テクニカルノート

設定ツールによるSELinuxアクセスパーミッション統合の安全性評価

田端利宏[†] 末安克也[††,☆] 櫻井幸一[†]

SELinuxにおけるポリシの設定補助ツールとして，SELinux Policy Editorが開発されている．このツールは，SELinuxのアクセス制御ポリシの設定項目を一部統合することによって，設定の簡易化を実現している．しかし，設定項目の統合によって，SELinuxの安全性が損なわれる可能性がある．本稿では，Webサーバソフトウェアである Apacheに関して，設定の簡易化がアクセス制御ポリシの設定結果に及ぼす影響を調査し，設定項目統合の安全性を評価した結果を報告する．

On the Security of Integration of SELinux Access Permissions

Toshihiro Tabata,[†] Katsuya Sueyasu[††,☆] and Kouichi Sakurai[†]

SELinux Policy Editor is a configuration tool for SELinux. As a part of its support of configuration, this tool simplifies the configuration of SELinux by integrating configuration items. However, the integration of configuration items may harm the fine-grained access control of SELinux. In this paper, we examine the effects of the simplification on access control policy and report the evaluation of the security about Apache web server.

1. はじめに

計算機ネットワークの普及にともない，クラッカによる侵入攻撃やウイルスへの感染の危険性が増大してきている．これらの脅威から計算機の被害を最小限に抑えるために，オペレーティングシステム（OS）レベルでのセキュリティの向上が求められている[1]．このため，Linuxのセキュリティ機能を強化したSecurity-Enhanced Linux（SELinux）に注目が集まっている．

SELinuxとは，米国家安全保障局（NSA）がLinuxをベースに開発しているセキュアOSである．SELinuxは，基となるLinuxをはじめとした多くのOSで採用されている任意アクセス制御（Discretionary Access Control，DAC）[2]に加え，強制アクセス制御（Mandatory Access Control，MAC）[2]を採用することで，セキュリティの向上を図っている．また，高いセキュリティを実現できる代償として，SELinuxはLinuxに比べて複雑なアクセス制御設定を必要とする．

そこで，SELinuxの設定に要する労力を軽減する目的で，設定補助ツールが開発されている[3)~5)]．そのツールの1つであるSELinux Policy Editor（以降，Policy Editorと略す）は，設定の簡易化の一環として，アクセス制御設定の項目を一部統合している．しかし，設定項目の統合により，計算機システムのセキュリティが損なわれる恐れがある．

本稿では，Webサーバソフトウェアである Apacheの利用に必要なファイルアクセス制御設定を対象として，Policy Editorによる設定項目の統合がシステムのセキュリティに与える影響を明らかにする．

2. Security-Enhanced Linux

本章では，SELinuxが持つ機能やそれを実現する機構について簡単に説明する．

SELinuxで実現されている機能を以下に示す．

(1) MAC
(2) Type Enforcement（TE）
(3) Role Based Access Control（RBAC）[2]

SELinuxのアクセス制御機構は，基となるLinuxのアクセス制御機構の外側に，上記の機能を持つアク

† 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University
†† 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, School of Engineering, Kyushu University
☆ 現在，株式会社ヒューマンテクノシステム
Presently with HumanTechnoSystem

セス制御機構を追加することで実現されている．つまり，両方の機構からアクセス許可を得てはじめてアクセスを行うことができる．アクセス制御ポリシは，アクセス制御設定ファイルに記述され，設定ファイルの編集権限を持つ者のみ設定を変更できる．また，アクセスは，それが設定ファイル内で定義されている場合にのみ許可され，定義されていないアクセスはいっさい許可されない．

ファイルのパーミッションは 17 種類，ディレクトリのパーミッションは，22 種類存在する[6]．次に，後半の説明で関係するディレクトリアクセスの search パーミッションと read パーミッションについて説明する．search パーミッションは，ディレクトリアクセス専用のパーミッションである．あるファイルやディレクトリにアクセスするには，そのファイルやディレクトリに対するアクセス権限のほかに，そのパス上にあるすべてのディレクトリの search パーミッションが必要になる．また，ディレクトリアクセスにおける read パーミッションは，そのディレクトリ内にどんなファイルやディレクトリが存在するか知るために必要なパーミッションである．

