Computer Security Symposium 2018
22 - 25 October 2018

API 呼び出しとそれに伴う経過時間とシステム負荷を用いた重み付けスコアに基づくマルウェア検知手法

佐藤 順子[†1]　花田 真樹[†2]　面 和成[†3]　村上 洋一[†2]
鈴木 英男[†2]　布広 永示[†2]　折田 彰[†4]　関口 竜也[†4]

概要：近年，マルウェアの自身の隠蔽が巧妙化しており，未知のマルウェアを高精度で検知する手法が求められている．マルウェアは実行環境の検知，セキュリティサービスの無効化，自分自身の削除などの機能 を用いて検知を免れようとしている．そこで筆者らはこれまで，API 呼び出しのパターンや経過時間，また システム負荷の変動に関する特徴情報を用いた単純ベイズ分類器による検知手法を提案した.本研究では，当該手法のさらなる検知精度向上のために，特徴情報ごとに分類器を作成し，それから算出されたスコアに重み付けをして足し合わせた値によってマルウェアを検知する手法の提案し，評価を行う．

キーワード：マルウェア検知，API，システム負荷，機械学習

Malware Detection Method using a Weighted Sum Model based on API Call Patterns, Elapsed Time and System Load between API Calls

Junko Sato[†1]　Masaki Hanada[†2]　Kazumasa Omote[†3]　Yoichi Murakami[†2]
Hideno Suzuki[†2]　Eiji Nunohiro[†2]　Akira Orita[†4]　Tatsuya Sekiguchi[†4]

Abstract: Malware detection method with high accuracy is strongly required, because mechanism of mal- ware is getting sophisticated to evade detections by antivirus software. The cunning malware tries to evade detection using functions which detect the system environment, disable the security protection and remove myself. We have so far proposed a malware detection method based on API call pattern, elapsed time and system load between API calls. In this study, in order to make further improvement of the detection accuracy, we propose and evaluate a method using a weighted sum of scores from different classifiers constructed based on those features.

Keywords: Malware detection method, API, System load, Machine Learning

1. はじめに

近年，マルウェアの自身の隠蔽が巧妙化しており，Nemesis[1]というマルウェアは，ボリューム・ブート・レコードの改変を行い，発見だけでなく削除も困難にしている．マルウェアは実行環境の検知[2]，セキュリティサービスの無効化，自分自身の削除などの機能を用いて検知を免れようとしている．そのため，未知のマルウェアを高精度で検知する手法が求められている．そこで筆者らはこれまで，アプリケーションプログラミングインターフェース（Application Programming Interface; API）呼び出しのパターンや経過時間，またシステム負荷の変動に関する特徴情報を用いた単純ベイズ分類器による検知手法を提案した．本研究では，さらなるマルウェア検知精度向上のために，特徴情報ごとに分類器を作成し，それから算出されたスコアに重み付けをして足し合わせた値によってマルウェアを検知する手法の提案し，評価を行う．また，単純ベイズ分類器（Naïve Bayes Classifier; 以下，NBC と称する）だけでなく，多項式カーネル（Polynomial kernel）を用いたサポートベクターマシン（Support Vector Machine; 以下，SVM_{poly} と称する），ラジアル基底関数カーネル（Radial Basis Function kernel; ガウシアンカーネルとも呼ばれる）を用いたサポートベクターマシン（以下，SVM_{RBF} と称する），ランダムフォレスト（Random Forest; 以下，RF と称する）についても同様に重み付けを用いた手法により評価を行う．本稿では，まず第 2 章では関連研究について述べ，第 3 章では提案手法について述べる．また，第 4 章で評価実験を行う際に用いた使用検体，実験環境，評価指標，実験結果を述べ，第 5 章ではまとめと今後の課題について述べる．

†1 東京情報大学 総合情報学研究科
Graduate School of Informatics, Tokyo University of Information Sciences
†2 東京情報大学 総合情報学部
Department of Information Sciences, Tokyo University of Information Sciences
†3 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†4 株式会社日立システムズサイバーセキュリティリサーチセンタ
Hitachi Systems, Ltd. Cyber Security Research Center

