科学研究費助成事業　　研究成果報告書

平成　２６　年　　６　月　　６　日現在

機関番号：　５８００１
研究種目：　基盤研究(C)
研究期間：　2011～2013
課題番号：　２３５００１０６

研究課題名（和文）ＳＶＭを用いた危険なシステムコールに着目した異常検知システムの開発

研究課題名（英文）Development of abnormality detection system focused on a dangerous system calls using the SVM.

研究代表者
伊波　靖（IHA, Yasushi）

沖縄工業高等専門学校・メディア情報工学科・教授

研究者番号：６０３９０５６４

交付決定額（研究期間全体）：（直接経費）　　3,600,000円、（間接経費）　　1,080,000円

研究成果の概要（和文）：本研究では、Windows系のプログラム検知方法について危険なシステムコールに着目しルールベースによる検知とSVMによる識別を組み合わせることで検知が可能となるシステムの開発を行い検知の可能性を確認した。また、SVMを用いてプログラムの特徴に着目した不正プログラム検知手法をWHIPSへ実装し、99.4%の高い検知率で検知が行えることを確認した。さらに、SVMによる不正プログラム検知手法を応用しWebアプリケーションに対する攻撃を検知するためのWAFをWebサーバのモジュールとして実装し有効性について検討した。

研究成果の概要（英文）：In this study, we propose the anomaly detection method of combining behavior of program and detection rule to detect a dangerous system call that affects important resource of Windows system. The proposed method first detects a doubtful system call by the detection rule using system call and argument. Then, a dangerous system call is identified by using Support Vector Machine from the history of the system call, and execution is intercepted. We performed an experiment by developing the prototype system based on the proposed method, and using realistic malicious program and usual program. Through the experiments, we have evaluated the detection rate of the proposed technique and the ratio of false positive.

研究分野：　総合領域

科研費の分科・細目：　情報学・計算機システム・ネットワーク

キーワード：　不正プログラム検知　ネットワークセキュリティ技術　SVM　WAF

１．研究開始当初の背景

(1) Windows 系 OS における不正なプログラムによる被害の増加

研究開発時点において，インターネットの急速な進展にともない，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が情報システムに対する大きな脅威となっていた．特にクライアント環境として普及している Windows 系の OS をターゲットとしたウイルス，ワーム，ボット，トロイの木馬などの多種多様な不正なプログラムが流通していた. Windows 系の OS では，システムに不慣れな一般ユーザであっても管理者権限が付与されているケースが多く，不正なプログラムを誤って実行したり，未知の脆弱性を用いた攻撃が行われたりした場合，システムに大きな被害を与えることになる．

(2) Web アプリケーションへの攻撃の増加

インターネットの急速な発展に伴い WWW(World Wide Web) が普及したことにより，企業や官公庁等において多くの Web アプリケーションが利用されるようになった．しかし，利用が増える一方で，Web アプリケーションを狙った XSS(Cross-Site Scripting) 攻撃や SQL インジェクション攻撃も後を絶たず，Web サイトの改ざんや非公開情報の漏えいなどの深刻な被害が報告されている．Web アプリケーションを XSS 攻撃や SQL インジェクション攻撃から防御する方法の一つとして，WAF(Web Application Firewall) の使用が推奨されている．

(3) シグネチャ方式の限界

不正プログラム対策ソフトウェアの多くは，予め解析により判明した不正なプログラムを特徴付けるシグネチャと呼ばれるパターンを用いて不正なプログラムを検出している．そのため，頻繁に出現する新種の攻撃コードや次々と登場する多岐にわたる亜種へのリアルタイムの対応が限界となりつつある．また，WAF は，入力値をホワイトリストおよびブラックリストとのシグネチャマッチングで識別し攻撃を遮断する．しかし，正常なリクエストを不正なリクエストとして誤検知する False Positive の問題や未知の攻撃に対して検知漏れを起こす可能性がある．

