サンドボックスを利用した未知マルウェア検出精度向上に関する一検討

市田 達也† 須藤 年章‡ 髙森 覚†

†NTT コムセキュリティ株式会社
105-7104 東京都港区東新橋 1-5-2 汐留シティセンター 4,5F
{tatsuya.ichida, s.takamori}@ntt.com

‡NTT コミュニケーションズ株式会社
108-8118 東京都港区芝浦 3-4-1 グランパークタワー16F
t.sudou@ntt.com

あらまし マルウェア解析技術の一つにサンドボックスと呼ばれる仮想環境での動的解析技術がある．本技術は，多機能化や難読化されコード解析が難しくなっている近年のマルウェアの挙動を解析するにあたり有益であるが，一方で悪性判定の閾値によってはマルウェアには値しない正常なファイルを過検知する事象も確認されている．本研究では産業面でネットワークトラヒック内のファイルに対し本技術によるマルウェア検出を行う上で，過検知の削減という観点からマルウェア検出精度向上のための特徴量を抽出し，その評価および課題を考察する．

A Study for improvement of unknown malware's detection accuracy on Sandbox Analysis

Tatsuya Ichida† Toshiaki Sudoh‡ Satoru Takamori†

†NTT Com Security (Japan) KK
Shiodome City Center 4,5F, 1-5-2,Higashi Shinbashi, Minato-Ku,Tokyo 105-7104,JAPAN
{tatsuya.ichida, s.takamori}@ntt.com
‡NTT Communications Corporation
Gran Park Tower 16F, 3-4-1, Shibaura, Minato-ku, Tokyo 108-8118, JAPAN
t.sudou@ntt.com

Abstract Dynamic Analysis on virtual machine called "Sandbox Analysis", is known as one of the malware analysis methods. This is useful against recent packed and obfuscated malware which is difficult to analyze by reading program codes statically. On the other hand, it raises False Positive related to the threshold value to decide as malicious. In this study, we explore and evaluate the features for the improvement of unknown malware's detection accuracy based on reducing "False Positive".

1　はじめに

マルウェアを利用したサイバー攻撃・犯罪が際立っている近年において，IPA が発行した「2013 年度 情報セキュリティ事象被害状況調査」[1]によると，日本企業におけるマルウェア遭遇率がはっきりと増加傾向にある．主な侵入経路は，Web サイト閲覧，電子メール，USB の順であった． グローバルの傾向としても，標的型攻撃アプライアンスベンダー「FireEye」の調査によると，企業は 1.5 秒ごとにマルウェアに関連した攻撃を受けており，日本も攻撃対象国の上位に含まれている[2]．

企業のマルウェア対策としては，ゲートウェイ型アンチウィルススキャン，エンドポイント型アンチウィルススキャン，IDS/IPS のシグニチャマッチングがあるが，これらは主に既知のマルウェアは検知できるが，シグニチャが対応していない未知のマルウェアを検知することは容易ではない．マルウェアをシグニチャで検知することが難しくなった背景としてマルウェアの高度化が関係している．ダウンローダーのように単体では悪性挙動が少なくシグニチャにて検出されにくいものや，マルウェアのプログラムコードが動的に難読化され，かつ亜種の大量生成ツールの出回りにより，シグニチャを容易に作成し難くなっている．このようなマルウェアの高度化により，セキュリティベンダーのシグニチャ配信にも遅延が発生している．またマルウェアによる攻撃は短期間に行われることが多い[3]ため，シグニチャが生成された頃には攻撃は存在しなくなっていることもある．

これらのシグニチャベースで対策が難しいマルウェアに有効な解析技術の一つにサンドボックスと呼ばれる仮想環境での動的解析がある．特に産業面では，感染から情報の搾取まで数時間以内で行う未知のマルウェアの攻撃に対しては，サンドボックスの解析機能のみが有効な手段である．最近では日本企業においてもサンドボックス技術を利用した防御装置が導入されはじめている．

