SCIS 2003 The 2003 Symposium on
Cryptography and Information Security
Hamamatsu,Japan,Jan.26-29,2003
The Institute of Electronics
Information and Communication Engineers

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

Configuration system for access control policy of Security-Enhanced Linux

中村 雄一* 鮫島 吉喜*
Yuichi Nakamura Yoshiki Sameshima

あらまし Security-Enhanced Linux(SELinux)のアクセス制御ポリシの設定を簡易化するシステムについて報告する。SELinuxは、カーネルレベルでプロセスごとに細かいアクセス制御設定を行うことができる。しかし、SELinuxの設定状況の把握や設定の追加は困難であるため、設定コストがかかるだけでなく、誤設定によるセキュリティ上の問題が起きうる。本報告ではSELinuxの設定を簡略化するための中間言語を実装した。中間言語をGUIにより編集し、中間言語を中間言語コンパイラによりSELinuxの設定に変換する。上記方式を採用したSELinuxアクセス制御ポリシ設定システムを実装し、その結果SELinuxのアクセス制御ポリシ設定が容易になった。

キーワード セキュアOS、Trusted OS、アクセス制御ポリシ

1 はじめに

近年高品質・低コストという理由でLinuxの普及が進んでいるが、その普及につれ、Linuxはクラッカーやワームによる攻撃を受ける機会が増大している。これらの攻撃に対する根本的な対策として、Security-Enhanced Linux (SELinux)[1][2]が注目されている。SELinuxはカーネルレベルで粒度の細かい強制アクセス制御(MAC)を行っており、攻撃を受けたとしてもその被害は最小限になる[3]。SELinux を用いてセキュアなシステムを構築するためには、SELinux のアクセス制御ポリシの設定を正しく行う必要がある。しかし、その設定状況の把握が困難である上に、SELinuxのバージョンアップごとに設定方法が変化する。そのため、SELinuxのアクセス制御ポリシ設定を行うには相当のノウハウと時間を要し、それがSELinuxの普及の妨げになっていると考えられる。本報告では SELinux のアクセス制御ポリシを簡易化するシステムと、その有効性の検証について報告する。

* 日立ソフトウェアエンジニアリング株式会社
〒140-0002 東京都品川区東品川4丁目12番7号; Hitachi Software Engineering Co., Ltd.; 4-12-7 Higashi-Shinagawa Shinagawa-ku, Tokyo, 140-0002 Japan

2 SELinuxのアクセス制御ポリシ設定の問題点

2.1 SELinuxの機能

SELinuxの主な機能はアクセス制御機能である。アクセス制御機能は次の３つの要素から成り立っている。

(1) 強制アクセス制御(MAC)

SELinux のアクセス制御は全てのプロセスに例外なく行われ、その設定はセキュリティポリシ管理者のみが行うことができる。

(2) ラベルによるプロセスごとのアクセス制御

SELinuxでは、プロセスとオブジェクトにラベルを付与し、プロセスごとのアクセス制御ができる。オブジェクトとは、プロセスがアクセスするリソースのことを指す。プロセスに付与されるラベルのことをドメインと呼ぶ。あるドメインを持ったプロセスがあるラベルをもったオブジェクトに対して操作を行う際に、パーミッションをチェックし、アクセス制御を行う。

(3) ドメイン遷移によるプロセスへのラベル付与

プロセスにドメインを割り当てるにはドメイン遷移という機能を利用する。プロセスがプログラムを

exec システムコールで実行すると、デフォルトでは実行されたプロセスのドメインは変わらない。しかし、設定により、実行されたプロセスに付与されるドメインを変えることができる。このことをドメイン遷移と呼ぶ。

これらの機能により、SELinux ではプロセスに必要最小限のパーミッションが割り当てられ、そのパーミッションの範囲内でプロセスが動作することが保証される。

2.2 SELinux のアクセス制御ポリシ記述言語

アクセス制御ポリシ設定を SELinux 設定言語で記述することで、前節で説明した SELinux の機能が有効になる。SELinux 設定言語の記法について述べる。

(1) ラベルの宣言

ラベルを使うためには宣言をする必要があり、その宣言は、次のように行う。

type *T*;

T はドメインまたはリソースのラベルである。

(2) ドメイン遷移

ドメイン遷移の記述は次のように行う。

type_transition *F E*:**process** *D*;

