SVMを用いたWindows向け異常検知システムの実装

伊波靖†　高良富夫‡
†沖縄工業高等専門学校メディア情報工学科　‡琉球大学工学部情報工学科

1 はじめに

インターネットの急速な進展とともに，不正なプログラムの感染および拡大方法の多様化と伝搬速度の高速化により，シグネチャによる不正プログラム対策ソフトウェアの限界が議論され，プログラムの振舞いに基づくビヘイビア型異常検知システムがさかんに研究されている．UNIX系のOSにおいては，システムコール列の学習による侵入検知の研究が早くから行われ，多くの手法が提案されてきた[1][2]．一方，Windows系のOSにおいて危険なシステムコールによるOSへの攻撃を予め登録されたアクセス制御データベースに基づき検知し，実行を阻止するWHIPSと呼ばれるシステムの提案が行われている[3]．しかし，WHIPSは危険なプロセスとそのプロセスによって発行されたシステムコール及び引数をルールとして予めアクセス制御データベースに登録する必要があり，未知の不正なプログラムへの対応が困難である．また，ビヘイビア型異常検知システムでは，高い検知率を得ることにともなうfalse positiveの割合を減少させることが課題となっている．我々はこの問題を解決するために，Windowsにおける危険なシステムコールに着目した侵入検知を行う手法を提案し，ルールベースで検知することで高い検知率を実現し，SVMを用いることでfalse positive率を減少できることを示した[4]．そこで，本研究では，我々が提案したSVMを用いた異常検知手法をWHIPSに組み込むことで，ルールベースとSVMの組合せによる高い検知率とfalse positiveの減少を実現したWindows侵入検知システムを実装する．

2 Support Vector Machine(SVM)

SVMは統計的学習理論に基づく新しい2クラスのパターン認識手法であり，ニューラルネットワークなどの従来法と比較して汎化能力が高い点と最適解が求まる点に特徴があり，学習に用いていないデータに対しても高い認識率を示す．SVMがこのような特徴を示すのは，その学習に認識誤りと汎化性能の両面から最適化が行われ，これが2次の凸計画問題として定式化されているため最適解を求める事ができるためである[5]．

SVMは学習の最適解として求められた分離超平面による線形識別を行っているが，学習用データを線形分離することが不適切な場合，学習データを元のパターン空間からより高次のパターン空間に非線形写像を行い高次元空間で分離超平面を構築し線形識別を行う．

3 危険なシステムコールに着目したWindows向け異常検知手法

危険なシステムコールに着目したWindows向け異常検知手法は，ルールベースによる検知とSVMによる識別を組み合わせており，以下の構成となっている．提案方式の流れを図1に示す．

3.1 クリティカルシステムコールデータベースの構築

過去の不正なプログラムの分析から，クリティカルなシステムコールとなりうるシステムコールと引数のリストを予め作成し，クリティカルシステムコールデータベース(CSCDB)を構築してしておく．

3.2 システムコールの監視

システムコールを監視し，システムコールの時系列を収集する．また，SVMの識別モデルを学習により生成するために，予めステムコールの監視を行い，正常なプログラムと不正なプログラムのシステムコール時系列を収集しておく．

3.3 クリティカルなシステムコールの検知

システムコールの監視を行い保護すべき資源に対して影響を与えるシステムコールが発行された際に，CSCDBと照合しクリティカルなシステムコールかどうかを判断する．ここで，クリティカルなシステムコールとは，システムの可用性に対して影響を与える可能性がある危険な引数を伴って発行されるシステムコールである．

3.4 SVMによる危険なシステムコールの識別

クリティカルなシステムコールと判断した場合は，そのシステムコールより過去に発行されたシステムコールの時系列から素性データを生成し，SVMを用いて危険なシステムコールかどうかを識別する．ここで，危険なシステムコールとは，悪意を持ったプログラムによって発行されるクリティカルなシステムコールである．SVMで用いる素性データは，システムコール時系列データおけるシステムコールを要素番号，システムコールの頻度を要素の値とするペアを用いた．なお，SVMは予め正常なプログラム及び不正なプログラムの学習データによって学習を行っている．

4 WHIPSとは

4.1 WHIPSの動作原理

WHIPS(Windows Host Intrusion Prevention System)は，Windows系OSにおいてカーネルモードで動作するホスト型侵入防止システムで，GNU General Public License V2に基づいて公開されている．WHIPSは，アプリケー

An Implementation of Detecting Anomalies System for Windows using SVM.
†Yasushi IHA (yasuc@okinawa-ct.ac.jp)
‡Tomio TAKARA (takara@ie.u-ryukyu.ac.jp)
†Department of Media Information Engineering, Okinawa National College of Technology
‡Department of Information Engineering, University of The Ryukyus

