ることで情報漏洩の防止を実現している．TaintEraser は，事前に Pin を用いて実行ファイルを解析する必要があるため，システム全体のアプリケーションに共通に適用させる場合には向かない．一方，TA-Salvia は，Argos を用いて物理メモリ上のデータを追跡し，アクセス制御を行うため，すべてのアプリケーションに適用可能である．また，TaintEraser は，プロセス間通信を考慮していないため，データが送信されたとき，テイントを伝播できず漏洩してしまう可能性がある．TA-Salvia は，共有メモリやソケットを用いた場合においても追跡が可能である．さらに，本手法により，ネットワークに送信された場合でも継続して追跡が可能である．

TaintEraser のような通信時にデータの追跡が途切れてしまうという問題を解決した TaintExchange[12] という研究がある．TaintExchange は，動的テイント解析フレームワークの libdft[13] で構成されたツールである．libdft は，TaintEraser と同様に Pin を用いて動的テイント解析機能を実現している．TaintExchange は，libdft の基本的な動的テイント解析機能に加え，クロスプロセス・クロスホストでも継続してデータの追跡が行える．TaintExchange は，追跡したデータがソケットやパイプなどを用いて送信されるときに，ヘッダとしてテイント情報をデータに付与して送信する．このヘッダ情報を利用することで，テイント伝播を実現している．TaintExchange は，ビットマップを採用しているため，複数のデータを識別できない．一方，TA-Salvia は，バイトマップを採用しているため，複数のデータを個々に識別することが可能である．また，TaintExchange は，通信相手の特定方法について本文中で述べていない．テイント解析機能を持たないプロセスに対してデータが送信された場合，正しく通信できない可能性がある．TA-Salvia は，コネクション確立前に通信相手が TA-Salvia であることを識別している．

7. おわりに

本論文では，TA-Salvia による保護範囲をネットワークにまで拡張する手法について述べた．この手法を実現するために，データの追跡に必要なタグとアクセス制御に必要なポリシの共有機能を実装した．ポリシの共有は，各 TA-Salvia 上でポリシ同期用のデーモンを稼働させておくことで実現した．また，タグの共有は，Netfilter を用いて TCP パケットの末尾にタグを付与することで実現した．実際に，ファイル共有やメール送信が行われる環境で機能評価を行い，ネットワークに送信されたとしても継続して追跡・制御が行えることを確認した．さらに，FTP を用いた性能評価を行った．何も変更を加えていない環境に比べて，約 3.7〜10.8 倍転送時間が増加してしまうことを確認した．ただし，まだチューニングの余地があるため，転送時間の短縮が可能である．今後は，TA-Salvia のチューニング，TA-Salvia が制御していないシステムコール（sendmsg や pwrite システムコールなど）への対応，暗黙的フローへの対策を検討していく必要がある．

参考文献

[1] NPO 日本ネットワークセキュリティ協会：2016 年情報セキュリティインシデントに関する調査報告書〜個人情報漏えい編〜，`http://jnsa.org/result/incident/index.html` (2017).
[2] Loscocco, P. and Smalley, S.: Integrating Flexible Support for Security Policies into the Linux Operating System, *Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference*, pp. 29–42 (2001).
[3] 原田季栄，保理江高志，田中一男：TOMOYO Linux - タスク構造体の拡張によるセキュリティ強化 Linux，*Proceedings of the Linux Conference 2004*, pp. 1–10 (2004).
[4] 鈴来和久，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(ACS13), pp. 1–15 (2006).
[5] 内匠真也，奥野航平，大月勇人，瀧本栄二，毛利公一：コンパイラと OS の連携によるデータフロー追跡手法，情報処理学会論文誌，Vol. 56, No. 12, pp. 2313–2323 (2015).
[6] 奥野航平，内匠真也，大月勇人，瀧本栄二，毛利公一：コンパイラを用いた情報フロー制御による情報漏洩防止機構，研究報告コンピュータセキュリティ（CSEC），Vol. 2015-CSEC-68, No. 13, pp. 1–8 (2015).
[7] 松本隆志，明田修平，齋藤彰一，毛利公一：動的テイント解析機能を利用した OS による細粒度データ出力制御手法，研究報告コンピュータセキュリティ（CSEC），Vol. 2016-CSEC-75, No. 1, pp. 1–8 (2016).
[8] Portokalidis, G., Slowinska, A. and Bos, H.: Argos: An Emulator for Fingerprinting Zero-day Attacks for Advertised Honeypots with Automatic Signature Generation, *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, EuroSys '06, pp. 15–27 (2006).
[9] Bellard, F.: QEMU, a Fast and Portable Dynamic Translator, *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '05, pp. 41–41 (2005).
[10] Zhu, D. Y., Jung, J., Song, D., Kohno, T. and Wetherall, D.: TaintEraser: Protecting Sensitive Data Leaks Using Application-level Taint Tracking, *SIGOPS Operating Systems Review*, Vol. 45, No. 1, pp. 142–154 (2011).
[11] Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., Wallace, S., Reddi, V. J. and Hazelwood, K.: Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation, *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '05, pp. 190–200 (2005).
[12] Zavou, A., Portokalidis, G. and Keromytis, A. D.: Taintexchange: A Generic System for Cross-process and Cross-host Taint Tracking, *Proceedings of the 6th International Conference on Advances in Information and Computer Security*, IWSEC'11, pp. 113–128 (2011).
[13] Kemerlis, V. P., Portokalidis, G., Jee, K. and Keromytis, A. D.: Libdft: Practical Dynamic Data Flow Tracking for Commodity Systems, *SIGPLAN Not.*, Vol. 47, No. 7, pp. 121–132 (2012).