F は遷移元のドメイン、*E* は実行ファイルのラベル、*D* 遷移先のドメインである。遷移元のドメインを持ったプロセスが指定されたラベルを持ったファイルを実行すると、実行されたプロセスは遷移先のドメインで動作する。

(3) パーミッションの付与

パーミッションの付与は、次のように行う。

allow *D L S*:{ P_1 P_2… };

D はドメイン、*L* はリソースのラベル、P_1、P_2…はパーミッションである。例えば「**allow** *httpd_t http_sys_content_t file*:{ *read write* };」という設定の場合、「httpd_t ドメインは、http_sys _content_t というラベルを持つファイルに対して read システムコールと write システムコールの利用ができる」という意味になる。

(4) リソースとラベルの関連付け

リソースとラベルの関連付けは次のように行う。

R **system_u:object_r:***L*

R はリソースの名前、*L* はリソースのラベルである。

(5) マクロ設定・属性設定

一括した設定を行うため、マクロ設定と属性設定が用意されている。マクロ設定の例を挙げる。

allow *httpd_t contents_t r_file_perms*;
define(`*r_file_perms*',`{ *read getattr lock ioctl access poll* }')

この中には r_file_perms というマクロが使われており、open・read システムコールでファイルを読み込むのに必要な 6 つのパーミッションに展開される。

属性設定の書式は次のようになる。

attribute *A*;
type *D A*;

A は属性名、*D* はドメインである。一行目で属性を宣言し、二行目でドメインに属性を持たせている。属性に対して許可されたパーミッションは、その属性を持っているドメインに対しても許可される。複数のドメインに一括して設定を行うことができる。

2.3 SELinux 設定の問題点

SELinux の設定を困難にしている原因として以下の 6 点が挙げられる。

(1) リソースとラベルの関連付けが分かりにくい

SELinux の設定を行う場合は、リソースを指定して設定を行うのではなく、リソースのラベルを指定して設定する。設定の度にリソースとラベルの関連付けを参照したり、また新規ラベルの定義時に、既存の関連付けに影響がないようにするための手間が生じる。

(2) パーミッション設定項目が膨大である

SELinux の設定言語では、例えばファイルに対するアクセス制御設定をする場合、ファイルの種別として

ファイル･ディレクトリ･デバイスファイルなど 7 種類の種別があり、それらの種別ごとに 17 種類ものパーミッションを設定可能である。このように、ファイルに関する設定だけでも 7×17=119 種類も設定可能項目があり、設定項目が膨大である。

(3) デフォルト設定ファイルでは、設定が散在し、設定状況が分かりにくい

SELinux を設定する場合、デフォルトで用意されている設定ファイルをカスタマイズする形で設定する。なぜなら、設定すべき項目が膨大なので、一から設定することは現実的でないからである。このデフォルト設定ファイルでは、設定が様々なファイルに散在しているため、あるドメインの設定を確認することが難しい。例えば、マクロや属性は様々なファイルに配置されている上に、それらは入れ子状に使われている。また、設定ファイルは基本的にはドメインごとに分かれているが、分割が徹底せず、あるドメインの設定が別のファイルで行われていたりする。

(4) リソースにアクセス可能なドメイン一覧が分かりにくく、設定ミスの発見が困難である

SELinux の設定は、ドメインに対してアクセス可能なリソースのラベルを指定する。したがって、リソース側から見たアクセス可能なドメインは、設定ファイルを一見しただけでは分からない。このような状況では、重要なリソースに余計なドメインがアクセス可能になっていても、発見することは難しい。

(5) ドメイン遷移の状況が分かりにくい

設定の安全性を調べるためには、あるドメインからどのようなドメインに遷移できるのかを知る必要がある。しかし、あるドメインからどのようなドメインに遷移できるのかを知るためには、設定ファイルから type_transition 文を見つけ出し、遷移先のドメインをたどっていく必要があり、一目でドメイン遷移状況は分からない。

(6) 設定方法が変化する

SELinux 開発プロジェクトは進行中のプロジェクトであるため、バージョンアップの度に設定方法の変化が起こる。例えば、設定言語の文法の変更やマクロや属性の改廃･追加である。そのため、管理者は設定方法の変化に追随する負担を負うことになる。

