単純化ポリシーによるSELinuxの設定 バージョン1.2

中村 雄一*
和訳 山口 拓人†

September 1, 2005

Contents

*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
†岡山大学

1　この文書について

この文書は、単純化ポリシ (バージョン 1.2.0) 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、ポリシが複雑になってきていることから、ポリシ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」は中間設定言語に書かれているポリシです。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。気軽に試してみて下さい。

3　単純化ポリシの構成

単純化ポリシは単純化ポリシコンパイラとサンプルポリシの 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. 単純化ポリシコンパイラ
   単純化ポリシのコンパイラです。単純化ポリシコンパイラは単純化ポリシを読み込み、m4 マクロと checkpolicy により SELinux ポリシ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
   サンプルの単純化ポリシです。

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 unlabel を利用して下さい。

5 単純化ポリシのサンプル

この章では単純化ポリシ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, staff_r, user_r. の 3 つのロールが用意されています。

  - sysadm_r
    全てが許可されたロールです。デフォルトでは root しか使えません。
  - staff_r
    限定された権限を持ちますが，su コマンドを利用して sysadm_r になれます。また/root へ読み込みアクセス可能です。root しか使えません。
  - user_r
    限定された権限を持ちます。su コマンドを使えないので気をつけます。

5.2 単純化ポリシ用ディレクトリのコンテンツ

単純化ポリシのサンプルは、「/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 を御覧下さい。

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パ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ィアだと思います。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 単純化ポリシコンパイラのオプション

単純化ポリシコンパイラ (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 を御覧下さい)。
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 を御覧下さい。

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ないことに注意して下さい。

```
# In simplifed_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らなりません。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。

```
# simplifed_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。[1]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

[1]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。

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。

- 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。)
allowxxx (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/http 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*;

   (c) allow *directoryname* exclusive -all [r],[w],[x],[s];

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 のファイルタイプ遷移 (*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* というラベルが付与されます。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 のタイプと同じです。
   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。

   (c) これは， *directoryname* 以下の，ファイルタイプ遷移で作成されたファイル全てのアクセス権限を設定します。

3. パーミッションの意味 [2]

   - r: 読み込みと属性の閲覧

[2] [6] において問題が指摘されており、パーミッション種別については再検討する予定です。

- 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。

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(Extend attributes:拡張属性) 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,/dev/tmpfs については allowfs を参照して下さい。/selinux については allowseop 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 (-tcp|-udp) -port *port number*;
   (e) 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。ウェ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) 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。

(e) 予約されていないウェ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],[n],[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、[n] は signull, [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 [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。

8.13 Configuring access control to files on misc file systems

SELinux は，拡張属性をサポートしたファイルシステム (ext2,ext3,xfs など) については，一つ一つのファイル単位に細かいアクセス制御が可能です。これらのファイルシステム上のファイルについては allow 文を使って設定を行えばよいです。しかし、これら以外のファイルシステムにあるファイルについては，以下の *allowfs* 文を使って設定する必要があります。ファイル単位の設定はできず、「このファイルシステムにあるファイル全てに読み書き可能」のような大雑把な設定になります。

8.14 allowfs 文

- 書式

  1. allowfs *name_of_filesystem* [s],[r],[x],[w];
     For *name_of_filesystem tmpfs sysfs autofs usbfs cdfs romfs ramfs dosfs smbfs nfs proc proc_kmsg proc_kcore xattrfs* can be used.
  2. allowfs *name_of_filesystem* exclusive *label*;
  3. allowfs *name_of_filesystem label* [s],[r],[x],[w];
  4. allowfs *name_of_filesystem* -all [s],[r],[x],[w];

- 意味

  1. 指定されたファイルシステム上のファイルに対するアクセス制御を設定します。例えば, *allowfs proc s,r;* という記述は，proc ファイルシステム上にあるファイルに対して「s,r」というパーミッションを許可します。アクセス拒否ログにファイルシステム名_t というログを発見したなら，allowfs 文を使う必要があるでしょう。

2. 以下は，より進んだ設定オプションです。これは，SELinux のファイルタイプ遷移 (*file_type_auto_trans*). 設定です。 *name_of_filesystem* に作成されたファイルには， *label* というタイプが付与されます。なお *name_of_filesystem* には tmpfs のみ指定可能です。現バージョンの SELinux では，ファイルタイプ遷移が tmpfs でしかサポートされていないからです。
3. ファイルタイプ遷移によってラベルが付与されたファイル全てのアクセス制御を行います。 *name_of_filesystem* には tmpfs のみ指定可能です。

- 注意
  allowfs *name_of_filesystem* exclusive *label*; では，*label* の名称は *domain prefix_name_of_filesystem_t*. である必要があります。例えば，httpd_t ドメインでの設定の場合， *allowfs tmpfs exclusive httpd_tmpfs_t*. とします。