2. 関連研究

マルウェアの検知手法や分類手法に関して，API を特徴として用いた研究が広く行われている．

API 呼び出しパターンに着目した検知手法[3]では，API 呼び出しパターンの特徴を用いて，マルウェアの種類判定を行っている．マルウェアは，その種類によって固有の API 呼び出しパターンを持っていることが知られており，検知に有効な API 呼び出しパターンの連鎖数が異なっていることが報告されている．

API を用いた機能に基づく検知手法[4]では，API がマルウェアの機能を決定しているという点に着目し，API の特徴を用いてマルウェアの有している機能ごとの分類を行っている．この手法では，複数の機械学習を用いた提案・評価を行っており，マルウェアの機能を高精度で分類することが可能である．

実行毎に動作を変える挙動に着目した検知手法[5]では，パターンマッチング法や C&C サーバーの特定を困難にするマルウェアに着目している．この手法では，特定の API の引数に注目しており，実行毎に引数が異なれば，マルウェアとして検知することが可能である．

筆者らが行った先行研究 1 つとして，マルウェアの API 呼び出しパターン，API 呼び出しによる経過時間とシステム負荷に注目して解析を行った研究[7]がある．この研究では，マルウェアは，API 呼び出しにより，システム負荷が高くならないようにするなどのユーザから身を隠すための特徴を有しているという仮定に基づき解析を行った．その結果として，マルウェアと正規プログラムでは，同じ API 呼び出しパターンでも，その API 呼び出しによる経過時間とシステム負荷に特徴が表れることを明らかにした．また，筆者らは，[7]の結果を踏まえた検知手法を提案した[8]．

その他にも，システムの振る舞いに着目した研究として，システムフォルダなどの別のフォルダに新たなファイルを生成する挙動に着目した検知手法[6]が提案されている．ユーザからの発見を免れるために行うボットの身を隠すため動作と侵入後消去されないようにする挙動に着目しており，侵入後生成されたファイルを侵入前の環境で実行したときに，侵入挙動が観測された場合にボットであると検知する．また，通信の時間的な振る舞いに着目した研究として，ボットの通信の挙動から検知する手法が提案されている．

本研究では，先行研究[8]で明らかにされた知見に基づき，より高精度な検知手法を提案する．

3. 提案手法

先行研究[8]と同様に，検体を動作させた時に得られた API やそれに伴う経過時間とメモリ使用量を用いた検知手法を提案する．以下，本提案手法の流れについて説明する．また，図 1 は提案手法の検知のフロー図である．

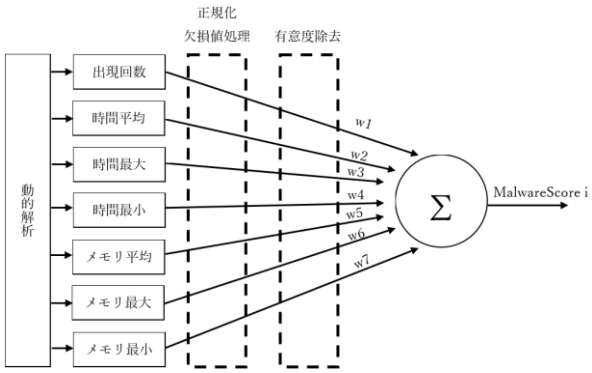

図 1　提案手法のフロー

Figure 1　Flow of the proposed method

(1) API 呼び出しによる経過時間やシステム負荷の特徴を抽出する手法は様々なものが考えらえるが，本研究では，検体を CuckooSandbox[9]を用いて動的解析し，API 呼び出しと，それに伴う呼び出しによる経過時間，その時のメモリ使用量の変動をログに記録する．