2　サンドボックスによる動的解析

2.1　マルウェア検知の仕組み

サンドボックスによる動的解析は，実ネットワークへの感染リスクを軽減するため，隔離された仮想環境にて実行される．被疑ファイル（以降，検体）を仮想環境にて意図的に実行し，プログラムの挙動をトレースしたログを出力する．サンドボックスは，通信先のブラックリスト，マルウェアも散見される特徴的な挙動，および挙動の状態遷移情報を悪性スコアと紐づけ，判定ルールとして保持している．仮想環境でのプログラム実行時に，トレース結果がその判定ルールにマッチした場合に悪性スコアを加算する．その後，トレース結果が判定ルールに一致するたび，悪性スコアは累積加算され，一定の閾値を超えた場合，マルウェアと判定する仕組みである．

サンドボックス技術の主な優位性は，プログラムコードが暗号化されていても，コードではなく実行結果に基づき検知できるため，未知のマルウェアを検知できる点，また産業面では仮想環境を自社環境に合わせることで，自社環境にて影響のあるマルウェアのみを検知できる点である．しかしながら一方で，検知精度について過検知(False Positive)があり，実務上の運用において，インシデント対応稼働の増大およびインシデントレスポンス品質の低下という課題を引き起こしている．

2.2　過検知した際の運用課題

動的解析によるマルウェア検知の大きな課題として，過検知(False Positive)がある．これはマルウェアでない正常なファイルを「マルウェア」として検出したり，脅威度の高い悪性挙動がなく，マルウェアであると言い難いグレーなファイルを検出することである．

一般的に企業ではマルウェアが検出されると，端末のネットワークからの隔離や，最新シグニ

チャでのアンチウィルスソフトの完全スキャン，マルウェアの通信先を Web プロキシの URL フィルタに登録する等の対応を行うことがあるが，検出されたマルウェアが過検知であった場合，対応稼働の浪費に始まり，隔離端末のユーザの生産性の低下，プロキシの URL フィルタ設定による正常通信の誤遮断を引き起こす．またマルウェア感染インシデント発生時に最も有効な対策は，OS のクリアインストールであるが，物理端末リソースが限られている中で対処するには過検知を無くし，脅威度の高い真のマルウェアだけに注力する必要がある．

真のマルウェアであるかの確認方法として，アンチウィルスソフトでの“ヒューリスティック（あいまい）検知でない”シグニチャ検知や複数アンチウィルスベンダーでのスキャン検知結果の参照が有効であるが，これらもシグニチャが対応しない未知マルウェアにおいては機能しない．マルウェアコードの静的解析ができる技術者が存在するならば，対応可能かもしれないが，人的リソースよりも未知マルウェア数の方が多いことが一般的であるため，ボリュームに対応しきれない．

一方で，過検知が発生するたびに，その都度後追いでデバイス検出ルールの修正（ホワイトリスト対応など）をするのでは，稼働の消費および見逃し(False Negative)をする可能性があり，進化し続けるマルウェアに追いつけないと考える．サンドボックス解析の仕組み上，利用環境の違いや新しいネットワークサービスやアプリケーションの登場により新たな動作不具合やいままでとの挙動逸脱により，過剰に反応することがあるため，トレース結果の挙動ログを基にどう判断するかが実務上重要なポイントとなる．

そのため，実務上，サンドボックスにて動的解析された挙動ログを独自の観点で分析，評価が必要である．真のマルウェアを抽出するにあたり，本手法の利点は難読化コードを解読する静的解析ほど技術的に難しくない点，サンドボックスの挙動ログを利用し機械学習等を用いた自動最適化を行いやすく，高速かつ機械的に過検知を減らせる点である．

本研究では，サンドボックスでの挙動ログより過検知に深く関連する特徴を抽出し，サンドボックス自体の検出ルールのチューニングではなく，その結果の挙動ログ分析エンジンのチューニングによって未知マルウェアの検知精度向上を図る．サンドボックスで検知された検体から過検知を機械的に除外し，脅威度の高い真のマルウェアのみを抽出することで，インシデント対応稼働の削減およびインシデントレスポンス品質向上を目指す．

2.3　過検知例の紹介

弊社独自環境にて Windows 系 OS を実装したサンドボックスにて検知された検体の内，ヒューリスティック検知を除いたアンチウィルススキャン結果および業務にて利用したファイルかどうかのヒアリングによって過検知と判断できたファイルは，表1のように大きく分けると5つに分類できた．

