

2007 情財第 977 号

2007年度  
オープンソースソフトウェア活用基盤整備事業

セキュリティ強化 Linux (SELinux) の  
管理運用手法の調査

第 IV 部  
プロトタイプ解説資料

2008年2月

独立行政法人 情報処理推進機構

## 目次

1. はじめに .....	4
1.1. 本文書の目的 .....	4
1.2. 本文書の構成 .....	4
2. OpenDRIM の概要 .....	4
2.1. OpenDRIM とは .....	4
2.2. OpenDRIM の構成 .....	5
2.3. 管理対象の追加方法 .....	7
3. プロトタイプの概要 .....	7
3.1. プロトタイプの全体構成 .....	7
3.2. プロトタイプの構成 .....	8
4. プロトタイプの動作環境構築 .....	8
4.1. OS のインストール .....	9
4.2. 開発ツールのインストール .....	9
4.3. Provider Common Library のセットアップ .....	10
5. プロトタイプ .....	11
5.1. プロトタイプのインストール .....	11
5.1.1. 管理対象マシン上での作業 .....	11
5.1.2. 管理クライアント上での作業 .....	12
5.2. プロトタイプの動作確認方法 .....	12
5.2.1. SELinux の ON/OFF 監視機能テスト .....	12

5.2.2.	SELinux のログ監視機能テスト.....	13
5.3.	プロトタイプソース解説.....	14
5.3.1.	SELinux の ON/OFF 監視機能 .....	14
5.3.2.	SELinux のログ監視機能 .....	15
Appendix I OpenDRIM Provider 開発環境		
6.	OpenDRIM Provider の開発環境構築 .....	17
6.1.	前提となる開発ツールのインストール .....	17
6.2.	Eclipse のインストール .....	18
6.3.	OpenDRIM Provider の開発用ツール等のインストール.....	20

## 1. はじめに

### 1.1. 本文書の目的

本文書は、第Ⅲ部「運用管理製品の SELinux 対応に必要な機能概要」の中で説明した「SELinux の ON/OFF 監視機能」及び「SELinux のログ監視機能」を実装したプロトタイプの解説文書である。

運用管理製品を SELinux 対応させるため、各機能の詳細設計及び実装を行う上で、プロトタイプを理解するために利用にしていきたい。

### 1.2. 本文書の構成

本調査では、後で説明する OpenDRIM の Provider として実装している。そこで、まず、第 2 章でプロトタイプに利用した OpenDRIM の概要を説明する。次に、第 3 章で、プロトタイプを動作させるために必要な環境構築方法を説明し、第 4 章でプロトタイプのインストール及び実行方法を説明する。

また、Appendix として、今回活用した OpenDRIM での Provider 開発環境の構築方法を簡単に説明する。

## 2. OpenDRIM の概要

### 2.1. OpenDRIM とは

OpenDRIM (Open Distributed Resource Information Management) は、DMTF<sup>1</sup> (Distributed Management Task Force) が定義し公開している標準仕様である CIM<sup>2</sup> (Common Information Model) / WBEM<sup>3</sup> ([Web-Based Enterprise Management](http://www.dmtf.org/standards/wbem/)) に準拠した、オープンな Linux ベースでのリソース情報マネージメント環境である。OpenDRIM を用いることにより、ネットワーク上に分散しているリソース (OS を含むソフトウェアリソース、ハードウェアリソースなど) を集中管理することができる。

OpenDRIM の開発は、中国・日本・韓国の三カ国で構成される北東アジア OSS 推進フォーラム<sup>4</sup>での活動を中心として、OpenDRIM プロジェクト<sup>5</sup>で進められている。OpenDRIM プロジェクトの成果物は、ライ

---

<sup>1</sup> <http://www.dmtf.org/standards/>

<sup>2</sup> <http://www.dmtf.org/standards/cim/>

<sup>3</sup> <http://www.dmtf.org/standards/wbem/>

<sup>4</sup> [http://www.ipa.go.jp/software/open/forum/north\\_asia/index.html](http://www.ipa.go.jp/software/open/forum/north_asia/index.html)

<sup>5</sup> <http://www.opendrim.org/opendrim>

ブラリ類が LGPL、Provider(後述)が LGPL で公開されている。また、OpenDRIM プロジェクトは、Red Flag Linux を管理するための OpenLIME プロジェクト、SMASH を管理するための OpenSMASH プロジェクト、仮想化を管理するための OpenVINE プロジェクトの親プロジェクトとなっている。

