3Y-5

データフローを主体としたアクセス制御を実現する *DF-Salvia* の設計と開発

井田 章三 † 河島 裕亮 † 樫山 武浩 †† 瀧本 栄二 ††† 毛利 公一 †††
† 立命館大学大学院理工学研究科 †† 立命館大学グローバル・イノベーション研究機構
††† 立命館大学情報理工学部

1 はじめに

近年，計算機が普及し，プライバシデータはデジタルデータとして保存，管理されている．これにより，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発し，深刻な社会問題となっている．文献 [1] では，2008 年に報道された個人情報漏洩事件の原因の調査について述べられており，アプリケーションの誤操作や個人情報の管理ミスなど，正当なアクセス権限を持つユーザのミスの比率が最も高いという結果が報告されている．具体的には,「顧客に他の顧客の情報を電子メールに添付して送信してしまう」「個人情報を記憶媒体に保存して外部に持ち出し，紛失する」といった事例が報告されている．このような情報漏洩は，暗号化や認証などの外部からの攻撃を防ぐことを目的としたセキュリティ技術で防ぐことが困難である．

以上の背景より，我々は，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オペレーティングシステム *DF-Salvia* の開発を行っている．*DF-Salvia* は，システムコールを監視・制御することにより，情報漏洩の可能性がある計算機資源に対する操作を制限し，データ保護を実現する．*DF-Salvia* は，このシステムコールの監視・制御をデータフロー単位で行う．これにより，計算機資源へのデータ書き込みを，書き込まれるデータの発生源に基づいて区別して制御でき，粒度の細かいアクセス制御が可能となる．

2 *DF-Salvia* のアクセス制御モデル

2.1 データ保護ポリシ

DF-Salvia は，データ提供者の意図する保護方針をデータ保護ポリシ (以下，ポリシ) として定義し，保護対象ファイル (以下，保護ファイル) と組にして管理する．ポリシには，保護ファイルのデータ (以下，保護データ) の読込み・書込みに課す制限と，その制限が行われる条件を記述する．条件は，コンテキストと呼ばれる計算機の状況を使用する．*DF-Salvia* は，保護データにアクセスするシステムコールに対し，ポリシに基づいたアクセス制御を課すことで，プライバシを考慮したデータ保護を実現する．例えば,「19:00〜7:00 はアクセスを禁止する」,「ハードディスク上のファイルは，自由にアクセスできるが，USB フラッシュメモリにコピーを禁止する」といったデータ保護を実現できる．なお，ポリシ記述法とコンテキストの詳細は文献 [2] を参照されたい．

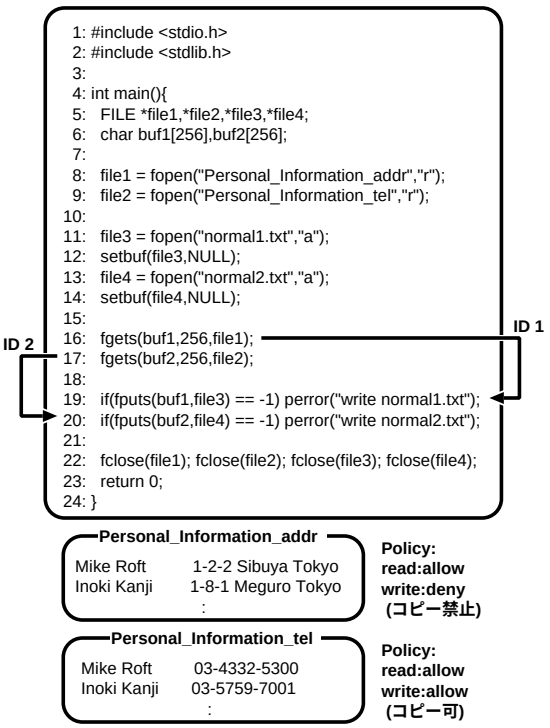

図 1 テストプログラム

表 1 データフロー ID 表

行数	命令アドレス	ID	最終使用点
16 fgets	0x080485d9	1	
17 fgets	0x080485f6	2	
19 fputs	0x0804860b	1	○
20 fputs	0x08048631	2	○

2.2 データフロー ID 表

保護データの書込みをポリシに基づき制御するためには，書き込まれようとしているデータがどのようなポリシで保護されるべきか特定する必要がある．そのためには，そのデータの発生源であるファイルを正確に識別する必要がある．*DF-Salvia* は，データフロー ID 表を用いて識別する．データフロー ID 表は次の要素から成る．

- プログラム中のライブラリ関数コール命令アドレス
- データフロー ID

Design and implementation of *DF-Salvia* which provides access control based on data flow
Shozo Ida†, Yusuke Kawashima†, Takehiro Kashiyama††, Eiji Takimoto†††, and Koichi Mouri†††
†Graduate School of Science and Engineering, Ritsumeikan University
††Ritsumeikan Global Innovation Research Organization
†††College of Information Science and Engineering, Ritsumeikan University