表1．サンドボックス過検知ファイル

ファイル種別	特徴的な挙動
①一部のアドウェア	・URL ブラックリストに存在しない広告コンテンツへの Web アクセス ・ウィンドウ表示 ・スリープ挙動
②インストーラ	・ユーザ許諾が必要な場合，動作が停止 ・ファイルを多数生成 ・レジストリを多数改変 ・スリープ挙動
③パッキングされたツール	・一時的に実行ファイルの生成 ・スリープ挙動 ・プロセスの多数起動
④自己解凍書庫ファイル（exe）	・ウィンドウ表示 ・スリープ挙動 ・ファイル削除
⑤オブジェクトの多い PDF	・オブジェクトに関するメモリの動的割当て

このような過検知ファイルが存在する原因は，サンドボックスによるマルウェア検知がスコア閾

値を持った判定ルールによるパターンマッチングであり，かつ，解析途中に動作が停止した場合を考慮したスコア構成にある．特に解析の停止が発生した場合には，それまでの悪性スコアの累計値によって判定する必要あるため，マルウェアの見逃しを防ぐために，悪性スコアを高めに設定することがある．

次に上記5種類のファイルについて，過検知原因を考察した結果を列挙する．

① 一部のアドウェア
- マルウェアかどうかグレーではあるが，脅威度の高い挙動がなく，アンチウィルスベンダーが最終的にパターン対応しないファイルは過検知として扱う．特徴的な挙動として，自身のダウンロード元 URL であるソフトウェア配信サイトへのコールバック通信や別のフリーツールダウンロードページへの HTTP HEAD メソッドによるアクセスや検体が自身を別プロセスで起動しなおす連鎖実行挙動のスコアが高く過検知に至りやすい．

② インストーラ
- 自身のダウンロード元 URL であるソフトウェア配信サイトへのコールバック通信や内部の一時ファイル作成挙動が際立つ．GUIプログラムにてユーザの許諾待ちによる動作停止も一部見受けられ，完全に解析されずに過検知に至りやすい．

③ パッキングされたツール
- 外部通信挙動はないが Setup.exe のような実行ファイルを内部生成挙動，またパッカーの利用が悪性スコアを上昇させ一般的に過検知に至りやすい．

④ 自己解凍書庫ファイル（exe）
- パスワード暗号化した自己解凍書庫ファイルをサンドボックスで解析すると，パスワードを入力させるウィンドウの表示した時点でユーザインタラクション待ちにより動作が停止する．しかしそれまでの端末情報を取得するAPIコールや静的解析によるexeファイル削除コードにより，完全に解析されずに過検知に至りやすい．

⑤ オブジェクトの多い PDF
- PDF文書内の画像や表に用いるStreamオブジェクトのメモリ割当方法は，PDF 実行時に動的にオブジェクトの個数分のメモリ領域が割り当てられる．オブジェクトの個数によっては，この挙動がバッファオーバーフロー攻撃に利用される「ヒープスプレー攻撃」に類似するため，過検知に至りやすい．

以上より，過検知されたファイルには，マルウェアと類似する特徴的な挙動があることがわかった．また①～④の過検知ファイルには共通して1回以上のスリープ挙動が確認された．本稿では，過検知除外における真のマルウェア識別のために，挙動ログより特徴量を抽出する．

2.4 挙動ログを用いた関連研究

Windows OS の API コールがほとんどのマルウェアにおいて利用されている．参考文献[4][5]からも API コールがマルウェア動的解析に有効であるとの見解があるため，API コールを用いて精度向上を検討された研究を紹介する．

藤野ら[6]は，API コールに関して「API 関数名」と「パラメータ引数」の組を単語として登録し，それを前処理した上で k-means 法および非負値行列因子分解(NMF)を用いてクラスタリングを行っている．各 API 関数やパラメータの意味を考慮していない点が課題ではあるが，パラメータ引数がクラスタリングに有効である点が示されている．

また青木ら[7]は特徴量に API コールの時系列パターンを用いている．決定木による識別を行い，連鎖要素数と検知精度に依存関係はなく，マルウェアの特徴に依存するとの知見が得ら

れている.

クラスタリングを行いマルウェアと正常ファイルの境界面を作成することは，学習し続けることで，柔軟に未知のマルウェアにも対応できる長所があるが，一方で学習検体に依存するため，クラスタが有効に分かれない場合も考えられる.