2008 年 2 月現在、OpenDRIM は以下のディストリビューションをサポートしている。

- Red Flag
- Asianux
- Booyo Linux
- Fedora
- CentOS

本文書では、CentOS5.1 を対象にして説明する。

## 2.2. OpenDRIM の構成

OpenDRIM は図 1 のように WBEM の標準に則った構成となっている。

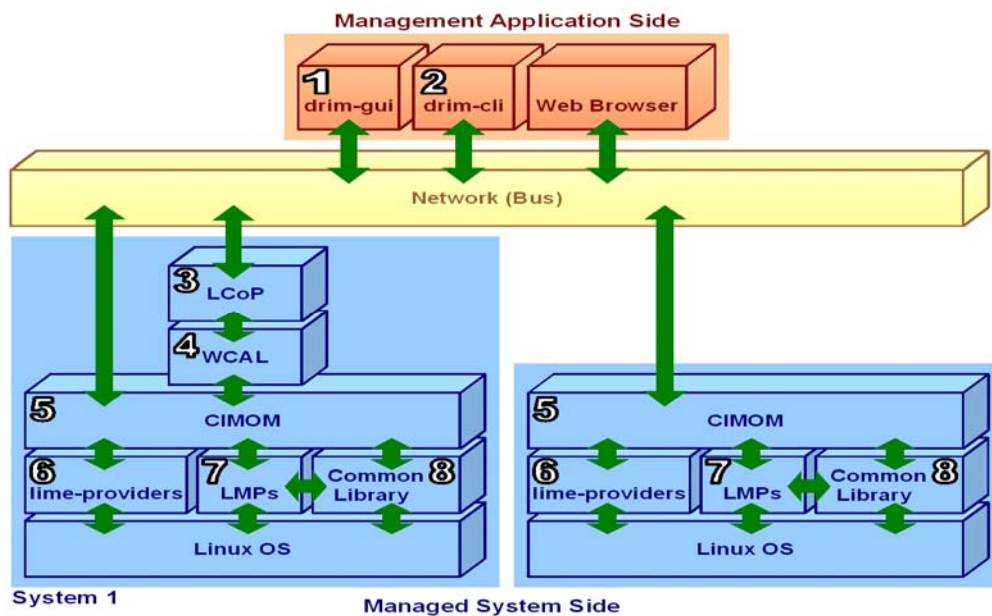


図 1

おおまかに分類すると、管理される対象となるシステム側(図 1 の Managed System Side)の構成と、管理を行うアプリケーション側(図 1 の Management Application Side)の 2 つに分類できる。

## Management Application Side

集中管理を行うためのアプリケーションとして、GUI ベース (drim-gui) / CUI ベース (drim-cui) / ブラウザベース (Web Browser) のものが提供されている。アプリケーションは API からネットワークを通し、XML を用いて CIMOM (Common Information Model Object Manager) と情報をやり取りする。GUI/CUI ベースの際には直接 CIMOM と通信を行い、ブラウザベースの際には LcoP/WCAL を経由して通信を行う。

## Managed System Side

管理対象となるシステム内では、CIMOM がネットワークを経由してきたアプリケーションからの通信に対して必要な認証チェックを行う。CIMOM は、CIM 標準で書かれている管理対象モデルと通信を行う。この通信には "Provider" と呼ばれる小さなインタフェース用コードを用いて行われる。OpenDRIM では、RedFlag Linux を管理するための lime-provider、及びその他の Linux を管理するための Linux Management Provider (LMP) が用意されている。また、Provider が CIMOM とやり取りするための Common Library も提供されている。

OpenDRIM プロジェクトでは、Linux を管理するための LMP (Linux Management Provider) として、下記の Provider を提供している<sup>6,7</sup>。

- Base\_Server/
- CPU/
- EXT3FS/
- Interop/
- LVM/
- Linux\_Partition/
- Logrotate/
- OS/
- PACKAGENAME/
- Record\_Log/
- Service/
- Simple\_Identity\_Management/

---

<sup>6</sup> 2008/01/18 に調査時点での情報である。

<sup>7</sup> <http://www.opendrim.org/opendrim/wiki/LinuxManagementProviders> の Linux Management Providers 階層図を参照のこと

## 2.3. 管理対象の追加方法

デバイス製造業者やシステム及びアプリケーション提供業者が、新たに OpenDRIM の管理対象とするためには、以下の手順で行う。

1. モデルを実装する。
2. provider を実装する。

ここで、DMTF では様々な管理対象リソースに対し、各々の管理情報の典型的なモデルを標準モデルとして公表している。例えば、システム情報、ネットワーク情報、アプリケーションやサービスの情報、デバイス情報などである。その中ではインタフェースが明確に定義され、呼び出しメソッドの種類も限定されている。そのため、1. では、基本的にこの標準モデルを拡張する方法を取る。

OpenDRIM プロジェクトでは、モデルやプロバイダ、GUI などを実装するための様々なツールやライブラリを提供している。以降で利用する Provider Common Library もその一つである。これらを用いれば、比較的簡単に provider を実装することができる。