3. SELinux Policy Editor

SELinux の設定を補助するツールである SELinux Policy Editor の主な機能について説明する．

(1) Graphical User Interface（GUI）を通して設定できる．これにより，設定を視覚的に把握できる．

(2) 独自の中間設定言語を採用している．利用者は GUI を通して，中間設定言語で記述された中間設定ファイルを編集する．それから Policy Editor 側で，中間設定ファイルを現在の SELinux のバージョンで有効な設定ファイルに変換する．中間設定言語は SELinux のバージョンに依存しないので，利用者は SELinux のバージョン変化を意識する必要がない．

(3) Policy Editor では，ファイルアクセスの 17 種類のパーミッションのうち 12 種類を 4 種類に統合し，設定を簡易化している．また，ディレクトリアクセスのパーミッションも同様に 4 種類に統合されている．統合前後のパーミッション間の対応関係を**表 1** と**表 2** に示す．r は読み取り（read），w は書き込み（write），x は実行（execute），s は探索（search）を意味する．

なお，この Editor では，中間設定言語を利用するため，統合パーミッションでのみ設定できる．

4. Example Policy

Example Policy[7] は，ポリシ定義ファイルの例として，SELinux とともに配布されている．SELinux のアクセス制御設定は複雑であるため，サンプルである example policy の構築に，数種類のマクロを用いた設定の簡易化が利用されている．たとえば，example policy のファイルアクセスに関する read パーミッションのうち，90%以上がマクロによって設定されている．また，マクロを用いずに個別に設定するべき部分もわずかながら存在する．なお，SELinux におけるマクロの使用は任意である．

表 1 ファイルアクセスの統合パーミッション
Table 1 Integration of permissions for file access.

統合パーミッション	対応する SELinux のパーミッション
r	read, getattr, ioctl, lock
w	write, setattr, append, create, unlink, link, rename
x	execute
s	getattr

表 2 ディレクトリアクセスの統合パーミッション
Table 2 Integration of permissions for directory access.

統合パーミッション	対応する SELinux のパーミッション
r	read, getattr, ioctl, lock
w	write, setattr, append, create, unlink, link, rename, add_name, remove_name, reparent, rmdir
x	execute
s	getattr, search, read

5. 簡易化手法の比較

SELinux のアクセス制御の設定には，マクロとパーミッション個別の設定を併用できる．このため，マクロを使用しても，アクセス制御設定の表現に制限は生じない．一方，Policy Editor を利用すると統合パーミッションを用いてのみ設定可能であるため，設定できるパーミッションの粒度に制限が生じる．

本章では，Apache のファイルとディレクトリアクセスを対象として，Policy Editor の統合パーミッションの問題点を明らかにする．なお，ファイルやディレクトリ以外について，セキュリティ上致命的な問題が生じないことを文献 8) で報告している．

5.1 パーミッション粒度の比較

ファイルおよびディレクトリに関するマクロと，統合パーミッションの関係を**表 3** と**表 4** に示す．プロセスが動作するのに必要なパーミッションが不足すると，プロセスの正常な動作が妨げられる．このため，表 3 と表 4 では，各マクロで定義されているパーミッションをすべて含む，最小の統合パーミッションの組合せで各マクロを表現した．表 3 と表 4 から，次の

表 3 ファイルに関するマクロと統合パーミッションの関係
Table 3 Correspondence between macros and integrated permissions for file access.

マクロ	統合パーミッション	余分なパーミッション
x_file_perms	x	—
r_file_perms	r s	—
rx_file_perms	r x	—
ra_file_perms	r w s	setattr, create, link, unlink, rename, write
rw_file_perms	r w s	setattr, create, link, unlink, rename
create_file_perms	r w s	—

表 4 ディレクトリに関するマクロと統合パーミッションの関係
Table 4 Correspondence between macros and integrated permissions for directory access.

