Privacy-aware OS *Salvia*における データフローを主体としたアクセス制御手法

井田 章三† 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，計算機が普及し，個人情報は電子データとして保存，管理されている．それに伴ない，記憶媒体やネットワークを通じて個人情報が流出する事件が頻繁に起きている．情報漏洩事件は，デジタルデータを劣化なく高速にコピーできるという特徴に加え，記憶媒体の大容量化や情報通信の発達により，事件の発生頻度，規模ともに増大し，深刻な社会問題となっている．

文献[1]では，2007年に報道された個人情報漏洩事件の調査・分析を行った．文献によると，情報漏洩の原因として「紛失・置き忘れ」,「管理ミス」,「盗難」,「誤操作」の比率が最も高い．特に「管理ミス」は前年と比較するとその比率は大幅に高くなっている．つまり，企業や自治体は情報漏洩対策を行っているが，その現状は対策が不十分で，情報漏洩を発生させてしまっている．

また，情報漏洩事件の事例として,「紛失・置き忘れ」,「管理ミス」,「盗難」では，顧客などの個人情報を記憶媒体に保存し，外部に持ち出したことが原因となっていることが最も多い.「誤操作」では，顧客に他の個人の情報を電子メールに添付して送信してしまったという事例が報告されている．これらの事例は，正当なアクセス権限を持つユーザによる情報漏洩ということができる．暗号化などの既存のセキュリティ技術は，外部からの攻撃を防ぐことを目的とした技術であって，このような情報漏洩を防ぐことはできない．よって，このような情報漏洩を防止する技術の開発は急務だといえる．

以上の背景より，我々は，このような正当なアクセス権限を持つユーザによる情報漏洩を防ぐことを目的としたPrivacy-aware OS *Salvia*(以下 *Salvia*) を開発してきた．*Salvia*は，プロセスによるファイルやソケットなどの情報漏洩の可能性がある計算機資源に対する操作を制御することにより，情報漏洩を防ぐ．すなわち，計算機資源に対する操作の制御をプロセス単位としている．しかし，プロセス単位の制御では情報漏洩とは関係のない操作まで制御してしまい，プログラムの動作を必要以上に制限してしまう可能性があることが判明している．そこで解決策として，現在，データフローを主体としたアクセス制御手法の開発を行っている．

以下本稿では，2章で今まで開発を行ってきた*Salvia*の概要と課題を説明し，3章でその課題の解決策として提案するデータフローを主体としたアクセス制御手法について述べ，4章で本稿のまとめを述べる．

2 Privacy-aware OS Salvia

2.1 概要

データ保護ポリシ プライバシデータは，そのデータ提供者の意志により利用目的や提供範囲が異なる．そのため，プライバシデータの漏洩を防止するためには，各データごとに異なるアクセス制御を行う必要があり，かつそのアクセス制御にはデータ提供者の意志が反映されなければならない．よって*Salvia*では，ユーザは，ファイルごとにデータ保護ポリシを設定可能である．データ保護ポリシには，当該ファイルにアクセスしたプロセスに課す各計算機資源に対するアクセス制御と，その制御が行われる条件を記述する．制御条件には，時刻やIPアドレス，電波強度などのプロセスや計算機の状況を使用でき，これらは*Salvia*が，コンテキストとして抽出する．コンテキストにより*Salvia*は，特定の時間，場所のみ閲覧，変更，転送可能なファイルを設定できる．

データ保護方式 *Salvia*は，ファイルアクセスは，プロセスのopenシステムコールにより開始する点に着目し，ファイルオープンの前処理としてファイルに設定されているデータ保護ポリシを読み出し，ファイルをオープンしたプロセスをアクセス制御の対象とする．アクセス制御の対象となったプロセスでは，各計算機資源へのアクセスが制限される．計算機資源へのアクセスは，システムコールによって行われるため，*Salvia*は，システムコールの実行の可否をコンテキストとデータ保護ポリシから判断することにより，アクセス制御を実現する．

2.2 Salviaの課題

前述の通り*Salvia*は，保護ファイルオープン以降，プロセスにその保護ファイルのデータ保護ポリシに基づくアクセス制御を課す．しかし，この手法では，保護ファイルの関わらない処理までアクセス制御の対象としてしまう可能性がある．例えば，図1では，保護ファイルのopen以降に非保護ファイルにwriteをしようとしても，そのwriteシステムコールも，保護ファイルへの書き込みでないにも関わらず，禁止されてしまう．この動

An access control method based on data flow in Privacy-aware OS *Salvia*
Shozo Ida†, Masayuki Iwanaga††, and Koichi Mouri†
†College of Information Science and Engineering, Ritsumeikan University
††Graduate School of Science and Engineering, Ritsumeikan University