(4) ビヘイビア型方式への期待と課題

この問題を解決するための方法として，不正なプログラムの検出をプログラムの振舞に基づいて行うビヘイビア型の異常検知システムが盛んに研究されている．Windows 系の OS においては, システムサービスの呼び出し列を N-gram 法と呼ばれる方法により特徴化し，予め通常のプログラムを実行して収集したデータに基づいて異常検知を行う手法が提案されている．また，システムコールをルールベースにより監視する異常検知システムとして，WHIPS が提案されている．一般的に，ビヘイビア型の異常検知システムにおいて高い検知率を得るためには，通常のプログラムを誤って不正なプログラムと判定してしまう False Positive が増加する問題を解決する必要があった.

２．研究の目的

(1) ビヘイビア型の異常検知システムの問題解決

一般的に，ビヘイビア型の異常検知システムにおいて高い検知率を得るためには，通常のプログラムを誤って不正なプログラムと判定してしまう False Positive が増加する問題を解決する必要があった．たとえば，ソフトウェアをシステムに導入するために使用されるインストーラーなどの通常のプログラムは，システムフォルダへのファイルの作成などの不正なプログラムと類似した動作を行うため，ビヘイビア型の異常検知システムでは不正なプロラムと判定されてしまう可能性が高い．そのため，セキュリティ対策ソフトの評価指標の 1 つとして，False Positive が小さいほど検知精度が高いということになる. そこで, 本研究では, Windows 系の OS において, ルールベースの検知とプログラムの振舞いを組み合わせた新たな異常検知方式を提案する．提案方式では，まず，クリティカルなシステムコールと呼ぶ OS の重要な資源に影響を及ぼす可能性のあるシステムコールを不正なプログラムの挙動に基づいてあらかじめ定義したルールで検知する．次に，検知したクリティカルなシステムコールについて，システムコールの発行履歴から N-gram 法を用いて生成した特徴ベクトルを素性とする Support Vector Machine（SVM）により不正なプログラムにより発行された危険なシステムコールかどうか判定する．

(2) WHIPS への実装

Windows 系の OS においては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 (ACD: Access Control Database) に基づき検知し，実行を阻止する WHIPS(Windows Host Intrusion Prevention System) と呼ばれるシステムの提案が行われている．しかし，このシステムは ACD に予め危険なプロセスとシステムコール及び引数の組をルールとして登録する必要があり，未知の危険なプログラムへの対応が困難であった．そこで，我々が提案した Windows 系の OS において OS の資源に危険を及ぼすクリティカルなシステムコールを，システムコールと引数の組み合わせによるルールのみで検知し，検知したクリティカルなシステムコールが不正なプログラムによって発行された危険なシステムコールかどうかを SVM(Support Vector Machine) を用いて識別し阻止する異常検知手法を WHIPS へ実装することで，高い検知率と False Positive の割合を減少

させた異常検知システムを開発する．
(3) Web アプリケーション防御への応用
開発した SVM を用いた異常検知システムを WAF へ応用する．WAF においても，N-gram 法と SVM を用いた検知手法の有効性は示されてきたが，Web サーバとしての利用実績が高い Apache Web サーバのモジュールとして実装された例はこれまで存在しなかった．Apache Web サーバの WAF モジュールとしては，ModSecurity が広く利用されているが，ModSecurity は大量のシグネチャパターンと正規表現による検知手法により Web サーバのパフォーマンスに影響を与えることが知られている．また，False Positive の発生を抑えるためにルールセットを適切に設定することが難しく，運用上の問題となっている．そこで，ModSecurity の Web サーバのパフォーマンスに与える影響を実際の攻撃パターンを含む大量のデータセットを用いた評価実験を通して確認するとともに，ModSecurity の代替として Apache のモジュールとして使用できる XSS および SQL インジェクション攻撃用の WAF の実装を行う．