## 3. プロトタイプの概要

### 3.1. プロトタイプの全体構成

プロトタイプの全体構成を図 2 に示す。OpenDRIM では、図 2 のように OpenPegasus を CIM Server として利用し、新たな管理対象に対する provider を実装する。本調査で作成したプロトタイプは、図 2 の OpenPegasus CIM Server 以外の部分である。

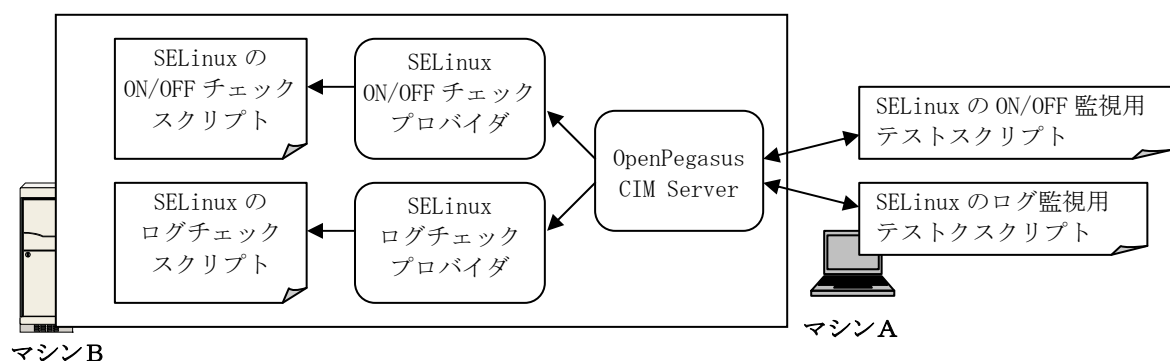


図 2

## 3.2. プロトタイプの構成

本調査の成果物である `proto_type.20080220.tar.gz` ファイルは、プロトタイプ自体のほか、yum で簡単に入手できないパッケージを、tar.gz 形式でアーカイブ化したものである。このファイルを展開すると図 3 のような構造になる。以降では、`proto_type.20080220.tar.gz` ファイルを `/root/` 以下に展開したとして説明する。

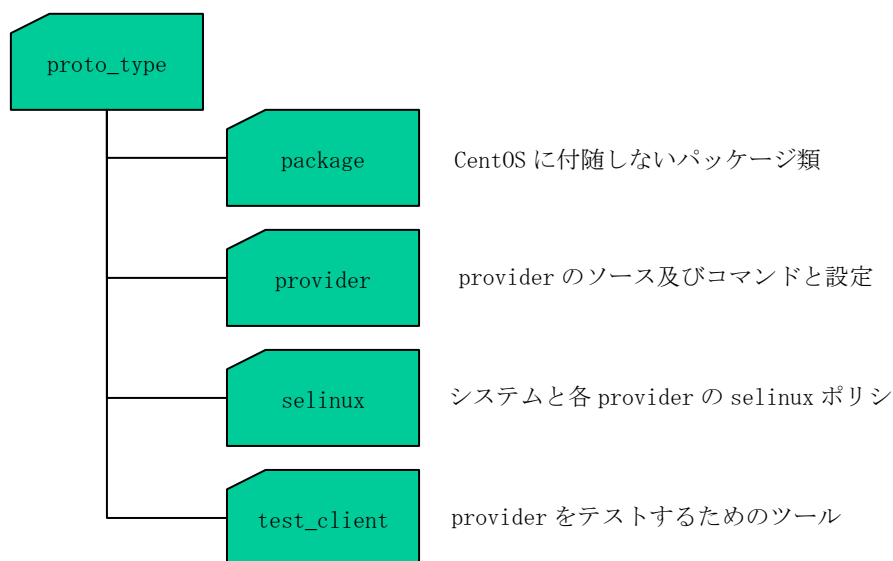


図 3

## 4. プロトタイプの動作環境構築

OpenDRIM の Provider を動作させるためには、基本的に以下の 2 つをインストールする必要がある。

- Provider Common Library
- OpenPegasus ヘッダ

この中で” Provider Common Library” は、Provider を動作させるサーバ上でコンパイルする必要がある。しかし、コンパイルを実行するには、autoconf など様々な開発ツールやヘッダ類が必要となる。本章では、OpenDRIM の Provider を動作させるために必要な動作環境を構築する手順を説明する。

## 4.1. OS のインストール

### 1. OS のインストールとアップデート

CentOS5.1 のインストールを行う。インストーラで「Server」を選択し、それ以外はデフォルトの状態で行う。

インストールが完了してシステムが再起動したら、以下でシステムを最新の状態にアップデートする。

```
#yum -y update
```

## 4.2. 開発ツールのインストール

### 1. CentOS 5.1 の Development tool のインストール

端末から以下を実行する。

```
# yum -y groupinstall "Development Tools"
# yum -y install pam-devel openssl-devel net-snmp-devel libxml2-devel
```