(2) API 呼び出しによる経過時間とシステム負荷を，統計的な手法(最大、最小、平均)を用いて特徴づける．本手法では，7 つの特徴を用いて機械学習を行う；(i) 各検体における API 呼び出しパターン(以降，遷移と呼ぶ)の出現回数，(ii) API 呼び出しによる経過時間の平均値，(iii) API 呼び出しによる経過時間の最大値，(iv) API 呼び出しによる経過時間の最小値，(v) メモリ使用量の平均値，(vi) メモリ使用量の最大値，(vii) メモリ使用量の最小値．例えば，遷移数が 2 の API （ある API1 呼び出し後に，次の API2 を呼び出したパターンの遷移）に着目した場合，API1 から API2 の遷移の出現回数が 2 で，API1 から API2 までの経過時間の平均値，最大値，最小値，またメモリ使用量の平均値，最大値，最小値の 7 つの特徴がある．特徴ベクトルは，API 呼び出しパターンが複数存在する場合は，この表記を連続で結合して表記する（図 2）．本研究では，10,712 個の API を用いており，検体中で出現した組み合わせは 5,451 個（API1→API2 のような表記が 5451 個）ある．すなわち，特徴ベクトルのサイズは，5,451 次元である．

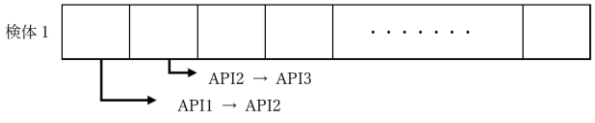

図 2　特徴ベクトルの構成

Figure 2　Construction of a feature vector.

(3) 各特徴値は，−1 から+1 に正規化する．欠損値については，正規化した-1 から+1 までの範囲以外の値であるとみなし，本研究では，−2 とした．さらに全ての検体において，ある特定の特徴の値が全て欠損している場合、もしくは同じ値の場合がある．そのような特徴は機械学習に寄与しないと考えられる．そこで，それらの特定の特徴ごとに t 検定を行い，p 値が 0.05 未満の項目以外は除去を行う．例えば，2 遷移のメモリ使用量の場合、特徴ベクトルの次元数が，5,451 から，1,283 に特徴が除去される。

(4) (2)で取得した特徴ごとに機械学習法を用いて分類器を構築する．本研究では，次の 4 つの機械学習法を用いて分類器を構築する; NBC，SVM_{poly}，SVM_{RBF}，RF. すなわち，各機械学習法に対して，特徴 7 つの分類器を構築する．また分類器には，最も予測精度が高かった機械学習モデルを選択して用いる．

(5) (4)で構築した 7 つの分類器を用いて計算されたスコア（s_j）に重み付けをして足し合わせた値（$MalwareScore_i$）によってマルウェアを検知する（Weighted Sum Model; 以下，WSM と称する）．

$$MalwareScore_i = \sum_{j=1}^{7} w_j \cdot s_j$$

ここで，i は i 番目の検体，j は j 番目の分類器である．

4. 評価実験

各特徴（出現回数, 時間間隔の平均値・最大値・最小値, メモリ使用量の平均値・最大値・最小値）に基づく分類器を構築し, マルウェア検知の予測精度を比較する. その際, API の遷移数は，2 遷移だけでなく，3 遷移と 4 遷移についても動的解析に基づくログから特徴を抽出し，2 遷移と同様の方法で比較を行う．また，各分類器から算出されたスコアに重み付けをして足し合わせた値によってマルウェアを検知する提案手法の評価を行う．以下に，使用検体，評価指標，実験結果について説明する．

4.1 使用検体

正規プログラムの検体は，先行研究[8]と同様に，Vector[10]から収集したファイルを用いる．そのうち，zip や tar.gz などで圧縮されていたファイルは解凍し，実行形式のファイルを用いた．マルウェア検体についても，先行研究 [8]と同様に，独自にマルウェアが配布されているサイトを紹介しているサイトから収集したものと同じものを用いた．実験では，評価用検体として 70%，学習用検体として 30%用いている．マルウェア・正規プログラムともに 50%の割合で使用している．