一方で，仲小路ら[8]は，サンドボックス検知にて過検知文書が見つかった場合に自社内のみで解析が完結できるように独自エンジンを検討している. 特に環境依存型マルウェアへの検討が進んでおり，マルウェアの特徴的な挙動として，デバッガ検知，プロセスへのコードインジェクション，時限的発動処理，外部ネットワーク接続判定の有無に着目している.

本研究では，サンドボックスを利用した未知マルウェア検知精度向上のために，動的解析エンジン内の判定ルールの内，よりマルウェアらしい特徴量に対するスコア重みを加算し，よりマルウェアらしくない特徴量に対するスコアの重みを減算することで，一般的なマルウェアと一部動作に類似性が見受けられるファイルの検知精度の向上を検討する.

3 提案手法

3.1 検知精度向上に有効な特徴量指標

弊社環境にて検出された 2.3 節の過検知ファイルの特徴により，これら過検知ファイルにはなく，マルウェアファイルに見られた挙動ログの特徴量を2つ抽出した. 特徴量を抽出する際の指標は既存のサンドボックスエンジン同様に以下を用いた.

- 正常ファイルの挙動との逸脱性
- マルウェアファイルの挙動との類似性

過検知ファイルの特徴の共通性から，Sleep 関数の API コールに着目した. そしてマルウェアがスリープ挙動を実行する際の，プロセス ID（以下，PID）とそのプロセス名の関連性に正常ファイルの挙動との逸脱した特徴を確認した. 一方，マルウェアファイルの挙動との類似性から，近年マルウェアは感染すると1時間以内に自身を削除する[3]という特徴に着目し，マルウェア自身の削除挙動に着目した. 次節より各特徴量の詳細を述べる.

3.2 マルウェアと正常ファイルのスリープ挙動の相違

2.3 節の①～④の過検知ファイルの特徴として，ユーザインタラクションを求められ，動的解析が途中終了しているものが多く見られた. これはインストーラ等のウィンドウを表示させる GUI プログラムにて多く，ユーザの入力待ち状態にて Sleep 関数がコールされ続けているためであった. Sleep コール自体はマルウェアの潜伏挙動にも確認される特徴であるが，Sleep を呼び出した PID，プロセス名の関係に着目するとマルウェアファイルと正常ファイルに下述の明確な違いが確認できた.

本稿では API 関数「Sleep」，「SleepEx」，「SleepConditionVariableCS」を対象とする.

- インストーラやアドウェア等のファイル

Sleep コールは同一検体ファイルより 1PID のみにてコールされていた. サンドボックスではユーザ操作が発生しないため，処理が進まず，Sleep コールはこの1回のみであることが多い.

- マルウェア

同一検体ファイルから 2PID 以上プロセスが起動し，コールされていた. 1回目は起動直後にコールされる. その後レジストリ改変等の別操作を行った後に，その同一検体のプロセスが別 PID にて起動して再度 Sleep コールを行う.

本特徴を以下では，Sleep 特徴量とする.

3.3 マルウェアが自身を削除する挙動

マルウェアが役目を終えて自身を削除したり，他のファイル，プロセスに挙動を遷移し，隠蔽のため検体自身を削除する挙動は，近年のマ

ルウェアによく見られる挙動である．対象の API 関数は「DeleteFile」であるが，呼び出し元のプロセスと削除対象ファイルも特徴量のパラメータとして利用する．

本特徴を以下では，Delete 特徴量とする．

4　特徴量評価実験

上記2つの特徴量が検知精度向上のために，マルウェアと正常ファイルの識別境界面に与える効果を一次評価するために，簡易評価実験を行った．

4.1　実験データ

解析対象となる検体の挙動ログにおいて
以下の実験データを用いる．

表2．実験データ

マルウェア検体（Windows 7 動作）	
Ⅰ.FFRI Dataset 2014[9]	3000 種
Ⅱ.FFRI Dataset 2013[9]	2638 種
Ⅲ.弊社独自取得マルウェア	1710 種
正常ファイル（Windows 7 動作）	
Ⅳ.Windows 正常実行ファイル	384 種
Ⅴ.弊社検出の過検知ファイル	154 種