### 2. OpenPegasus のインストール

/root/proto\_type/package 以下に同梱されている OpenPegasus の Source RPM (tog-pegasus-2.7.0-2.src.rpm) を利用し<sup>8</sup>、端末から以下を実行する。

```
# cd /root/proto_type/package
# rpm -ivh tog-pegasus-2.7.0-2.src.rpm
# rpmbuild -bb /usr/src/redhat/SPECS/tog-pegasus.spec
# cd /usr/src/redhat/RPMS/i386
# rpm -ivh tog-pegasus-2.7.0-2.i386.rpm tog-pegasus-devel-2.7.0-2.i386.rpm
```

ここで、rpmbuildで作成される tog-pegasus-devel-2.7.0-2.i386.rpm はインストールしない。

### 3. CIM Schema の入れ替え

/root/proto\_type/package 以下に同梱されている OpenDRIM の CIM スキーマファイル (cimv216Experimental-20070928.zip) を利用し<sup>9</sup>、端末から以下を実行する。

<sup>8</sup> OpenPegasus のサイトからダウンロードしても良い。

(<http://www.openpegasus.org/pr/uploads/40/14962/tog-pegasus-2.7.0-2.src.rpm>)

```
# cd /root/proto_type/package
# rm -rf /var/lib/Pegasus/repository/root#cimv2
# mkdir cimv216e
# cp cimv216Experimental-20070928.zip cimv216e
# cd cimv216e
# unzip cimv216Experimental-20070928.zip
# cimmofl -R /var/lib/Pegasus/ -aE -n root/cimv2 cimv216.mof
```

#### 4. OpenPegasus の起動設定の追加及び起動

端末から以下を実行する。

```
# chkconfig tog-pegasus on
# service tog-pegasus start
# cimconfig -s enableHttpConnection=true -p
```

#### 5. SBLIM WBEM のインストール

端末から以下を実行する。

```
# yum -y install sblim-wbemcli
```

### 4.3. Provider Common Library のセットアップ

#### 1. Provider Common Library の入手とコンパイル/インストール

proto\_type/package 以下に同梱されている Provider Common Library ファイル (Provider\_Common\_Library.tar.gz) を利用し、端末から以下を実行する。

```
# cd /root/proto_type/package
# tar -zxvf Provider_Common_Library.tar.gz
# cd Provider_Common_Library
# autoreconf -vfi
# ./configure --prefix=/usr CIMSERVER=pegasus
# make
# make install
```

---

<sup>9</sup> OpenDRIM プロジェクトから Subversion で入手しても良い。

([http://www.opendrim.org/svn/opendrim/Provider\\_Common\\_Library/](http://www.opendrim.org/svn/opendrim/Provider_Common_Library/))

## 2. SBLIM CIM Client のインストール

端末から以下を実行する。

```
# yum -y install sblim-cim-client
```

## 5. プロトタイプ

本章では、本調査で作成したプロトタイプの Provider を動作させるまでの手順を説明する。

### 5.1. プロトタイプのインストール

#### 5.1.1. 管理対象マシン上での作業

管理対象マシン(図 2 のマシン B)にプロトタイプの Provider をインストールする。下記の手順で行う。

##### Provider 動作のための構築

- 4 章で説明した Provider の動作環境構築を行う。
- 外部から Provider に接続するためのユーザを作成する。作成するユーザのユーザ名は"opendrim"、パスワードは"opendrim"とする。
- /etc/Pegasus/access.conf を修正し、下記のように"opendrim"ユーザが外部から接続できるようにする<sup>10</sup>。  
-: ALL EXCEPT opendrim pegasus:wbemNetwork
- 外部から Provider へ接続可能にするために、iptables の設定を修正する。  
system-config-securitylevel ツールを起動し、Customize を選択して、5988/tcp のポートへの外部からの接続を許可する。

##### プロトタイプ Provider のインストール

- /root/proto\_type/provider/以下に同梱されている semode.sh, selog.sh を/usr/local/sbin 以下にコピーする。
- /root/proto\_type/provider/selog/OpenDRIM\_Selinux に移動する。
- 以下の手順でコンパイル及びインストールする。

```
# ./configure CIMSERVER=pegasus --prefix=/usr
# make
# make install
# make postinstall
```

<sup>10</sup> proto\_type/provider/access.conf を参照の事

```
# make clean
```

4. proto\_type/provider/semode/SelinuxService に移動する。
5. 3. の手順と同様にコンパイル及びインストールする。

### SELinux ポリシのインストール

1. /root/proto\_type/selinux に移動し、"install\_policy.sh"を実行し次を実行する。

```
# restorecon -RF /usr/local/sbin
```