4.2 評価指標

特徴ごとに構築した分類器及び各分類器から算出されたスコアに重み付けして足し合わせた値を用いた WSM の予測精度の評価は，5-分割交差検定によって行う．この検定では，用意した検体データセットを 5 個に分割する．そして，そのうちの 1 つの検体データセットをテスト用のデータセットとして，残る 4 個を学習用のデータセットとする．5 個分割されたデータセットをそれぞれテスト用のデータセットとして 5 回検証を行い，それらの検証結果を平均して 1 つの評価指標を得る検定方法である．

評価指標は，ROC 曲線を作成した時に，その曲線より下の部分の面積 Area Under the Curve（AUC と呼ぶ）を用いる．AUC は，0 から 1 までの値をとり，値が 1 に近いほど判別能力が高いことを示し，値が 0.5 のとき，判別能力がランダムであることを示す. ROC 曲線の作成と AUC の計算は，統計分析ソフトの R の ROCR パッケージを用いる．

4.3 各特徴に基づく分類器の予測結果

API 呼び出しとそれに伴う経過時間とシステム負荷を用いてマルウェア検知する問題に対して，どの機械学習法が実際に適しているのかを調査するために，各特徴に対して 4 つの機械学習法(NB，SVM_{poly}，SVM_{RBF}，RF)を用いて分類器を作成し, それらの予測精度を評価した. 表 1 は，7 つの特徴ごとに構築した分類器の予測精度を示している．また，表中の遷移の項目は，API 呼び出しパターンの遷移数を表している．例えば，API1→API2 のように，API 1 呼び出し後に, 次の API2 を呼び出したパターンの遷移数は 2 となる．また，API1→API2→API3 のように，API1 呼び出し後に次の API2 を呼び出し，その後に次の API3 を呼び出したパターンの遷移数は 3 となる．なお，提案手法では，ある API 呼び出し後に次の API を呼び出すまでに経過する時間を想定しているため，遷移数 1 は考えないこととする. さらに表 1 では, AUC の記載されているセルの背景を，赤（最も高い AUC 値）から白（最も低い AUC 値）までの色で表している．

結果として，既存手法で提案されている出現回数を用いた場合, 2 から 4 遷移までのいずれにおいても, RF の AUC が 0.89 であり，4 つの機械学習法の中で最も予測精度が高かった．また RF は 7 つの特徴のいずれにおいても，他の機械学習法に比べて高い精度でマルウェアを予測できることがわかった（表 2）．出現回数以外の本研究で提案する特徴に基づく分類器の中で，出現回数に基づく分類器よりも高い予測精度を示した分類器は，2 遷移の場合は，メモリの最小値に基づく分類器（AUC=0.90），3 遷移の場合は，時間の最小値に基づく分類器（AUC=0.91），4 遷移の場合は，時間の最小値またはメモリの平均値に基づく分類器（AUC=0.90）であった．

分類器別にみると，NBC は 3 遷移が最も AUC が高くな

る傾向があった．また SVM_{poly} は 4 遷移が高くなる傾向があり，SVM_{RBF} は，2 遷移が高くなる傾向があった．RF に限っては，2 から 3 の遷移数に関係なく，いずれの場合においても，ほぼ同じ AUC であった．すなわち，RF 以外の機械学習法は，遷移数が精度に影響する傾向があったが，RF は影響する傾向はほとんど見られなかった．AUC の平均値は，RF が最も高く 0.885，NBC が 0.727，SVM_{poly} が 0.766，SVM_{RBF} が 0.611 となっている．

以上の各特徴に基づく分類器の予測精度の評価から，いずれの特徴及び遷移において，RF が最も高い予測精度を示した．すなわち，本研究が対象とする問題に対して，RF が最も有効性が高いと考えられる．