3 SELinux アクセス制御ポリシ設定システム

SELinux の設定を容易に行うため、SELinux アクセス制御ポリシ設定システムを試作した。その構成と機能を述べる。

3.1 システムの設計方針

設定ツールの設計をする際には、2.3 節の問題点を解決することが最も重要となる。したがって、以下の 6 点が設定ツールの設計方針となる。

(1) リソースを直接指定した設定ができること
(2) パーミッション設定項目を少なくすること
(3) あるドメインに関係する設定は一箇所に集中させること
(4) あるリソースにアクセス可能なドメインが分かるようにすること
(5) ドメイン遷移の状況を分かりやすくすること
(6) 設定方法の変化があっても設定者には意識させないこと

3.2 システムの構成

前節で述べた設計方針に準拠するために、SELinux 設定言語を簡略化した独自の設定言語(以下中間言語と呼ぶ)及び中間言語を SELinux 設定言語に変換するコンパイラ（以下中間設定変換コンパイラと呼ぶ）を設け、中間言語による設定を表示･編集するための GUI プログラムを設ける。ユーザのニーズによって最適な GUI 環境は異なるので、開発者はユーザのニーズに応じて様々な GUI プログラムを開発することが考えられる。したがって、開発工数を削減するために GUI プログラムのロジックは単純なものにする必要がある。GUI プログラムのロジックを単純にするために、中間言語を前節の設計方針をできる限り満たすように設計し、満たせなかった部分を GUI によって補う。設定ツールの全体構成を図 3.1 に示す。処理の流れは以下のようになる。

(1) ユーザは GUI プログラムにアクセスする。
(2) GUI プログラムはユーザの操作に応じて、中間設定ファイルを編集する。
(3) 設定が終わったら GUI プログラムは、中間設定変

換コンパイラを起動する。

(4) 中間設定変換コンパイラは、中間設定を読み込み、SELinux の設定ファイルを出力する。

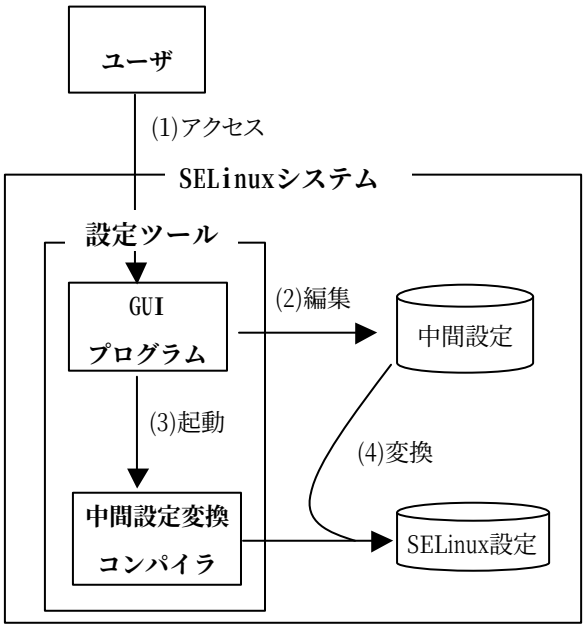

図 3.1 設定システムの構成

3.3 中間設定変換コンパイラ

3.3.1 中間言語の機能

中間言語には以下のような四つの機能がある。

(1) リソースを直接指定した設定

図 3.2 は中間言語で記述した httpd_t ドメインの設定例である。2 行目は httpd_t ドメインに対する設定であることを示している。以下の文でアクセス制御の設定をしている。3 行目は、/etc/httpd というファイルに対して読み込みを許可し、4 行目は tcp の 80 番ポートの利用を許可している。このように直接リソースを指定して設定する。

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

図 3.2 中間言語による設定の例

(2) パーミッション設定項目の絞込み

設定対象とパーミッションを使用頻度が高いものに統合することによる設定の簡素化を図った。例えば、ファイルの場合、4 種類のファイル種別と 17 種類のパーミッションがあったが、ファイルの種別をなくし、読み・書き・実行及びディレクトリ一覧取得という 4 種類のパーミッションにまとめた。同様に他のリソースについてもパーミッション設定項目を絞った。

(3) ドメインに対する設定を一箇所に集中

ドメインに対する設定は{}で囲んだ部分に記述する文法を採用した。これにより、ドメインに対する設定を集中させる。図 3.2 の例では、httpd_t ドメインに対する設定は{}に囲まれた部分にのみ存在する。