マクロ	統合パーミッション	余分なパーミッション
r_dir_perms	r s	—
ra_dir_perms	r w s	setattr, create, link, unlink, rename, reparent, rmdir, remove_name
rw_dir_perms	r w s	setattr, create, link, unlink, rename, reparent, rmdir
create_dir_perms	r w s	—

ことが分かる．

(1) rw_file_perms と rw_dir_perms マクロ（以降，write マクロと呼ぶ）を過不足なく表現できる統合パーミッションの組が存在しない．つまり，SELinux Policy Editor では，write マクロが create_file_perms または create_dir_perms マクロ（以降，create マクロと呼ぶ）と同一視される．したがって，ファイルへの書き込みを許すとファイル情報の変更も許すことになる．

(2) ra_file_perms と ra_dir_perms マクロ（以降，append マクロと呼ぶ）を過不足なく表現できる統合パーミッションの組が存在しない．つまり，Policy Editor では，追記の設定ができない．

以上のことから，Policy Editor の統合パーミッションを利用すると，ファイル情報の変更をともなわない書き込みのみの設定や追記のみの設定の概念が失われる．

また，Policy Editor では，統合された各パーミッションを個別に設定できないため，パーミッションの粒度が荒い．そこで，Example Policy で，個別のパーミッションで設定されている部分を統合パーミッションで設定した場合の安全性を調査した．調査の結果，ディレクトリの search パーミッションを単独で設定できないことにより，セキュリティに影響を及ぼす可能性があることが分かった．具体的には，統合パーミッション “s” により，search と read パーミッションが統合され，ディレクトリ内の構成ファイル名が漏洩する可能性がある．

5.2 考　　察

ここでは，攻撃者はすべてのドメインの権限を得ることができるものと仮定し，未知の脆弱性に対する安全性について述べる．なお，現実にも，httpd プロセスの権限で任意のコードを実行される可能性がある Apache の脆弱性が報告されている[9]．

5.2.1 パーミッション統合の影響範囲

5.1 節で述べた問題の影響を受けるファイルやディレクトリの範囲について述べる．書き込みや追記の概念が失われることで影響を受けるファイルやディレクトリは，write マクロまたは append マクロでパーミッションを与えられていて，create マクロでパーミッションを与えられていない以下のファイルやディレクトリである．

- キャッシュディレクトリ
- ログファイル格納ディレクトリ
- CGI スクリプト格納ディレクトリ
- CGI スクリプトがアクセスする追記専用のファイルおよびディレクトリ

なお，同じファイルやディレクトリに対して，他のドメインが create マクロによる権限を得ている場合，上記の影響は無視できる．これは，攻撃者がより強い create マクロによる権限を取得できるためである．

また，search と read パーミッションの統合により，Example Policy で search パーミッションが個別に設定されていた以下のディレクトリに影響がある．

- /boot ディレクトリ
- ユーザのホームディレクトリ

5.2.2 write マクロと append マクロに対する余剰なパーミッションの影響

Apache の example policy では，キャッシュディレクトリおよびログディレクトリに対しては，write 権限が与えられている．キャッシュディレクトリは/var/cache であり，Apache が proxy として用いられる場合には直下に httpd ディレクトリが作成され，そこにキャッシュが蓄えられる．また，/var/cache には 1 つ上の/var ディレクトリと同じタイプが付与されているので，/var に対しても同じ権限が与えられていることになる．

Policy Editor による設定の簡易化の結果，rmdir（ディレクトリ削除）などの余分なパーミッションが与えられる．ディレクトリを削除するには，ディレクトリ内のファイルをすべて削除しなければならない．しかし，攻撃者が利用できるドメインは，/var 以下のほとんどのファイルに対してアクセスする権限を持たない．したがって，この余分なパーミッションによって可能になるのは，空になった Apache 関連のディレクトリを削除することである．これはログディレクトリに関しても同様である．