３．研究の方法
(1) SVM の特徴データの収集とモデルの構築
システムで用いる SVM の特徴データを収集するために，システムコール時系列を収集する．収集した時系列データを基に，N-gram 法により特徴データを生成する．生成した特徴データを SVM の学習データとして SVM のモデルを構築する．システムコール時系列は，次に示す危険なシステムコールの定義に基づき，クリティカルなシステムコールを検知した場合に，N-gram 法により特徴データを生成する．
(a) 危険な引数とは，システムの可用性に対して影響を与える可能性がある引数である．
(b) クリティカルなシステムコールとは，危険な引数を伴って発行されるシステムコールである．
(c) 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．
(2) 異常検知手法の Windows 系 OS への実装を行う
ルールベースと SVM による識別を組み合わせた異常検知手法を，Windows 系 OS へデバイスドライバとして実装を行う．その際，オープンソースで開発が行われている，WHIPS と呼ばれるルールベースの Windows 用異常検知システムをベースとして，危険なシステムコールに着目したルールの拡張と，SVM による識別を組み合わせたモジュールを実装する．
(3) 実装したシステムの性能評価
実装したシステムについて，検知率及び False Positive の割合とスループット等の実行性能について評価試験を行い，システムの有効性とボトルネックについて調査を行う．評価試験は，実験環境においてベンチマークテストを稼働させ，システムを適用した場合と，適用しなかった場合のスループットについて計測する．計測した結果により，システムのオーバーヘッドが大きい場合は，プロファイリングを行い，システムのボトルネックを調査し，チューニングを行う．また，ワームやウイルスなどの実際の不正なプログラムと，不正なプログラムとして誤検知されやすいインストーラプログラムなどの正常なプログラムを実際に実行し，システムによる検知率と False Positive 率について評価する．
(4) 実装したシステムの総合評価
実装したシステムについて，総合評価及び実環境における性能について検証する．検証結果に基づきシステムのチューニングを行っていく．また，ユーザに通常業務を行ってもらい，体感的な速度や動作について調査を行う．
(5) 実装したモジュールを用いた異常検知システムの新規開発
WHIPS の独立したモジュールとして実装した部分に基づいて，異常検知システムの新規開発を行う．開発したシステムについて評価試験を行い，実用的なシステムの完成を目指す．開発するシステムでは，危険なシステムコールを検知するために，デバイスドライバ内でシステムコールをフックし，システムコール時系列の履歴を収集すると共に，ルールに基づいて危険なシステムコールを検知し，SVM により識別するシステムを構築する．
(6) Web アプリケーション防御への応用
SVM を用いた異常検知システムを WAF へ応用する．
４．研究成果
(1) 危険なシステムコールに着目した Windows 向け異常検知手法
ここでは，我々が提案した異常検知手法の概要について説明する．
・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) を構築しておく．
・システムコールの監視
システムコールの監視手法として SSDT(System Service Descriptor Table) の書き換えを使用する．SSDT をフックすることで，ユーザランドのプログラムからカーネルランドの OS の機能を呼び出す際に引数の取得や呼び出したプロセスの情報などを所得することが出来る．
・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．
・SVM による危険なシステムコールの認識
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についてSVM を用いて危険なシステムコールかどうかを判断する．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．

(2) SVM を用いた Windows 向け異常検知システムの実装

ここでは，WHIPS へ実装した SVM を用いた Windows 向け異常検知機能について説明する．
・ルールの拡張

WHIPS のルールを拡張しルールの汎用化を行った．プロセス名をワイルドカードで表現することで，クリティカルなシステムコールのみで検知できるようにした．また，複数システムコールの組合せによりクリティカルなシステムコールを検知できるようにした．これは，単一のシステムコールをルールでチェックするより，複数のシステムコールをブロックとして扱ってチェックした方が識別率が高くなるためである．
・SYSENTER フック

