICMP を使う場合、raw ソケットの使用が必要になります。
- (d) netlink ソケットの使用を許可します (カーネルとの通信に使用されます)。
- (e) このパーミッションは現在使用されていません。よって、何も動作しません。互換性を保つために残されているだけです。
- (f) ウェルノウンポートを使用したい場合、これによってポート番号を予約します。

- 例)

```
{
domain httpd_t;
allownet -tcp 80;
```

...

httpd\_t は tcp80 番ポートを予約し、使用できるようにしています。

- (g) 予約されていないウエルノウンポートの使用を許可します。

## 3. 制約

これらは一度宣言されると取り消すことができません。global ドメインで設定する場合は、慎重に設定して下さい。global ドメインでこれらを使用する場合、すべてのドメインにアクセス権限が明記され、各々のドメインで deny できなくなります。

## 8.10 プロセス間通信のアクセス制御設定

### 8.10.1 allowcom (ネットワークソケット)

1. シンタックス  
allowcom -tcp|-udp|-unix *todomain*;
2. 意味  
プロセス間通信においてソケットの使用を制御します。 *todomain* が *global* の場合、ドメインはすべてのドメインと通信できます。
3. 例  
{  
domain httpd\_t;  
allowcom -unix syslogd\_t;  
...  
httpd\_t は unix ドメインソケットによって、syslogd\_t を持つプロセスと通信できることを意味します。
4. 制約  
-tcp と -udp はカーネル 2.6 ベースの SELinux において使用できません。明記しても何も起こりません。

### 8.10.2 allowcom (IPC)

1. シンタックス  
allowcom -sem|-msg|-msgq|-shm|-pipe *to domain* [r],[w];
2. 意味  
指定された IPC により、 *to domain* を用いての通信を許可する。  
*to domain* が *self*(自分自身) の場合は、そのドメイン内の IPC を意味します。また、 *to domain* が *global* の場合は、すべてのドメインに対して IPC の設定ができます。

### 8.10.3 allowcom(シグナル)

1. シンタックス  
allowcom -sig *to domain* [c],[k],[s],[o];
2. 意味  
*to domain* へのシグナル送信を許可します。 [c] は sigchld、 [k] は sigkill、 [s] は sigstop、 [o] は他のシグナルを表しています。ただし、 signull についてはサポートがなされていません。すなわち、 signull についてはすべてのドメインにおいて使用が許可されていることを意味しています。

## 8.11 tty/pts デバイスのアクセス制御設定

### 8.11.1 allowtty

allowtty は tty デバイスファイル (/dev/tty\*) のアクセス制御を行うために使用されます。SELinux 環境において、tty デバイスファイルはログインしたユーザのロールによって異なったラベルが与えられています。そのため、tty デバイスファイルは単純化ポリシ記述言語において異なった取り扱いがなされます。

#### 1. シンタックス

- (a) allowtty -create;
- (b) allowtty role [r],[w];
- (c) allowtty -change role;

#### 2. 意味

- (a) これは通常ロールセクションにおいて使用されます。ロールに自身の tty デバイスを持たせることを許可します。ログイン時に、ログインプログラムによって、そのロールの tty デバイスファイルのタイプ *role prefix\_tty\_device\_t* が与えられます。
- (b) ロールの tty デバイスに読み書き権限を与えます。
- (c) tty デバイスのラベルの変更、名前の変更、消去を許可します。

#### 3. 特殊ロール

*role* が *general* であった場合、ラベル付け前の tty デバイス (タイプは *devtty\_t* と *tty\_device\_t*) へのアクセスを許可します。*role* が *global* であった場合は、すべての tty デバイスを意味します。

### 8.11.2 allowpts

allowpts は擬似 tty デバイスファイル (/dev/pts) のアクセス制御に使用されます。/dev/pts 以下のデバイスはリモートログインや gdm からのログイン用の端末です。

#### 1. シンタックス

- (a) allowpts -create;
- (b) allowpts role [r],[w];
- (c) allowpts -change role;

#### 2. 意味

意味はターゲットが擬似 tty デバイスであること以外は allowtty と同様の使い方です。

## 8.12 /proc ファイルシステムのアクセス制御設定

### 8.12.1 allowproc

allowprocによって procfs と sysfs のアクセス制御を記述できます。

1. シンタックス  
allowproc -self|-other|-system|-kmsg [r],[w];
2. 意味
  - (a) -self  
/proc/*pid*のアクセス制御です。自ドメインに関連した/proc/*pid*へのアクセスを許可します。
  - (b) -other  
他のドメインプロセス用の/proc/*pid*を意味します。
  - (c) -system  
/proc/kcoreと/sysfsです。
  - (d) -kmsg  
/proc/kmsgです。
  - (e) r,w  
rは読み込み、wは書き込みです。ドメインは proc/sysfs ファイル以上を読み込み、もしくは書き込みできます。