表 1　各特徴に基づく分類器の精度

Figure 1.　Performance of Classifiers based on each feature

特徴	分類器	AUC		
		2 遷移	3 遷移	4 遷移
出現回数	NBC	0.7	0.76	0.74
	SVM_{poly}	0.8	0.81	0.83
	SVM_{RBF}	0.71	0.61	0.58
	RF	0.89	0.89	0.89
時間平均	NBC	0.68	0.74	0.73
	SVM_{poly}	0.71	0.79	0.79
	SVM_{RBF}	0.72	0.57	0.56
	RF	0.89	0.87	0.88
時間最大	NBC	0.69	0.74	0.73
	SVM_{poly}	0.69	0.79	0.8
	SVM_{RBF}	0.71	0.57	0.55
	RF	0.89	0.89	0.88
時間最小	NBC	0.69	0.74	0.72
	SVM_{poly}	0.71	0.79	0.81
	SVM_{RBF}	0.7	0.57	0.56
	RF	0.89	0.91	0.9
メモリ平均	NBC	0.71	0.75	0.74
	SVM_{poly}	0.72	0.76	0.79
	SVM_{RBF}	0.67	0.59	0.57
	RF	0.87	0.89	0.9
メモリ最大	NBC	0.69	0.78	0.75
	SVM_{poly}	0.71	0.74	0.78
	SVM_{RBF}	0.66	0.57	0.56
	RF	0.87	0.89	0.88
メモリ最小	NBC	0.72	0.74	0.76
	SVM_{poly}	0.73	0.79	0.76
	SVM_{RBF}	0.66	0.58	0.57
	RF	0.9	0.9	0.88

表 2　2 遷移 WSM の精度

Table 2.　Performance of WSM for 2-transition

分類器（2 遷移）	各特徴の重み							AUC
	出現回数	時間平均	時間最大	時間最小	メモリ平均	メモリ最大	メモリ最小	
NBC	0.36	0.00	0.14	0.18	0.25	0.07	0.00	0.78
SVM_{poly}	0.47	0.05	0.00	0.11	0.16	0.16	0.05	0.84
SVM_{RBF}	0.09	0.91	0.00	0.00	0.00	0.00	0.00	0.74
RF	0.25	0.08	0.08	0.00	0.00	0.00	0.58	0.91

表 3　3 遷移 WSM の精度

Table 3.　Performance of WSM for 3-transition

分類器（3 遷移）	各特徴の重み							AUC
	出現回数	時間平均	時間最大	時間最小	メモリ平均	メモリ最大	メモリ最小	
NBC	0.45	0.05	0.09	0.36	0.00	0.05	0.00	0.77
SVM_{poly}	0.59	0.00	0.12	0.00	0.06	0.00	0.24	0.86
SVM_{RBF}	0.21	0.71	0.00	0.00	0.07	0.00	0.00	0.61
RF	0.08	0.00	0.00	0.75	0.00	0.08	0.08	0.91

表 4　4 遷移 WSM の精度

Table 4.　Performance of WSM for 4-transition

分類器（3 遷移）	各特徴の重み							AUC
	出現回数	時間平均	時間最大	時間最小	メモリ平均	メモリ最大	メモリ最小	
NBC	0.00	0.00	0.86	0.00	0.00	0.14	0.00	0.77
SVM_{poly}	0.50	0.00	0.00	0.36	0.14	0.00	0.00	0.85
SVM_{RBF}	0.50	0.00	0.00	0.00	0.00	0.50	0.00	0.59
RF	0.07	0.00	0.00	0.71	0.21	0.00	0.00	0.90

4.4　Weighted Sum Model に基づく予測結果

各特徴の分類器で算出されたスコアに重み付けし足し合わせた値を用いて予測を行なった．表 3 から 4 は，2 遷移，3 遷移，4 遷移の WSM の AUC と各特徴の重みを示す．なお，各特徴に基づく各分類器のスコアに対する重みの総和が 1.0 になるように正規化した際の重み（少数第 2 位ま

で）を示している．また重みの値の組み合わせが異なるにもかかわらず，同じ AUC になる場合がある．表では，各機械学習法に基づく分類器相互で，重みの値の組み合わせの相関係数を求め，その積の値が最も高かった重みの値の組み合わせを記載している．