CGI スクリプトを格納しているディレクトリや，CGI スクリプトが取り扱う追記専用ファイルおよびディレクトリに対しては，いずれも追記権限が与えられている．Policy Editor による設定の簡易化の結果，攻撃者による追記専用のファイルやディレクトリの改ざんや削除が可能になる．ただし，CGI スクリプト本体など，もともと書き込み不可であるファイルを削除することはできない．

5.2.3 余剰な read パーミッションによる影響

/boot ディレクトリ以下には，OS の起動に必要なファイルが格納されている．この/boot に対する設定は，一般的なドメインに共通して行われる．Policy Editor による設定の簡易化の結果，攻撃者はこのディレクトリ内にあるファイルの一覧を得られるようになる．ただし，ファイルの内容を読み取ることも改竄することもできない．ユーザのホームディレクトリに関しても同様に，それ以下のディレクトリやファイルの一覧を得られるようになる．

5.2.4 考察のまとめ

SELinux Policy Editor による Apache の example policy の簡易化がシステムのセキュリティに与える影響として考えられるのは，

- Apache が扱う追記専用ファイルおよびディレクトリの破壊や改竄の可能性，
- ユーザのホームディレクトリなど，一部のディレクトリの構成ファイル名が漏洩する可能性，

である．

ファイルの存在を知られることがシステムのセキュリティを損なう場合には，search と read パーミッションの統合が問題となる．たとえば，どこからもリンクされていないファイルが Web サーバに存在する場合，統合パーミッションによってディレクトリに対する read パーミッションが余分に付加されると，第三者に対してそのファイルの存在を知る権限が与えられることになる．これにより，Web サーバ経由のアクセスで，そのファイルの情報が漏洩する可能性がある．

6. おわりに

本稿では，Apache を例として，SELinux の設定を簡易化する SELinux Policy Editor の統合パーミッションの安全性を評価した．評価の結果，統合パーミッションにより，設定を簡易化できるものの，Policy Editor の統合パーミッションには安全性に問題があることを明らかにした．今後の課題として，システムのセキュリティを損なわないパーミッションの統合方法に関する検討がある．

参考文献

1) 総務省：セキュア OS に関する調査研究会報告書 (2004). http://www.soumu.go.jp/s-news/2004/040428_1.html
2) 土居範久（監修），佐々木良一，内田勝也，岡本栄司，菊池浩明，寺田真敏，村山優子（編）：情報セキュリティ事典，共立出版 (2003).
3) 日立ソフトウェアエンジニアリング株式会社：SELinux Policy Editor (2003). http://www. selinux.hitachi-sk.co.jp/tool/selpe/selpe-top.html
4) 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，2003 年暗号と情報セキュリティシンポジウム（SCIS2003）予稿集，Vol.II, pp.831–836 (2003).
5) TresysTechnology, Security-Enhanced Linux research (2003). http://www.tresys.com/selinux/index.html
6) 日立ソフトウェアエンジニアリング株式会社：オペレーティングシステムのセキュリティ機能拡張の調査，IPA/ISEC 情報セキュリティ関連の調査・開発に関する公募．http://www.ipa.go.jp/security/fy13/report/secure_os/secure_os.html
7) 日立ソフトウェアエンジニアリング株式会社，サン・マイクロシステムズ株式会社：セキュアなインターネットサーバー構築に関する調査，IPA/ISEC 第二回情報セキュリティ関連の調査・開発に関する公募 (2003). http://www.ipa.go.jp/security/fy14/contents/trusted-os/guide.html
8) 末安克也，田端利宏，櫻井幸一：SELinux アクセス制御設定項目の安全な統合方法に関する考察，2004 年暗号と情報セキュリティシンポジウム（SCIS2004），Vol.I, pp.287–292 (2004).
9) CERT/CC, CERT Advisory CA-2002-17 Apache Web Server Chunk Handling Vulnerability (2002). http://www.cert.org/advisories/CA-2002-17.html

(平成 16 年 10 月 18 日受付)
(平成 17 年 2 月 1 日採録)