## 8.13 tmpfs のアクセス制御設定

### 8.13.1 allowtmpfs

1. シンタックス
  - (a) allowtmpfs -create;
  - (b) allowtmpfs *domain* r|w;
2. 意味
  - (a) tmpfsにおいてドメイン自身のファイルを作成することを許可します。作成されたファイルは、*domain\_prefix*.tmpfs.t というタイプを持ち、allowtmpfsの(b)によって明示的に記述がなされていない限り、他のドメインからアクセスされません。
  - (b) *domain*のtmpfsファイルへのアクセスを許可します。*domain*が *general*の場合、デフォルトのtmpfsのタイプ(tmpfs.t)へのアクセスを許可します。*domain*が *global*の場合、すべてのtmpfsへのアクセスを許可します。

## 8.14 他の管理上のアクセス権限のアクセス制御設定

### 8.14.1 allowadm

#### 1. シンタックス

```
allowadm [relabel][part_relabel][getsecurity][setenforce][load_policy][net][boot]
[[insmod][quotaon][swapon] [mount][raw_io][ptrace][chroot][search] [unlabel]]
[read] | [write] | [all];
```

#### 2. 意味

##### (a) relabel

すべてのファイルを relabel することを許可する。この時、getsecurity も allow しなければなりません。

##### (b) part\_relabel

そのドメインが書き込みできるファイルについて relabel することを許可する。この時、getsecurity を allow しなければなりません。

##### (c) getsecurity

/selinux にアクセスすることによって、SELinux のアクセス制御情報を得ることを許可する。

##### (d) setenforce

enforcing モードもしくは permissive モードの切り替えを許可します。

##### (e) load\_policy

カーネルに対してポリシーのロードを許可します。

##### (f) net

*CAP\_NET\_ADMIN*(例: NIC 管理、ルーティングテーブル管理) ケイパビリティを許可します。

##### (g) boot

*CAP\_SYS\_BOOT* ケイパビリティを許可します。システムコール `reboot` の使用を許可することを意味します。ただし、このケイパビリティを持っていなくとも、`reboot` コマンドは使えてしまいます。`reboot` コマンドによるリブートを制限する場合は、`/dev/initctl` へのアクセスを制御します。

##### (h) insmod

*CAP\_SYS\_MODULE* ケイパビリティを許可します。カーネルモジュールのインストールを許可することを意味します。

##### (i) quotaon

quotaon を許可します。

##### (j) swapon

swapon を許可します。

##### (k) mount

デバイスのマウントを許可します。

##### (l) raw\_io

*CAP\_SYS\_RAWIO* ケイパビリティを許可します。ioperm、iopl シス

テムコールの使用と /dev/mem へのアクセスを許可することを意味します。

- (m) ptrace  
ptrace の使用を許可します。
- (n) chroot  
chroot の使用を許可します。
- (o) unlabel  
(unlabeled\_t もしくは file\_t のような) ラベル付けされていないファイルへのアクセスを許可します。これはまたサポートされていないファイルシステムへのアクセスも許可します。何がサポートされていないかについては、base\_policy/genfs\_contexts を御覧下さい。
- (p) search  
s パーミッションをすべてのファイルに許可します。
- (q) read  
r パーミッションをすべてのファイルに許可します。
- (r) write  
w パーミッションをすべてのファイルに許可します。
- (s) all  
何もかもすべてを許可します!!

## 8.15 global とドメインにおける矛盾

ファイルに対する allow を除くと allow は一度宣言されると撤回することはできません。global ドメインにおいての使用は慎重に行ってください。global ドメインにおいて allow を使用すると、すべてのドメインにアクセス権限を与えてしまい、各々のドメインで deny できなくなります。

## 9 連絡先

コメント、提案、フィードバック等ありましたら、次の連絡先までお願いします。  
e-mail:seedit-admin@lists.sourceforge.net

## 10 TODO

- 単純化ポリシ記述言語のセキュリティを再検討する。デバイスのアクセス制御、ファイルのパーミッションについて問題が指摘されている。
- シンタックスを再検討し統一する。
- SELinux のポリシーそのものを含めて使用できるように言語を拡張する。
- ファイル以外のアクセス制御用の deny シンタックスをサポートする。
- チュートリアルを用意する。

## References

- [1] Security-Enhanced Linux URL=<http://www.nsa.gov/selinux>
- [2] 中村雄一 鮫島吉喜 ”Security-Enhanced Linux のアクセス制御ポリシー設定の簡易化”, 暗号と情報セキュリティシンポジウム (SCIS 2003), 静岡, 日本, 2003, Vol. II, 831.836. URL=<http://seedit.sourceforge.net/papers/scis2003paper.pdf>
- [3] Yuichi Nakamura ”Simplifying Policy Management with SELinux Policy Editor”, presentation on SELinux Symposium 2005, URL=<http://www.selinux-symposium.org/2005/presentations/session4/4-2-nakamura.pdf>
- [4] 日立ソフトウェアエンジニアリング株式会社 URL=<http://www.selinux.hitachi-sk.co.jp/>
- [5] SELinux Policy Editor Project URL=<http://seedit.sourceforge.net/>
- [6] Katsuya SUEYASU, Toshihiro TABATA, Kouichi SAKURAI, ”On the Security of SELinux with a Simplified Policy,” Proc. of the IASTED International Conference on Communication, Network, and Information Security (CNIS 2003), pp.79-84, Dec. 2003. URL=<http://www.swlab.it.okayama-u.ac.jp/~tabata/research/CNIS2003.sueyasu.pdf>