- *name_of_filesystem* に関する注意
  - proc ファイルシステム
    proc ファイルシステム (/proc 以下にマウントされたファイル) に対するアクセス制御はより細かく行われています。proc_kmsg は, /proc/kmsg, proc_kcore は /proc/kcore に対するアクセス制御を意味します。そして，proc はその他の proc ファイルシステム上のファイルを意味します。
  - xattrfs
    これは拡張属性をサポートしているが，SELinux のラベルを使うように設定されていないファイルシステムを意味します。例えば，USB メモリを非 SELinux マシンで ext3 ファイルシステムでフォーマットしたとします。次に，USB メモリを SELinux マシンにマウントします。すると，xattrfs 上のファイルとして認識されます。 *allowfs xattrfs ¡パーミッション¿;* を使う必要があります。
  - cdfs
    Iso9660 and UDF ファイルシステムを意味します。
  - dosfs
    Windows 上のファイルシステム (fat, vfat, ntfs) を意味します。
  - smbfs
    cifs,smbfs を意味します。

8.15 その他システム管理に関するアクセス制御

8.16 allowkernel

カーネルとの通信，カーネルの管理に関するアクセス制御を設定します。

- 書式
  allowkernel netlink|klog_read|klog_write|klog_adm|insmod;

- 意味

1. netlink
   Netlink ソケットを使ってカーネルと通信することを許可します。
2. klog_read
   syslog(2) システムコールを使ってカーネルメッセージを読み込むことを許可します。通常は dmesg コマンドを使いたい場合に許可します。
3. klog_write
   カーネル上の Audit subsystem(カーネルでログを取るしくみ) にログメッセージを送信する場合に許可します。ケーパビリティaudit_writeと同じです。
4. klog_adm
   カーネルのログ取り設定変更を許可します。ケーパビリティaudit_control,sys_pacctと同じです。
5. insmod
   カーネルモジュールのインストールを許可します。

8.17 allowseop

- 書式
  allowseop load_policy|setenforce|relabel|part_relabel|getsecurity;
- Meaning
  SELinux を管理する特権を与えます。

  1. relabel
     すべてのファイルを relabel することを許可する。この時、allowseop getsecurity と allowpriv search もする必要があります。
  2. 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。
  3. 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。
  4. setenforce
     enforcing モードもしくは permissive モードの切り替えを許可します。
  5. load_policy
     カーネルに対してポリシのロードを許可します。

8.18 allowpriv

- 書式
  allowpriv net|boot|quotaon| swapon|mount |rawio|ptrace*mid*chroot|unlabel |memlock|nice|resource| time|devcreate|setattr|search|read |write|all
- 意味
  他の特権を与えます。

1. 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。
2. 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。
3. 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。
4. quotaon
   quotaon を許可します。
5. swapon
   swapon を許可します。
6. mount
   デバイスのマウントを許可します。
7. rawio
   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。
8. ptrace
   ptrace の使用を許可します。
9. chroot
   chroot の使用を許可します。
10. unlabel
    ラベル付けが設定されていないもしくはラベル付けが壊れたファイル (labeled unlabeled_t タイプが付与されたファイル) にフルアクセス権限を与えます。
11. memlock
    ケーパビリティ*CAP_IPC_LOCK* と同じです。メモリをロックし，スワップされないようにする権限です。
12. nice
    ケーパビリティ *CAP_SYS_NICE* と同じです。プロセスの nice 値を変更する権限です。
13. resource
    ケーパビリティ *CAP_SYS_RESOURCE* と同じです。rlimit を利用してリソース上限を変更するなどの権限を与えます。
14. time
    ケーパビリティ *CAP_SYS_TIME* と同じです。システムクロックの変更を許可します。

15. devcreate
    書き込み可能なディレクトリにデバイスファイルを作成する権限を与えます。これを忘れるとたとえ書き込み可能に設定されたディレクトリであってもデバイスファイルを作成することができません。
16. setattr
    通常 setattr(所有者、更新時間などファイルの属性を変更) は，w パーミッションの中で許可されていますが，これを使うと s パーミッションが許可されたファイルに対して「setattr」を許可します。 書き込みはしないが、setattr アクセスだけを要求する場合に使います。通常は login_t,gdm_t ドメインにのみ使います。
17. search
    s パーミッションをすべてのファイルに許可します。
18. read
    r パーミッションをすべてのファイルに許可します。
19. write
    w パーミッションをすべてのファイルに許可します。
20. all
    何もかもすべてを許可します!!

8.18.1 出力される SELinux の設定

allowkernel,allowseop,allowpriv によって許可される SELinux のポリシを知るには，macros/seedit_macros.te にある allow_admin_xxxx マクロを見るとよいでしょう。例えば， *allowkernel klog_adm* を記述することで許可される設定を知るには，allow_admin_klog_adm を見ます。

8.19 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下さい。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。特にファイルとネットワークのアクセス制御。
- 詳細設定できるよう拡張

- 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。
- さらにドキュメントを用意する

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