2. 以下で tog-pegasus を再起動する。

```
# killall cimserver cimservermain
```

```
# /etc/init.d/tog-pegasus restart
```

3. ps -eZ | grep cimserver でプロセスを確認し、pegasus\_t ドメインで動作していることを確認する。

### 5.1.2. 管理クライアント上での作業

管理対象クライアント(図 2 のマシン A)にテストスクリプトをインストールする。以下の手順で行う。

1. 以下のコマンドで、wbemcli コマンドと tog-pegasus パッケージをインストールする。

```
# yum install sbim-wbemcli
```

2. /root/proto\_type/test\_client ディレクトリをマシン B 上にコピーする。

## 5.2. プロトタイプの動作確認方法

5.1.2 項でインストールしたテストスクリプトを用いて、管理対象マシンに接続し、情報が取得できることを確認する。

### 5.2.1. SELinux の ON/OFF 監視機能テスト

#### テストエージェントの実行

テストスクリプト(check\_semode.sh)を使用する。

```
# check_semode.sh "管理対象対象マシン(図 2のマシンB)のIPアドレス"
```

#### テストの出力結果

SELinux の ON/OFF が次のように出力される。

```
-EnabledState=1
```

```
-OtherEnabledState="Enforcing" ← SELinuxの現在の状態(Enforcing/Permissive/Disabled)
```

## 5.2.2. SELinux のログ監視機能テスト

### テストエージェントの実行

テストスクリプト (check\_selog.sh) を使用する。

```
# check_selog.sh "管理対象対象マシン(図 2のマシンB)のIPアドレス"
```

### テストの出力結果

SELinux のログが次のように出力される。

```
localhost:5988/root/cimv2:OpenDRIM_SeLogEntry.InstanceID="0" Caption,Description,ElementName,RecordFormat,RecordData="",Locale,InstanceID="0",LogInstanceID,LogName,RecordID,CreationTimeStamp
localhost:5988/root/cimv2:OpenDRIM_SeLogEntry.InstanceID="1" Caption,Description,ElementName,RecordFormat,RecordData="----&#10;time->Mon Feb 18 00:26:46 2008&#10;type=SYSCALL msg=audit(1203312406.312:46): arch=40000003 syscall=11 success=yes exit=0 a0=9d0ad00 a1=9d0a5e0 a2=9d0a6f8 a3=9d0a278 items=0 ppid=2989 pid=2990 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="ifconfig" exe="/sbin/ifconfig" subj=root:system_r:ifconfig_t:s0 key=(null)&#10;type=AVC msg=audit(1203312406.312:46): avc: &#32;denied &#32;{ noatsecure } for &#32;pid=2990 comm="ifconfig" scontext=root:system_r:pegasus_t:s0 tcontext=root:system_r:ifconfig_t:s0 tclass=process&#10;type=AVC msg=audit(1203312406.312:46): avc: &#32;denied &#32;{ rlimitinh } for &#32;pid=2990 comm="ifconfig" scontext=root:system_r:pegasus_t:s0 tcontext=root:system_r:ifconfig_t:s0 tclass=process&#10;type=AVC msg=audit(1203312406.312:46): avc: &#32;denied &#32;{ siginh } for &#32;pid=2990 comm="ifconfig" scontext=root:system_r:pegasus_t:s0 tcontext=root:system_r:ifconfig_t:s0 tclass=process&#10;","Locale,InstanceID="1",LogInstanceID,LogName,RecordID,CreationTimeStamp
localhost:5988/root/cimv2:OpenDRIM_SeLogEntry.InstanceID="2" Caption,Description,ElementName,RecordFormat,RecordData="----&#10;time->Mon Feb 18 00:26:46 2008&#10;type=SYSCALL msg=audit(1203312406.323:47): arch=40000003 syscall=5 success=yes exit=7 a0=61eab0 a1=8000 a2=1b6 a3=8928008 items=0 ppid=2989 pid=2990 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="ifconfig" exe="/sbin/ifconfig" subj=root:system_r:ifconfig_t:s0 key=(null)&#10;
```

## 5.3. プロトタイプのソース解説

### 5.3.1. SELinux の ON/OFF 監視機能

#### ソース概要

SELinux の ON/OFF 監視機能を実現する Provider の概要は図 4 のようになっている。この Provider は、SELinux の状態を取得するために `getenforce` コマンドを使用し、その出力結果をクライアントに渡している。