※Ⅰ，Ⅱ，Ⅳ，Ⅴ　挙動ログ取得時間　90 秒
※Ⅲ　挙動ログ取得時間　500 秒
※Ⅳ　Windows OS にデフォルトでインストールされている実行ファイルや信頼できるサイトからのダウンロードしたフリーソフト
※Ⅴ　上記①～⑤種類の過検知ファイルを含む

4.2　評価方法

本稿にて取り上げた2つの特徴量を評価するにあたり，まず表2のⅠ，Ⅱ，Ⅳ，Ⅴのデータに対して，上記の Sleep，Delete API を含めマルウェア解析のための主要 API[10]の1検体ごとの平均コール数を調査した．この結果を図 1，図2の API コールのヒストグラムにて示す．次に各データセットにおける提案手法の Sleep，Delete 特徴量の該当検体数を調査し，本特徴量が該当したマルウェアの分類を表3に示す．

4.3　実験結果および考察

4.3.1　API コールのヒストグラム

図1，2の通り，「DeleteFile」および「Sleep」系 APIコールは，検体ごとの平均APIコール数において，マルウェアと正常実行ファイルおよび過検知ファイルのすべてに確認され，「Sleep」コールについては特に他の API コールに比べ差異が少ないことが確認された（レジストリ操作は除く）．よって本稿においても単一の API コールのみの特徴ではマルウェアと正常ファイルの識別境界を定めることは難しいと考える．次に API コールの実行プロセスやパラメータも活用している提案手法の Sleep，Delete 特徴量にて，マルウェアと正常ファイルの違いを示す．