(4) global ドメイン設定

設定の利便性を高めるために、global ドメイン設定という設定を設けた。global ドメイン設定では、global という名前のドメインに対するアクセス制御設定は、全てのドメインに反映される。これにより一括した設定が可能になり設定の利便性が高まる。

3.3.2 中間設定言語から SELinux 設定言語への変換機能

変換処理の基本的な流れは以下の(1)から(5)のようになる。

(1) ラベル名の生成

コンパイラは中間設定を読み込むと、リソースの名前からラベルを自動的に生成する。

(2) ドメインの宣言

ドメインを type 文で宣言し、同時にドメインに global 属性を持たせる。これにより、ドメインは global 属性に対して行われた設定を継承する。

(3) リソースのラベルの宣言

(1)で生成したラベルを type 文で宣言する。

(4) ドメインとリソース間のパーミッションの設定を出力

各ドメインについて、allow 文を使ってリソースのラベルを指定してパーミッションを出力する。ここで、global ドメイン設定で許可されているリソースについては、allow 文のドメインの部分を global として出力する。(2)でドメインに global 属性を与えているので、この設定は全てのドメインに反映される。

(5) ラベルとリソースの関連付けを出力

(1)で計算したラベルとリソースの関連付けを出力する。

設定の変換例を図 3.3 に示す。左が中間言語で記述された設定、右が変換後の SELinux 設定言語による設定である。(1)の処理で/lib、/var/www というファイル名から lib_t、var_www_t というラベル名を計算し、(2)から(5)の処理で、図の右のような SELinux 設定を出力している。

{
domain *global*;
allow */lib* r;
}
{
domain *httpd_t*;
allow */var/www* r;
}
	#(2) ドメインの宣言
	attribute *global*;
	type *httpd_t,global*;
	#(3) リソースのラベルの宣言
	type *lib_t*;
	type *var_www_t*;
	#(4)パーミッション付与
	allow *global lib_t* :*r_file_perms*;
	allow *httpd_t var_www_t file*:*r_file_perms*;
	#(5) ラベルとリソースの関連付け
	/lib(/.)* *system_u*:*object_r*:*lib_t*
	/var/www(/.)* *system_u*:*object_r*:*var_www_t*

図 3.3 設定の変換例†

3.4 GUI プログラムの機能

GUI プログラムの機能には、ドメインから見た権限の表示・編集機能・リソースから見たドメインの持つ権限の表示機能及びドメイン遷移の表示・編集機能がある。

(1)ドメインから見た権限の表示・編集機能

ドメインから見た権限の設定状況を表示・編集できる。この機能の例を図 3.4 に示す。図の例では httpd_t というドメインが/var/www 以下のファイルに対して持つ権限がチェックボックスを用いて示されている。r・w・x・s の列はそれぞれ読み・書き・実行・ディレクトリサーチの権限を表す。チェックボックスは global ドメインで与えられた権限も考慮されて表示される。

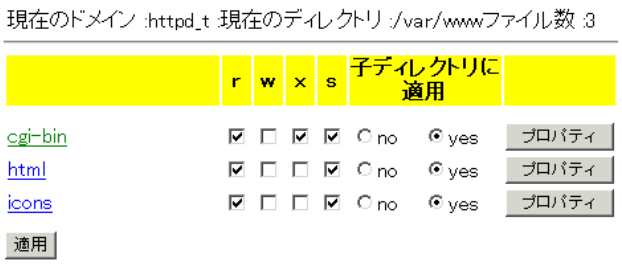

図 3.4 ドメインから見た権限の表示・編集画面

† 図右のパーミッションの付与で 1 種類のファイル種別に対する allow 文しか示していないが、実際は 4 種類のファイル種別に対して allow 文を出力する。

(2) リソースから見た権限の表示機能

あるリソースに対してどのようなドメインがアクセスできるのかを見ることができる。その例を図 3.5 に示す。この例では/var/www/html というディレクトリにアクセスできるドメインの一覧が表示されている。

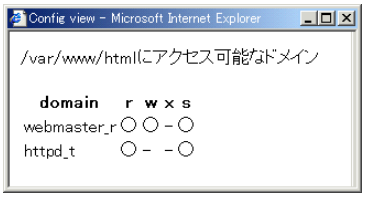

図 3.5 リソースから見た権限の表示例

(3) ドメイン遷移状況の表示・編集機能

ドメイン遷移の表示・編集機能は図 3.6 のようになる。図 3.6 の左側ではドメイン遷移の様子をツリー状に表示している。これを見ることにより、あるドメインがどのようなドメインに遷移できるのか一目で知ることができる。また、右側の画面でドメイン遷移を編集することができる。