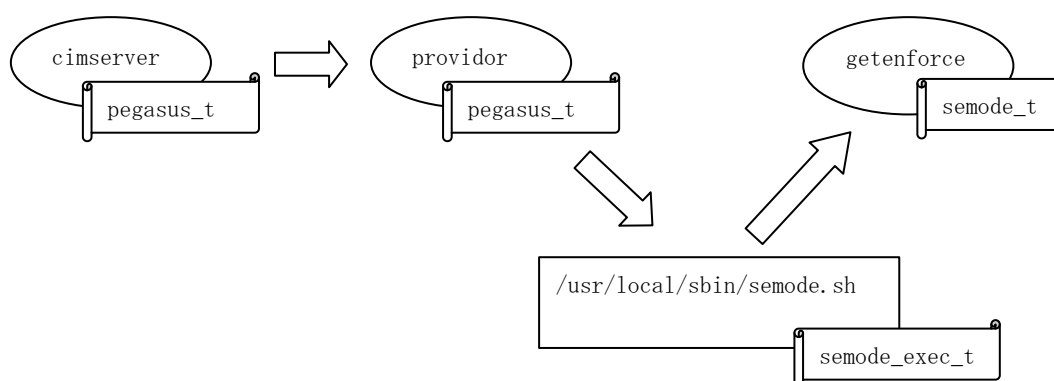


図 4

#### コア部分の解説

6 章で説明する Provider Code Generator を用いて、Provider の雛型を作成した。SELinux の ON/OFF 監視機能のコアとなるソースファイルは以下である。

```

/root/proto_type/provider/semode/SelinuxService/OpenDRIM_SelinuxService/
OpenDRIM_SelinuxServiceAccess.cpp
  
```

ここで、この Provider は、SELinux の設定でドメインを分割ができるように、`semode.sh` スクリプト経由で、`getenforce` コマンドを実行するように実装した。

`/usr/local/sbin/semode.sh` スクリプトは、`getenforce` コマンドを実行し、その結果を返すシェルスクリプトとなっている。SELinux の ON/OFF 監視機能のコアとなるソースファイルのコアとなるソースファイルで、この `/usr/local/sbin/semode.sh` スクリプトを実行し、その結果をクライアントに返している。

## SELinux の設定の解説

図 4 で示したようなドメイン遷移が行われるように SELinux の設定を行った。

まず、`/usr/local/sbin/semode.sh` に `semode_exec_t` タイプを付与した。また Provider プロセスが `semode_exec_t` タイプのファイルを実行した際に、起動されるプロセスが `semode_t` ドメインになるように設定した。この結果、`getenforce` コマンドは `semode_t` ドメインで実行されるようになる。さらに、`semode_t` ドメインが `selinux` の状態を取得できるようにアクセス権を付与した。

この設定により、この Provider のみが、SELinux の ON/OFF 情報を取得できるようになっている。また、仮にこの Provider に脆弱性があっても、他に影響を与えないようになっている。

### 5.3.2. SELinux のログ監視機能

#### ソース概要

SELinux のログ監視機能を実現する Provider の概要は図 5 のようになっている。この Provider は、SELinux のログを取得するために” `ausearch -m AVC`” コマンドを使用し、その出力結果をクライアントに渡している。

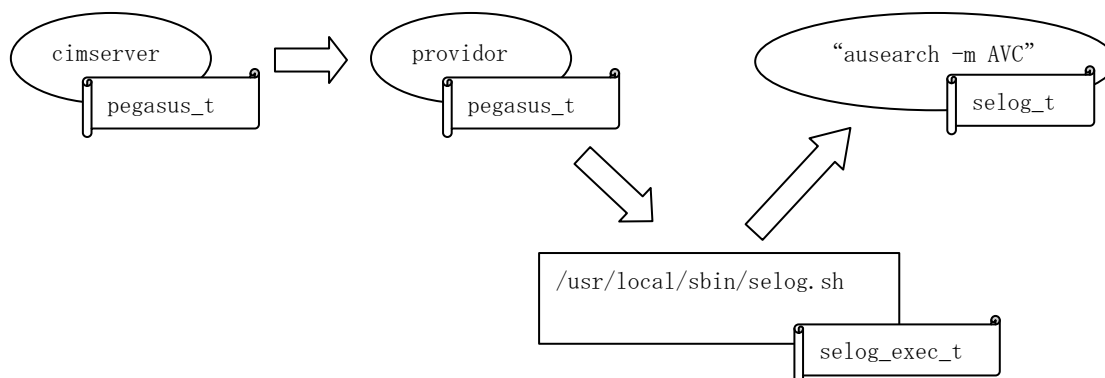


図 5

#### コア部分の解説

6 章で説明する Provider Code Generator を用いて、Provider の雛型を作成した。SELinux のログ監視機能のコアとなるソースファイルは以下である。

```
/root/proto_type/provider/selog/OpenDRIM_Selinux/OpenDRIM_SeLogEntry/  
OpenDRIM_SeLogEntryAccess.cpp
```

ここで、この Provider は、SELinux の設定でドメインを分割ができるように、`selog.sh` スクリプト経由で、`getenforce` コマンドを実行するように実装した。