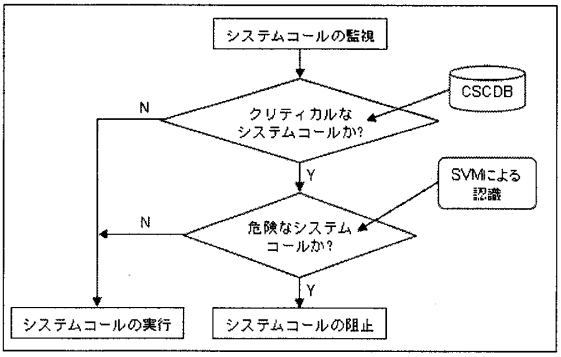

図 1: 提案方式の流れ

ション，サービス，ドライバから構成されており，カーネルモードで動作するドライバにおいて，ルールに基づいたシステムコールの実行制限を行っている．WHIPS は SSDT (System Service Descriptor Table) Patching と呼ばれる手法を用いて，Windows 本来のシステムコール処理ルーチンを，WHIPS 内部の処理ルーチンへとフックしている．WHIPS 内部の処理ルーチンでは，Reference Monitor と呼ばれる関数で，各システムコールの呼び出しについて ACD (Access Control Database) へ登録されているルールとのマッチングを行い，もし，実行が禁止された危険なシステムコールであれば，システムコールを発行したプロセスに STATUS_UNSUCCESSFUL を返すことで，危険なシステムコールの処理を阻止する．

4.2 WHIPS の課題

WHIPS において ACD に登録されたルールは，システムコールを制限したいプロセス名と，そのプロセスが発行したシステムコール及びシステムコールの引数を記述する必要がある．不正なプログラムのプロセス名が予め判明している場合は，高い検知率を示すが，プロセス名の分からない未知の不正プログラムについては，ルールを記述することが困難となっている．また，WHIPS の ACD へのルールの設定は，一般の利用者には難しく，また，未知の不正プログラムへの対応も不十分である．

5 WHIPS への実装方法

WHIPS の課題を解決する目的で，ACD におけるルールの拡張と，SVM を用いた異常検知手法を以下のように実装した．

5.1 SYSENTER HOOK によるシステムコール時系列の収集

WHIPS がシステムコールをフックする為に使用している SSDT Patching では，全てのシステムコールをフックするためには，全てのシステムコールについて処理を行う関数を用意する必要があり，現実的ではない．そこで，プロセスが呼び出したシステムコールの時系列を収集するために SYSENTER HOOK と呼ばれる方法を用いた．SYSENTER 命令は，ユーザモードからカーネルモードへ遷移する際に呼ばれる命令で，SYSENTER でカーネルモードに遷移した後で，SSDT による各処理関数への分岐が行われる．この命令の動作を変更することで，カーネルモードに遷移した際に，予め用意していた関数へ実行制御を移すことができ，各システムコールの処理関数へ分岐する前に必要な処理を行うことが可能となる．SYSENTER を処理する関数では，プロセス ID をキーとしたシステムコール時系列のキューにシステムコールを格納することで，システムコール時系列を収集している．なお，システムコール時系列のキューの長さは 200 としている．

5.2 システムコールと引数によるルールマッチング

プロセス名を指定することなく，クリティカルなシステムコールになりうる，システムコールと引数の組合せを予め登録した．また，WHIPS の ACD によるルールマッチングを拡張し，複数のシステムコールの組合せによるルール記述が可能なように変更した．例えば，NtCreateFile と呼ばれるファイルの生成やオープンを行うシステムコールと実際にオープンしたファイルにデータを書き出すための NtWriteFile を組み合わせることで，クリティカルなシステムコールの検出精度の向上を図っている．

5.3 SVM による危険なシステムコールの識別

ACD によるルールマッチングによって，クリティカルなシステムコールと判断された場合は，キューに格納されているシステムコール時系列より，N-Gram 法を用いて SVM の素性データを生成する．N-Gram の N は 4 とした．生成した素性データを用いて，予め用意した SVM の学習モデルにより危険なシステムコールかどうかを識別する．

6 まとめと今後の課題

Windows における危険なシステムコールに着目した侵入検知手法を WHIPS に組み込むことで，ルールベースによる検知と SVM による識別を組み合わせて異常検知を行うシステムを実装した．現在，システムの実装が終了し，動作の確認を行っている．今後，スループットの測定と検知率及び false positive の割合について実験を行い，システムの有効性を検証する予定である．

参考文献

[1] Forrest, S., Hofmeyr, S. A., Somayaji, A. and Longstaff, T. A.: A sense of self for Unix processes, *IEEE Symposium on Security and Privacy*, pp. 120–128 (1996).

[2] Sekar, R., Bendre, M., Dhurjati, D. and Bollineni, P.: A Fast Automaton-Based Method for Detecting Anomalous Program Behaviors, *IEEE Symposium on Security and Privacy*, pp. 144–155 (2001).

[3] Battistoni, R., Gabrielli, E. and Mancini, L. V.: A Host Intrusion Prevention System for Windows Operating Systems, *Proceedings of the 9th European Symposium On Research in Computer Security (ESORICS 2004)*, pp. 352–368 (2004).

[4] 伊波靖，高良富夫：危険なシステムコールに着目した Windows 向け異常検知手法，情報処理学会論文誌，Vol. 50 No. 9, pp. 2173–2181 (2009)

[5] Cristianini, N. and Shawe-Taylor, J.: サポートベクターマシン入門，共立出版 (2005).