Linux Security Moduleを用いたPrivacy-aware OS Salviaの構築

鍛治 輝行 † 岩永 真幸 †† 毛利 公一 †
† 立命館大学情報理工学部 †† 立命館大学大学院理工学研究科

1 はじめに

近年，プライバシ情報が電子化され，その情報が計算機で管理されている．このプライバシ情報が，記憶媒体やネットワークを通じて漏洩する事件が多発している．個人情報の保護は，2005年4月1日から施行された個人情報保護法により，保護が義務付けられている．暗号化，認証，侵入検知といった技術により，セキュリティ侵害における情報漏洩を防ぐ事が可能である．しかし，日本ネットワークセキュリティ協会の情報セキュリティインシデントに関する調査報告書[1]における，情報漏洩比率によると，セキュリティ侵害といった外部からの攻撃による漏洩の比率は低い．漏洩原因の多くを占めるものが，ユーザの管理ミス，紛失，誤操作といった正当なアクセス権限を持つものによる漏洩である．暗号化，認証，侵入検知といった技術では，このような正当なアクセス権限を持つ者が引き起こす漏洩を防ぐことが困難である．以上の背景により，我々は，正当なアクセス権限を持つ者による情報漏洩を防止するために，Privacy-aware OS *Salvia*[2]の開発を行っている．*Salvia*を用いることにより，正当な権限を持つユーザによる情報の漏洩を防止する事が可能となる．

これまで，*Salvia*はLinuxカーネルを改変する形で開発してきたが，より移植性を高くし，ユーザによる*Salvia*の導入を容易にすることを目的として，Linux Security Module[3](以下，LSMと略す)に基づいて開発を行っている．以下，本稿では，2章で*Salvia*の概要について述べ，3章でLSMの概要について述べる．4章でLSMに対応した*Salvia*の構成について述べ，5章で本稿のまとめとする．

2 *Salvia*の概要

*Salvia*は，アクセス制御を行い，情報の漏洩を防止するOSである．図1に，*Salvia*のデータ保護モデルを示す．*Salvia*は，ファイルを保護の単位としており，保護対象ファイルをオープンしたプロセスを制御の対象としている．データ作成者は，保護対象ファイルごとに保護ポリシを作成する．保護ポリシには，対応する保護対象ファイルをオープンしたプロセスに課す制約と，その制約を課す条件を記述する．*Salvia*では，プロセスを制御するための条件として，コンテキストの記述を可能としている．プロセスを制御する際に用いるコンテキストには，端末の位置，現在の時刻，ユーザID，過去の動作履歴などがある．*Salvia*は，この保護ポリシに基づき，制御対象プロセスに対し，ファイル，ソケット，パイプなどの計算機資源へのアクセスを制御することにより，データの漏洩を防止する．

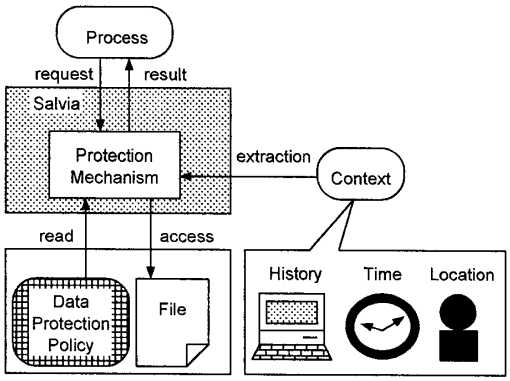

図1 *Salvia*のデータ保護モデル

3 LSMの概要

LSMは，カーネルにセキュリティ機能を拡張するためのフレームワークであり，Linuxカーネル2.6シリーズ以降，正式に採用されている．LSMが採用されているカーネルには，ソースコード中のファイルシステム操作の可否などのセキュリティ・チェックポイントで，コールバック関数呼び出しが挿入されている．対応するコールバック関数を用意することで，ファイルシステム操作の可否の判断に，より詳細なチェックを加える，といったことが可能となる．

LSMに基づき作成したセキュリティモジュールは，LSMが採用されているLinux2.6シリーズ以降のカーネルであれば，移植が可能となる．現在の*Salvia*は，Linuxカーネル2.6.8を基に実装を行っており，ソースコードに変更を加えることにより開発を行っている．LSMに基づき*Salvia*を構築することで，LSMが採用されているカーネルへの移植が可能となる．

A construction of Privacy-aware OS Salvia based on Linux Security Module
Teruyuki Kaji†, Masayuki Iwanaga††, and Koichi Mouri†
†College of Information Science and Engineering, Ritsumeikan University.
††Graduate School of Science and Engineering, Ritsumeikan University.