Windows 系 OS においてシステムコールの監視には SSDT を書き換えフックすることで，監視するシステムコールを個別に指定可能にする方法とユーザランドからカーネルランドへ遷移する際に使用される命令 SYSENTER をフックすることにより全ての System Call を同一のフック関数で監視可能にする方法がある．WHIPS では SSDT フックのみを使用していたが，システムコールの時系列を作るために，全てのシステムコールをプロセス ID 毎に作成したキューに格納する必要があり，SSDT によるフックだけでは，全てのシステムコールについて処理関数を用意する必要があるため SYSENTER でフックすることで，システムコール時系列の採取する処理関数を一つにまとめることが出来る．
・SVM による危険なシステムコール識別

SVM で識別を行うためには特徴ベクトルを定義する必要がある．本研究では，ルールによってクリティカルなシステムコールを検知した際に，SYSENTER フックにより採取したクリティカルなシステムコールにいたるシステムコール履歴の時系列から各システムコールの N-gram における頻度を特徴ベクトルとしている．生成した特徴ベクトルを用いて， SVM により危険なシステムコールかどうか識別する．なお，SVM には SVMlight を用いた．

(3) システムの評価と考察
・学習データの収集方法

本提案方式の有効性を確認するために認識実験を行った．学習用の正常なプログラムは，ソフトウェア配布サイトから無作為にダウンロードした 140 種類のインストーラプログラムとした．また，不正なプログラムはウイルス及びワームを中心として 140 種類とした．正常及び不正なプログラムについて仮想環境にインストールされた Windows XP 上で実行し，システムコールの監視によりシステムコール列をログとして取得し，N-gram 法により特徴ベクトルを生成した学習データを用いて SVM を学習し，SVM のモデルを構築した．なお，予備実験の結果から時系列の長さ L = 200，N-gram の N = 4 とした．
・クリティカルなシステムコール検知結果

取得したシステムコール列を用いて提案方式によるクリティカルなシステムコールの検知率を調べるために行った検知実験の結果を表 1 に示す．CSCDB に基づく検知により，不正なプログラムによるクリティカルなシステムコールを 100%検知することが可能となっている．なお，正常なプログラムとしてインストーラプログラムによるログを用いているため，正常なプログラムにおいても検知率が高くなっている．このことより不正なプログラムの挙動とインストーラプログラムの挙動は様々な面で類似性が高く，ルールベースのみで識別するのは困難であることが分かる．

表 1: クリティカルなシステムコール検知結果

不正なプログラム		正常なプログラム	
学習用	認識用	学習用	認識用
100.0%	100.0%	95.0%	100.0%

・危険なシステムコール認識結果

クリティカルなシステムコールとして検知されたシステムコールについて，システムコール時系列を素性として SVM により認識した結果を表 2 に示す．なお，表中の認識率は，検知したクリティカルなシステムコールを正常なプログラムによるものと不正なプログラムによるものとに正しく認識できた結果である．FP(False Positive) は，正常なプログラムによるものを不正なプログラムとした結果で，FN(False Negative)は，不正なプログラムによるものを正常なプログラムとした結果である．認識結果から一部のシステムコールを除き提案方式による危険なシステムコールの認識率が高く，FN が低いことが分かり，提案方式の有効性が高いことが言える．

表 2: 危険なシステムコール検知結果

NtWriteFile			NtSetValueKey		
認識率	FP	FN	認識率	FP	FN
96.3%	11.2%	1.4%	97.0%	66.7%	0.3%
NtMapViewOfSection			NtReadFile		
認識率	FP	FN	認識率	FP	FN
97.0%	50.0%	1.5%	83.2%	9.0%	55.2%

・検知実験の結果

通常のプログラム（インストーラー）と実際のマルウェアをそれぞれ 5 種類リアルタイムで実行させ検知可能かどうか実験を行った．検知実験の結果を表 3 に示す．実験結果からインストーラーはクリティカルなシステムコールを発行しているため，ルールベースでは不正となるが SVM により通常のプログラムとして認識を行っていることが分かる．また，マルウェアについてはルールベースおよび SVM の両方で不正なプログラムとして検知が行われている．この結果から，リアルタイムでの検知が可能となっていることが言える．