/usr/local/sbin/selog.sh スクリプトは、” ausearch -m AVC” コマンドを実行し、その結果を返すシェルスクリプトとなっている。SELinux のログ監視機能のコアとなるソースファイルで、この /usr/local/sbin/selog.sh スクリプトを実行し、その結果をクライアントに返している。

### SELinux の設定の解説

図 5 で示したようなドメイン遷移が行われるように SELinux の設定を行った。

まず、/usr/local/sbin/selog.sh に selog\_exec\_t タイプを付与した。また Provider プロセスが selog\_exec\_t タイプのファイルを実行した際に、起動されるプロセスが selog\_t ドメインになるように設定した。この結果、” ausearch -m AVC” コマンドは selog\_t ドメインで実行されるようになる。さらに、semode\_t ドメインが selinux のログを取得できるようにアクセス権を付与した。

この設定により、この Provider のみが、SELinux のログを取得できるようになっている。また、仮にこの Provider に脆弱性があっても、他に影響を与えないようになっている。

## Appendix I OpenDRIM Provider 開発環境

### 6. OpenDRIM Provider の開発環境構築

OpenDRIM の Provider の開発方法は、OpenDRIM プロジェクトでドキュメント化されている<sup>11</sup>。しかし、開発環境構築を行うための手順が整理されておらず、本調査のプロトタイプ作成でも苦労した。そこで本章では、CentOS5.1 における Provider の開発環境構築の手順を説明する。この手順には、本章のみで開発環境構築ができるようにするために、4 章の内容と一部重複する部分が含まれている。

実際の Provider 開発については、OpenDRIM プロジェクトのドキュメント<sup>12,13</sup>を参照して頂きたい。

#### 6.1. 前提となる開発ツールのインストール

OpenDRIM の Provider 開発の前提となる環境構築の構築手順は、4.2 節と同様である。以下に再度手順を示す。

##### 1. CentOS 5.1 の Development tool のインストール

端末から以下を実行する。

```
# yum -y groupinstall "Development Tools"
# yum -y install pam-devel openssl-devel net-snmp-devel libxml2-devel
```

##### 2. OpenPegasus のインストール

proto\_type/package 以下に同梱した OpenPegasus の Source RPM (tog-pegasus-2.7.0-2.src.rpm) を利用し<sup>14</sup>、端末から以下を実行する。

```
# cd /root/proto_type/package
# rpm -ivh tog-pegasus-2.7.0-2.src.rpm
# rpmbuild -bb /usr/src/redhat/SPECS/tog-pegasus.spec
# cd /usr/src/redhat/RPMS/i386
# rpm -ivh tog-pegasus-2.7.0-2.i386.rpm tog-pegasus-devel-2.7.0-2.i386.rpm
```

<sup>11</sup> <http://www.opendrim.org/opendrim/wiki/HOWTOs>

<sup>12</sup> <http://www.opendrim.org/opendrim/wiki/ProviderCodeGenerator>

<sup>13</sup> <http://www.opendrim.org/opendrim/wiki/CodeGeneratorBundle>

<sup>14</sup> OpenPegasus のサイトからダウンロードしても良い。

ここで、rpmbulid で作成される tog-pegasus-devel-2.7.0-2.i386.rpm はインストールしない。

### 3. CIM Schema の入れ替え

proto\_type/package 以下に同梱した OpenDRIM の CIM スキーマファイル (cimv216Experimental-20070928.zip) を利用し<sup>15</sup>、端末から以下を実行する。

```
# cd /root/proto_type/package
# rm -rf /var/lib/Pegasus/repository/root#cimv2
# mkdir cimv216e
# cp cimv216Experimental-20070928.zip cimv216e
# cd cimv216e
# unzip cimv216Experimental-20070928.zip
# cimmofl -R /var/lib/Pegasus/ -aE -n root/cimv2 cimv216.mof
```

### 4. OpenPegasus の起動設定の追加及び起動

端末から以下を実行する。

```
# chkconfig tog-pegasus on
# service tog-pegasus start
# cimconfig -s enableHttpConnection=true -p
```

### 5. SBLIM WBEM のインストール

端末から以下を実行する。

```
# yum -y install sblim-wbemcli
```

## 6.2. Eclipse のインストール

OpenDRIM の Provider の開発では Eclipse を用いる。また、OpenDRIM の Provider の開発には、様々な Eclipse のプラグインをインストールする必要がある。この手順を以下に示す。

### 1. JDK のインストール

- 以下の URL から JDK 6 Update 4 をクリックし、Linux RPM in self-extracting file jdk-6u4-linux-i586-rpm.bin をダウンロードする。
  - <http://java.sun.com/javase/downloads/index.jsp>
- ダウンロードしたファイルを実行、ライセンスに同意してインストールする。