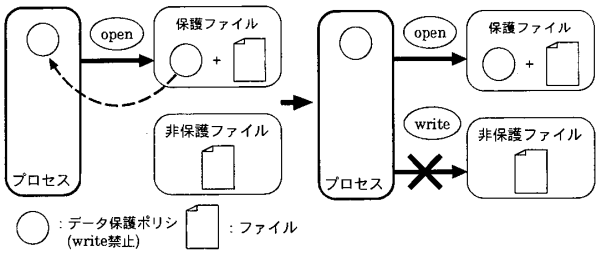

図 1　*Salvia* における問題点

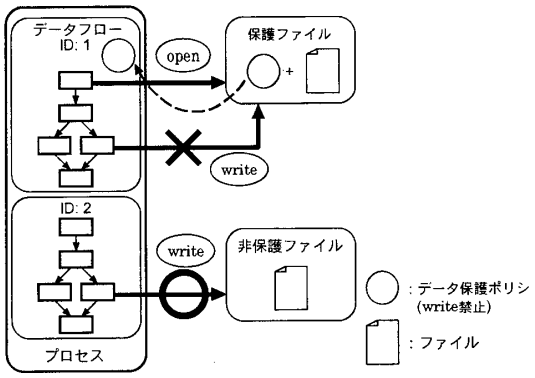

図 2　データフローを利用した *Salvia*

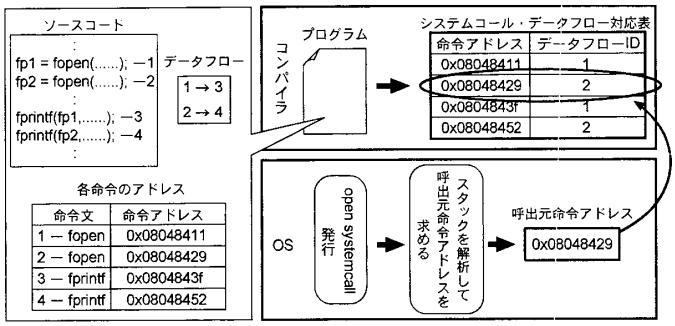

図 3　データフロー ID 特定の流れ

作により，長命なプロセスや扱うファイル数が多いプログラムでは，ユーザの意図した動作とならない可能性がある．例えば，一時ファイルを作成するプログラムやログファイルや設定を保存するプログラムが挙げられる．

この問題は，アクセス制御の主体がプロセスなので，プロセスが行う処理を，保護ファイルを扱うものとそうでないものとの区別がつけられないために発生する．よって問題を解決するには，プロセスが行う各処理を区別し，より粒度の細かいアクセス制御を行う必要がある．

3　提案手法

各処理の区別は，プログラム中のデータの流れを解析することによって実現させる．データの流れの解析は，コンパイラのデータフロー解析を利用する．データフロー解析は，コンパイラの最適化において行われるプログラム中の変数定義の流れの解析である．

保護ファイルが格納された変数定義の流れと，それ以外の変数定義の流れは，別のものとして扱うことができる．よって，データフローをアクセス制御の主体とすれば，プログラム中の各処理を区別することができる．その様子を図 2 に示す．プロセスが保護ファイルをオープンすると，オープンしたデータフローに保護ポリシに基づくシステムコールの制御が課せられ，非保護ファイルのデータフローはアクセス制御の対象とならないため，write を実行できる．データフローに基づくアクセス制御方式を以下に示す．

1. コンパイラが生成したデータフローに対し，データフロー ID を割り当てる．
2. 保護データ読み込み時，保護データが格納されている変数が属するデータフロー ID に対して，データ保護ポリシを適用する．
3. 監視対象のシステムコールが実行された場合，データが格納された変数が属するデータフロー ID に対して適用されたデータ保護ポリシの有無を確認する．
4. データ保護ポリシが適用されている場合は，保護データが含まれている可能性があるため，ポリシに従ってシステムコールの実行の可否を判断する．

OS が実行時に，上記の処理を行うには手順 2，3 において，システムコールが発行された時点で使用されている変数が属するデータフロー ID を特定できなければならない．解決方法として，システムコールの発行元の命令アドレスからデータフロー ID を求める．

具体的には，コンパイラにシステムコール・データフロー対応表を作成させる．システムコール・データフロー対応表とは，システムコール呼出しの大元となるユーザプログラムの命令アドレスとデータフロー ID を対応付けさせたものである．OS 実行時のシステムコールの発行元の命令アドレスを検出できれば，その命令アドレスとシステムコール・データフロー対応表に記録してあるシステムコールの呼出元命令アドレスを比較し，そのシステムコールで使われる変数が属するデータフロー ID を求めることができる．OS 実行時のシステムコール発行元命令アドレスの検出は，システムコール実行時にプロセスのスタックを解析することにより求める．データフロー ID 特定の流れを図 3 に示す．

4　おわりに

今後は，システムコール・データフロー対応表を作成するコンパイラの開発や，システムコールが発行されたアドレスを求めるためのスタックバックトレーサを *Salvia* へ実装していく予定である．

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.