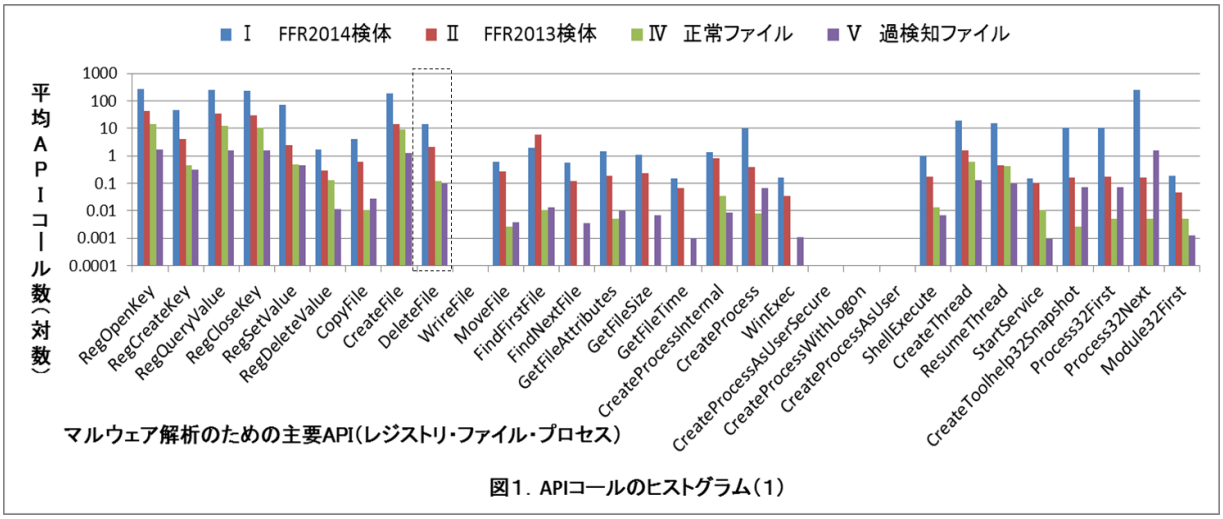

図1．APIコールのヒストグラム（1）

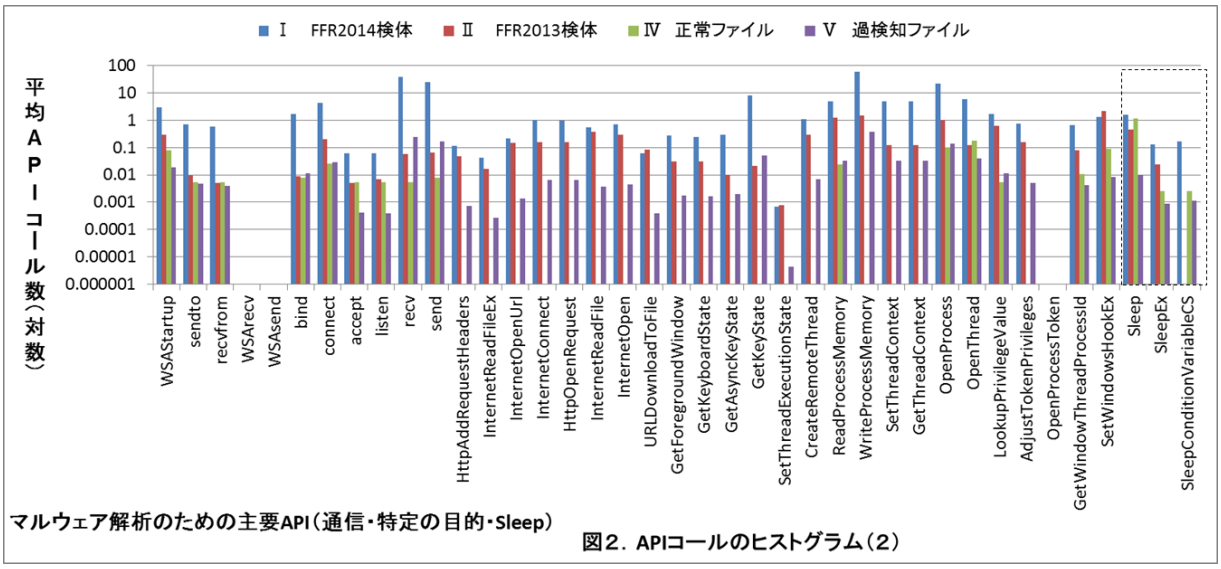

図2．APIコールのヒストグラム（2）

4.3.2　特徴量が該当した検体数

表3．特徴量の該当数

実験データ	Sleep 特徴量	Delete 特徴量
Ⅰ 3000 種	342 種	43 種
Ⅱ 2638 種	28 種	23 種
Ⅲ 1710 種	222 種	126 種
Ⅳ 384 種	0 種	0 種
Ⅴ 154 種	0 種	0 種

表3より，両特徴量ともにマルウェア検体のみで確認され，正常・過検知ファイルにおいて確認されず，実験データⅢより挙動ログ取得時間が長いほど，該当数が多くなる結果となった．

また実験データⅠ，Ⅱについて，Sleep 特徴量，Delete 特徴量が該当したマルウェアに関してカスペルスキー社での検知名の上位10種（亜種は統合）を図3，4にそれぞれ示す．

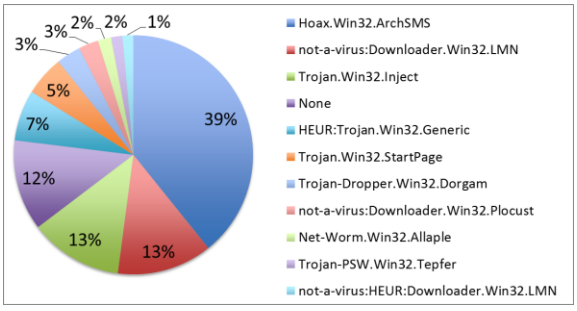

図3．Sleep 特徴量の該当マルウェア分類

Sleep 特徴量では，実験データⅠ（FFRI2014）にて，「Hoax.Win32.ArchSMS」というSMSでのやり取りを介して，ユーザは必要なファイルを受け取る代わりに金銭を盗み取られる偽のパッカー[11]が全体の4割を占めた．また一部ドロッパーやダウンローダーという名称にても検知されている．実験データⅡ（FFRI 2013）では「Zbot」や「Worm.Win32.ALLaple」が確認された．