<sup>15</sup> OpenDRIM のサイトからダウンロードしても良い。

3. 環境変数に以下を追加する。

```
JAVA_HOME=/usr/java/jdk1.6.0_04
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
```

## 2. Eclipse の入手と展開

1. 以下のURLから Eclipse Classic 3.3.1.1 - Linux (137 MB)をクリックし eclipse-SDK-3.3.1.1-linux-gtk.tar.gz をダウンロードする。
  - <http://www.eclipse.org/downloads/>
2. ダウンロードしたファイルを/usr/lib/eclipse 以下に展開し以下を実行する。

```
# ln -s /usr/lib/eclipse/eclipse /usr/bin/eclipse
```

## 3. CDT プラグインのインストール

1. Eclipse を起動する。
2. Help→Software Updates→Find and Install をクリックする。
3. Search for new features to install を選択→Next をクリックする。
4. Europa Discovery Site にチェックを入れて Finish をクリックする。
5. ミラーを選択する。
6. Europa Discovery Site→C and C++ Development にチェック→Next をクリックする。
7. ライセンスの確認に同意して Next をクリックする。
8. プラグインのインストール先は変更せず Finish をクリックする。
9. 確認して Install All をクリックする。

## 4. EMF/XSD/SDO プラグインのインストール

1. Eclipse を起動する
2. Help→Software Updates→Find and Install をクリックする。
3. Search for new features to install を選択→Next をクリックする。
4. Europa Discovery Site にチェックを入れて Finish をクリックする。
5. ミラーを選択する。
6. Europa Discovery Site→Models and Model Development で以下をチェックする。
  - Eclipse Modeling Framework (EMF) Runtime、Eclipse Modeling Framework (EMF) Extender SDK
  - XML Schema Infoset Model (XSD)、
  - XML Schema Infoset Model (XSD) Extender SDK、
  - EMF Service Data Objects (SDO)、
  - EMF Service Data Objects (SDO) Extender SDK
7. ライセンスの確認に同意して Next をクリックする。
8. プラグインのインストール先は変更せず Finish をクリックする。

9. 確認して Install All をクリックする。

### 5. Subclipse プラグインのインストール

1. Eclipse を起動する
2. Help→Software Updates→Find and Install をクリックする。
3. Search for new features to install を選択→Next をクリックする。
4. New Remote Site をクリックし、以下の名前と URL を記入して OK をクリックする。
  - Name: Subclipse
  - URL: http://subclipse.tigris.org/update
5. Subclipse にチェックを入れて Next をクリックする。
6. ライセンスの確認に同意して Next をクリックする。
7. プラグインのインストール先は変更せず Finish をクリックする。
8. 確認して Install All をクリックする。

## 6.3. OpenDRIM Provider の開発用ツール等のインストール

OpenDRIM の Provider 開発には、Provider Common Library、SBLIM CIM Client と OpenDRIM Code Generator のインストールが必要である。前者二つは、4.3 節と同様である。以下に手順を示す。

### 1. Provider Common Library の入手とコンパイル/インストール

proto\_type/package 以下に同梱した Provider Common Library ファイル (Provider\_Common\_Library.tar.gz) を利用し、端末から以下を実行する。

```
# cd /root/proto_type/package
# tar -zxvf Provider_Common_Library.tar.gz
# cd Provider_Common_Library
# autoreconf -vfi
# ./configure --prefix=/usr CIMSERVER=pegasus
# make
# make install
```

### 2. SBLIM CIM Client のインストール

端末から以下を実行する。

```
# yum -y install sblim-cim-client
```

### 3. OpenDRIM Code Generator のインストール

1. Eclipse を起動する。

2. Help→Software Updates→Find and Install をクリックする。
3. Search for new features to install を選択→Next をクリックする。
4. New Remote Site をクリックし、名前と URL を記入して OK をクリックする。
  - Name: OpenDRIM
  - URL: ftp://www.opendrim.org/pub/PCG
5. OpenDRIM にチェックを入れて Next をクリックする。
6. ライセンスの確認に同意して Next をクリックする。
7. プラグインのインストール先は変更せず Finish をクリックする。

確認して Install All をクリックする。

ここでインストールした OpenDRIM Code Generator を利用することで、OpenDRIM の Provider の雛形を作ることができる。実際の Provider 開発は、この雛形を用いて行う。

OpenDRIM Code Generator の利用法や、実際の Provider 開発方法については、OpenDRIM プロジェクトのドキュメント<sup>16,17</sup>を参照して頂きたい。

---

<sup>16</sup> <http://www.opendrim.org/opendrim/wiki/ProviderCodeGenerator>

<sup>17</sup> <http://www.opendrim.org/opendrim/wiki/CodeGeneratorBundle>