結果として，RF の予測精度は，いずれの遷移において最も高い予測精度(AUC>=0.90)であった．2 遷移の場合，出現回数のみに基づく RF の AUC が 0.89 であり，WSM の 2 遷移の AUC が 0.91 であることから，0.02（2%）の予測精度の改善をすることができた．すなわち，出現回数だけでなく，メモリの最小値や時間の平均値・最小値を加えたことにより，マルウェア検知の予測精度が向上したと考えられる．特に，出現回数の重みが 0.25 であるのに対して，メモリの最小値の重みは 0.58 であることから，後者は予測に約 6 割程度寄与していることがわかった．3 遷移と 4 遷移の RF においても，同様の改善が見られた．この場合，メモリの最小値ではなく，時間の最小値が約 7 割程度も寄与していることがわかった．これは，メモリの最小値に，判別能力がないことを示唆しているのではない．実際に，4.3 節の表 1 では，メモリの最小値のみを用いた RF は高い AUC の値を示している．WSM によって AUC の値を最大化するために，各分類器からの算出されたスコアが予測に寄与する割合を重みによって調整した結果であると考える．

以上のことから，既存手法で提案した出現回数だけでなく，本研究で提案する特徴を加えることによって，マルウェアの検知能力を向上させることができることがわかった．

5. 今後の課題

本研究では，先行研究[8]の結果を基に，更なるマルウェア検知率の向上のために，特徴ごとに分類器を作成し，それから算出されたスコアに重み付けをして足し合わせた値によってマルウェアを検知する手法を提案した．その結果として，各特徴に基づく分類器の予測精度の評価から，いずれの特徴及び遷移において，RF が最も高い予測精度を示すことがわかった．WSM に基づく予測結果としては，RF の 2 遷移と 3 遷移では，既存手法より 2%の判別能力の改善を行うことができた．今後は，マルウェアの種別を判別することができる検知手法や機能の分類を行える検知手法の提案も行う予定である．

参考文献

[1] Thriving Beyond The Operating System: Financial Threat Group Targets Volume Boot Record FireEye, https://www.fireeye.com/blog/threat-research/2015/12/fin1-targets-boot-record.html, 参照 July 28, 2017

[2] An Overview of Malware Self-Defense and Protection － McAfee Blogs https://securingtomorrow.mcafee.com/mcafee-labs/overview-malware-self-defense-protection/, 参照 Aug 28, 2017

[3] 青木一樹，後藤 滋樹，“マルウェア検知のための API コー ルパターンの分析,”電子情報通信学会総合大会講演論文集 2014 年 情報・システム， Vol.2， No.179， 2014-03-04.

[4] 川口 直人， 面 和成 “動的解析ログの API を用いた機能に基つくマルウェア分類” 暗号と情報セキュリティシン ポジウム 2015.

[5] 笠間貴弘，吉岡克成，井上大介，松本勉，“実行毎の挙 動の差異に基づくマルウェア検知手法,”コンピュータセ キュリティシンポジウム 2011 論文集， No.3 pp. 726-731, 2011-10-12.

[6] 酒井崇裕，竹森敬祐，安藤類央，西垣正勝，“侵入挙 動の反復性を用いたボット検知手法,”コンピュータセ キュリティシンポジウム 2009 論文集， Vol.51 No.9 pp. 1591-1599, 2010-09-15.

[7] 佐藤順子, 三須剛史, 花田真樹, 鈴木英男, 布広永 示，“API 呼び出しとシステム負荷を用いたマルウェア の特徴抽出に関する一検討,” コンピュータセキュリティ シンポジウム 2016 論文集， Vol.2016 No.2 pp. 305-309， 2016-10-04.

[8] 佐藤順子, 花田真樹, 面和成, 山口崇志, 鈴木英男 ， 布広 永示， 折田 彰， 関口 竜也 ，“API 呼び出しとそれ に伴う経過時間とシステム負荷を用いたマルウェア検知 手法”, コンピュータセキュリティシンポジウム 2017 論文 集, Vol.2017 No.2， 2017-10-16.

[9] CuckooSandbox https://www.Cuckoosandbox.org/

[10] Vector http://www.vector.co.jp/