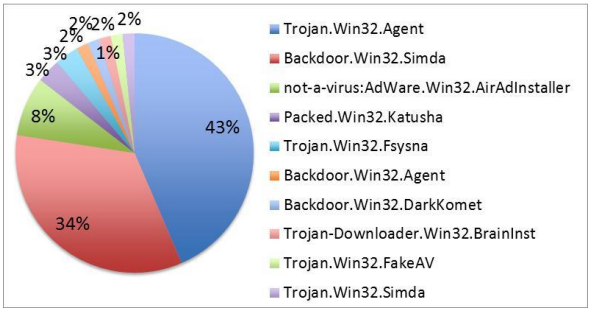

図4．Delete 特徴量の該当マルウェア分類

一方で，Delete 特徴量では実験データⅠ，Ⅱを通して「Trojan.Win32.Agent(Dropper)」や「BackDoor.Win32.Simda」にマルウェアが偏っていた．これらは一般的に確認されているドロッパーが別のマルウェアをドロップして検体自身を削除する特徴や，多くのバックドアに当てはまる，常駐しているプロセスにコードインジェクションをした後，検体自身を削除する特徴に一致する．

5　まとめと今後の検討

本稿では，「サンドボックスを利用した未知マルウェア検出精度向上に関する一検討」としてサンドボックス技術の大きな課題に過検知「False Positive」が存在すること，産業面での過検知対応の課題を述べた．また提案手法として，過検知ファイルの特徴を活かしたマルウェアと正常ファイルを識別できる Sleep 特徴量・Delete 特徴量を提案した．さらに簡易評価実験においてこれらの特徴量を用いると過検知ファイルをすべて正常「True Negative」と判定できた．つまりこれらの特徴量を利用すると，過検知ファイル 154 種分，本実験データ全体における約 2%の検体が正しく判定されるため，本特徴量は検知精度向上の一つの手段となりうる．また本特徴量が該当するマルウェア種別にも傾向が見えた．

一方で Sleep 特徴量，Delete 特徴量のみでマルウェアだと識別することは新たな過検知「False Positive」を生む可能性があるため，活用方法としてはこれらの特徴量にも悪性スコアを付けて悪性判定までの1コンポーネントとして利用することを考える．今後の検討として，これら悪性スコアの重み付け手法の検討を行い，評価実験を元に，最も精度向上できた悪性スコアを選定する．また引き続き新たな過検知ファイルの特徴抽出を行いながら，他の API の評価およびその組み合わせの検討も行っていく．

参考文献

[1]「2013 年度 情報セキュリティ事象被害状況調査-報告書-」
http://www.ipa.go.jp/files/000036465.pdf（参照 2014/08/24）

[2]「FireEye 高度な攻撃に関する脅威レポート: 2013 年版」
http://www2.fireeye.com/advanced-threat-report-2013-ja.html?x=FE_WEB_IC
（参照 2014/08/24）

[3] Zheng Bu, Rob Rachwald,”Ghost-Hunting With Anti-Virus,” FireEye Blog, May 5 2014.
http://www.fireeye.com/blog/corporate/2014/05/ghost-hunting-with-anti-virus.html（参照 2014/08/24）

[4] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Krgel, and E. Kirda, “Scalable, behavior-based malware clustering.,” in NDSS, The Internet Society, 2009.

[5] M. Alazab, S. Venkataraman, and P. Watters, “Towards Understanding Malware Behaviour by the Extraction of API Calls,” in Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second, pp. 52-59, 2010.

[6] 藤野朗稚,森達哉,”自動化されたマルウェア動的解析システムで収集した 大量 API コールログの分析,” MWS 2013(2013 年 10 月).

[7]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， D-19-3,2014(2014 年 3 月).

[8]仲小路ら, ”進化する標的型攻撃に対抗するマルウェア自動解析技術,” 日立評論，社会インフラセキュリティ, Vol.96 No.03 218-21,
http://www.hitachihyoron.com/2014/03/pdf/03a12.pdf(参照 2014/08/24)

[9] 秋山満昭, 神薗雅紀, 松木隆宏, 畑田充弘, ″マルウェア対策のための研究用データセット～MWS Datasets 2014～,″ 情報処理学会 研究報告コンピュータセキュリティ（CSEC） Vol. 2014-CSEC-66, No. 19, pp. 1 - 7, 2014.

[10] 新井悠, 岩村誠, 川古谷裕平, 青木一史, 星澤裕二, ”アナライジング・マルウェア フリーツールを使った感染事案対処,” オライリー・ジャパン,2010 年 12 月

[11]カスペルスキーラボ, “マルウェアマンスリーレポート : 2010 年 10 月,”
http://www.kaspersky.co.jp/info?id=207581643
（参照 2014/08/24）