4 LSMに対応した *Salvia* の構築

4.1 現在の *Salvia* の処理の流れ

現在のSalviaにおいて，プロセスのアクセスを制御する処理の流れを，以下に述べる．

- プロセスがファイルをオープンする際に発行するopenシステムコールのフックを行う．
- open対象ファイルに対応する保護ポリシを取得し，プロセスの動作の監視を開始する．
- プロセスが計算機資源に対し，アクセスを行う際に発行するシステムコールをフックし，保護ポリシに基づき計算機資源へのアクセスを制御する．

以上の処理を行うことにより，保護ポリシに基づき，ファイルの保護が可能となる．*Salvia* では，アクセスの種類を，以下に述べる4つに分類を行い，この分類に基づきアクセスの制御を行っている．

- read グループ ファイルからのデータの読み出し
- write グループ ファイルへのデータの書き込み
- send_local グループ 同一計算機上の他のプロセスのデータ領域への書き込み
- send_remote グループ 他の計算機上のプロセスのデータ領域への書き込み

4.2 LSMを用いた *Salvia* の構築

LSMを用い *Salvia* を構築し，移植性を向上させるためには，カーネルソースに変更を加えずに開発する必要がある．しかし，現在の *Salvia* では，ソースコードに変更を加え，システムコールテーブルを書き換えることにより，システムコールのフックを行っている．そのため，LSMのフック関数を用い，カーネルに変更を加えることなく，前節で述べたような *Salvia* のアクセス制御を実現する必要がある．

以下に，LSMを用いた *Salvia* において，フックが必要な処理と，その際に用いるフック関数について述べる．

はじめに，ファイルのオープン時，その処理をフックする．この際，security_inode_permission 関数を用いる．ファイルをオープンする際は，そのファイルに対応するi-nodeへのアクセスが行われる．security_inode_permission 関数は，そのi-nodeへのアクセスの可否を判別する箇所に挿入されている．この時点で，オープン対象ファイルが，保護対象ファイルであるか調べる．保護対象であるか否かは，オープン対象ファイルに対応する保護ポリシの有無で判別を行う．現在，この保護ポリシは，保護対象ファイルのi-node拡張属性に格納している．i-nodeへのアクセスの可否を判別するとともに，i-node拡張属性に格納されている保護ポリシを取得することが，security_inode_permission 関数の時点で可能である．保護ポリシが存在した場合，その情報を取得し，プロセスへの制御を開始する．

次に，監視対象プロセスが計算機資源に対するアクセスをフックを行う．この際には，アクセスの種類ごとに異なるフック関数を用いる必要がある．アクセスの種類は，上記した *Salvia* における分類に従う．現在，read グループ，write グループのフックの実装が完了している．read グループ，write グループのフックには，同一のフック関数 file_permission 関数を用いる．read グループ，write グループは，共にファイルへのアクセスであるため，そのファイルへのアクセスの前に，パーミッションチェックを行う箇所のフック関数を用いることにより，アクセスのフックが可能となる．read グループと write グループは，file_permission 関数の引数である mask により区別が可能である．アクセスのフック後，ファイルのオープン時に取得した保護ポリシに基づき，アクセスの可否を決定する．send_locak グループ，send_remote グループのフックに関しては，現在調査中であり，今後の課題となる．

以上の処理を行うことにより，*Salvia* のアクセス制御と同様の処理が実現可能となる．

5 おわりに

本論文では，*Salvia* におけるアクセス制御を，LSMに対応する形で構築を行う手法について述べた．LSMのフック関数を利用し，処理のフックを行う箇所は，保護対象ファイルのオープンを行う箇所と，監視対象プロセスが計算機資源に対しアクセスを行う箇所である．プロセスが計算機資源へのアクセスをする際，適宜，LSMのフック関数を用いる事により，アクセスのフックが可能となる事を述べた．本手法により，移植性が向上し，導入が容易となる．今後の課題として，send_local グループ，send_remote グループのフックをし，アクセスの制御を行う必要がある．また，保護ポリシの取得を行い，保護ポリシに基づいたアクセスの制御を行う機能の実装をする必要がある．

参考文献

[1] NPO 日本ネットワークセキュリティ協会: JNSA 2007年情報セキュリティインシデントに関する調査報告書，2008.

[2] 鈴木 和久, 一柳 淑美, 毛利 公一, 大久保 英嗣: Privacy-Aware OS *Salvia* におけるデータアクセス時のコンテキストに基づく適応的データ保護方式, 情報処理学会論文誌: コンピューティングシステム, Vol. 47, No. SIG3, pp. 1-15, 2006.

[3] Chris Wright, Crispin Cowan, Stephen Smalley, James Morris, Greg Kroah-Hartman : Linux Security Modules:General Security Support for the Linux Kernel, In Proc. of the 11th USENIX Security Symposium, pp. 17-31, 2002.