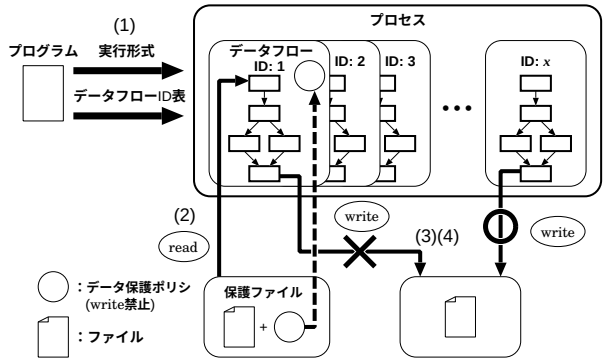

図 2　*DF-Salvia* のアクセス制御モデル

- 最終使用点 (後述) フラグ

データフロー ID 表は，定義使用連鎖というデータフロー情報を基に作成される．定義使用連鎖は，変数の定義点と，定義された変数がどの文で使用されるかをその変数の使用点の集合で表したものである．*DF-Salvia* では，ファイルからデータを読み込むライブラリ関数を定義点，そこで定義された変数を用いて計算機資源に書き込むライブラリ関数を使用点とする．これにより，データの発生源となるライブラリ関数をコールする命令を特定できる．この定義使用連鎖は複数のデータフローから成るため，それぞれを区別するためにデータフロー ID を割り振る．データフロー ID とライブラリ関数コールの命令アドレスを対応付けした表がデータフロー ID 表である．図 1 のプログラムから作成されたデータフロー ID 表を表 1 に示す．図 1 のプログラムには，変数 buf1(定義点 16 行目 fgets，使用点 19 行目 fputs) と変数 buf2(定義点 17 行目 fgets，使用点 20 行目 fputs) の二つの定義使用連鎖が存在し，それぞれにデータフロー ID 1，2 が割り振られる．16 行目と 17 行目の fgets の命令アドレスがそれぞれ 0x080485d9，0x080485f6 とすると，これらの命令アドレスに割り振られるデータフロー ID はそれぞれ 1，2 となる．19 行目と 20 行目の fputs の命令アドレスも同様に，それぞれデータフロー ID 1，2 を割り振られる．

2.3　アクセス制御手法

DF-Salvia のアクセス制御手法を図 2 に示す．*DF-Salvia* は，以下の手順でアクセス制御を行う．

1. プログラム実行直前にデータフロー ID 表を *DF-Salvia* が読み込む．
2. read システムコールが発行された時，read 対象が保護ファイルなら，システムコールを発行したライブラリ関数のデータフロー ID (図 2 では ID 1) に対して，当該保護ファイルのポリシを適用する．
3. write システムコールが発行された時，システムコールを発行したライブラリ関数のデータフロー ID に対して適用されたポリシの有無を確認する．
4. ポリシが適用されている場合は，そのポリシに従って write システムコールの実行の可否を判断する．

```
salvia:~/program# ls
Personal_Information_addr            copy
Personal_Information_addr.slvpolicy  normal1.txt
Personal_Information_tel             normal2.txt
Personal_Information_tel.slvpolicy
salvia:~/program# cat normal1.txt
salvia:~/program# cat normal2.txt
salvia:~/program# ./copy
write normal1.txt: Permission denied
salvia:~/program# cat normal1.txt
salvia:~/program# cat normal2.txt
Mike roft         03-4332-5300
salvia:~/program#
```

図 3　実行結果

5. 制御したシステムコールを発行したライブラリ関数が最終使用点なら，適用されたポリシを削除する．

手順（2）,（3）における，システムコールを発行したライブラリ関数のデータフロー ID は，ライブラリ関数コールの命令アドレスとデータフロー ID 表を比較することで求める．ライブラリ関数コールの命令アドレスは，システムコール発行時にプロセスのスタックを解析することで求める．また，最終使用点とは，データフローの末端の命令文のことである．最終使用点のアクセス制御終了後，当該データフローに適用されたポリシは，以降のアクセス制御に不要となるので削除される．

3　評価

DF-Salvia は，Linux カーネル 2.6.8 を拡張する形で実装している．今回，図 1 のプログラムと表 1 のデータフロー ID 表を用いて，*DF-Salvia* がデータを正確に識別し，データフロー主体のアクセス制御をできるか実験した．実行結果を図 3 に示す．図 3 では，normal1.txt が空で，normal2.txt にデータが書き込まれていることが分かる．すなわち，write deny (コピー禁止) の保護データを読み込んだデータフロー ID 1 に属する 19 行目 fputs は書き込みを制限され，write allow(コピー可) の保護データを読み込んだデータフロー ID 2 に属する 20 行目 fputs は書き込みを実行している．以上より，書き込まれようとしているデータの発生源であるファイルを正確に識別し，それぞれの書き込みを適切なポリシに基づいて制御できていることが分かる．

4　おわりに

データフローを主体としたアクセス制御を実現する *DF-Salvia* の設計について述べ，機能検証実験を行った．今後は，データフロー ID 表自動生成機能の開発を行う．

参考文献

[1] NPO 日本ネットワークセキュリティ協会: “2008 年情報セキュリティインシデントに関する調査報告書 Ver.1.3,” http://www.jnsa.org/result/2008/surv/incident/2008incident_survey_v1.3.pdf，2009.

[2] 鈴来 和久 他: “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.