表 3: 検知実験の結果

	プログラム名	ルールベース	SVM 識別
通常	A-Downloader320	不正	正常
	CravingExplorer-0-18-8	不正	正常
	GOMPLAYERJPSETUP	不正	正常
	Lhaca124	不正	正常
	XeloPDF201	不正	正常
マルウェア	Antinny	不正	不正
	Bagle	不正	不正
	Gibe	不正	不正
	NetSky-b	不正	不正
	Sasser	不正	不正

(4) SVM を用いたプログラムの特徴に基づく異常検知システムの実装

プログラムがインポートしている DLL と API を用いてプログラムの特徴ベクトルを生成し，SVM により判別する手法の WHIPS への実装を行っている．予備実験により評価データを用いた Open Test において認識率 98.35%が得られ，未知のデータについても検知が行えることを確認した．また，WHIPS においてプロセス生成時に DLL と API のリストを取得し特徴ベクトルを生成することが可能となった．

(5) SVM を用いたシステムコール履歴に基づく異常検知システムの BitVisor への実装

PC のセキュリティ向上を目的としたハイパーバイザ（仮想マシンモニタ）として開発されている BitVisor に Windows のシステムコール履歴を取得する機能を実装し，取得したシステムコール履歴から N-gram 法により特徴ベクトルを生成した．また，SVM を BitVisor に実装するために数値演算ライブラリを組み込み数値計算関数の実行と浮動小数点演算を可能にした．

(6) SVM を用いた WAF への異常検知機能の実装

Web アプリケーションへの攻撃を防ぐ WAF への実装を目的として Web サイトへのリクエスト中のクエリ文字列から文字に着目して N-gram 言語モデルにより特徴ベクトルを生成し，生成した特徴ベクトルを SVM で認識させる異常検知手法を Apache のモジュールとして実装した．実装したモジュールの性能を評価するために Apache の標準的な WAF である ModSecurity との性能比較実験を行った．性能比較実験の結果から，Accuracy = 99.98%，Precision = 100.0%，Recall = 99.92%，F 値= 0.9996 が得られ，False Positive を低減させながらも ModSecurity を上回る認識性能と処理性能で検知を行えることを明らかにし，実装したモジュールが ModSecurity の代替モジュールとして有効であることを示した．

５．主な発表論文等

（研究代表者，研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 1 件）

(1) 伊波靖,高良富夫：サポートベクタマシンを用いた WAF への異常検知機能の実装と評価，情報処理学会論文誌 コンピューティングシステム，Vol.7，No.1，pp．1-13，2014.

〔学会発表〕（計 5 件）

(1) 新垣杏里，伊波靖：SVM を用いたプログラムの特徴に基づく異常検知システムの改良，情報処理学会第 76 回全国大会講演論文集（3）pp．621-622，2014.
(2) 伊波靖，新垣杏里：SVM を用いたプログラムの特徴に基づく異常検知システムの実装，情報処理学会第 75 回全国大会講演論文集（3）pp．513-514，2013.
(3) 伊波靖，HENDRA GUNTUR：SVM を用いたシステムコール履歴に基づく異常検知システムの BitVisor への実装，情報処理学会第 75 回全国大会講演論文集（3）pp．515-516，2013.
(4) 伊波靖，高良富夫：SVM を用いた Windows 向け異常検知システムの実装と評価，FIT2012（第 11 回情報科学技術フォーラム）講演論文集 第 4 分冊 pp．203-206，2012.
(5) 伊波靖，高良富夫：SVM を利用した WAF への異常検知機能の実装と評価，情報処理学会第 74 回全国大会講演論文集（3）pp．561-562，2012.

６．研究組織

(1)研究代表者

伊波　靖（IHA Yasushi）
沖縄工業高等専門学校・メディア情報工学科・教授
研究者番号：60390564