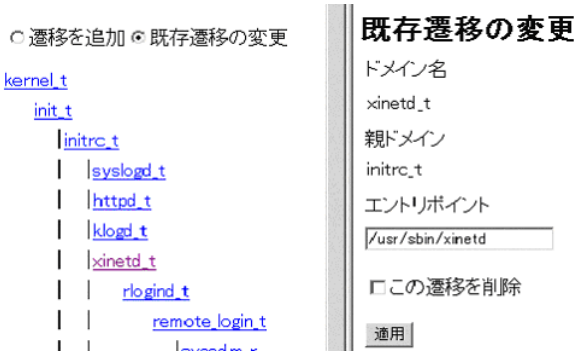

図 3.6 ドメイン遷移の表示・編集画面

4 システムの有効性評価

試作した設定ツールで、2.3 節(1)～(6)の問題点がどの程度まで解決できているか、新たな問題点はないかを検討する。

(1) リソースとラベルの関連付けが分かりにくい問題

中間言語によって、リソースを直接指定して設定をすることができるようになった。ただし、プロセスによって動的に生成されるファイルには対応できていない。動的に生成されるファイルとは、プロセスが起動中に生成・消去されるファイルである。通常はファ

イルが消去され、再生成された時は以前のラベル情報は失われてしまう。SELinux ではこのようなファイルには特殊なラベル付け設定を行うようにしている。このようなファイルに対しては、中間言語ではラベルを指定した設定を行わざるをえない。しかし、この問題は GUI でラベルに関連付けられているファイル一覧を示すことで、カバーできると考えられる。

(2) パーミッション設定項目が膨大である問題

中間言語によって、パーミッションの設定項目は減少した。従来はファイルだけでも 117 種類のパーミッション設定項目があったが、それを 4 種類に絞り込み、設定項目が大きく減少した。ただし、細かいパーミッションは設定できない。この問題を解決するには、中間言語を拡張し、細かい設定を行う文法を追加し、GUI で詳細設定の表示と絞り込んだ設定の表示を切り替えられるようにすれば対応できる。

(3) デフォルト設定ファイルでは、設定が散在し、設定状況が分かりにくい問題

設定ツールでは、あるドメインの持つ権限を知るためには、図 3.4 のような GUI の画面を見るだけで設定状況を知ることができるので、この問題は解決できた。

(4) リソースにアクセス可能なドメイン一覧が分かりにくく、設定ミスの発見が困難である問題

図 3.5 のように、あるリソースにアクセス可能なドメインを表示することができるようになったので、この問題は解決できた。

(5) ドメイン遷移の状況が分かりにくい問題

図 3.6 の画面を見るだけでドメイン遷移の状況を知ることができるようになったので、この問題は解決できた。

(6) 設定方法が変化する問題

SELinux 設定言語での設定方法が変わっても中間言語は変わらないので、設定者は設定方法の変化についていく必要はなくなった。

このように、試作した設定ツールにより 2.3 節(1)～(6)までの問題点はほぼ解決され、SELinux の設定は容易になったといえる。

5 結論

SELinux のアクセス制御ポリシ設定を困難にしている要因を考察し、SELinux アクセス制御ポリシ設定システムを試作した。本システムにより、リソースを直接指定した設定・設定項目の減少・設定状況の一元的把握・SELinux のバージョンに依存しない設定を実現した。その結果、SELinux の設定の操作性が向上し、設定に要するコストが減少すると考えられる。ただし、設定漏れが起きる可能性は否定できない。現システムでは図 3.5 の画面を用い、目視で設定を確認する。今後の課題は設定の安全性を自動的に検証する方式の開発である。

参考文献

[1]Security-Enhanced Linux
URL=http://www.nsa.gov/selinux
[2]P.Loscocco and S.Smalley: Integrating Flexible Support for Security Policies into the Linux Operating System: the Proceedings of the FREENIX Track of the 2001 USENIX Annual Technical Conference (2001)
[3]P.Loscocco and S.Smalley: Meeting Critical Security Objectives with Security-Enhanced Linux: Proceedings of the 2001 Ottawa Linux Symposium (2001)