非 Windows におけるサンドボックスを利用したマルウェア検出に関する一考察

山本 粋士 †　　平木 康介 †　　須藤 年章 †

†NTT コミュニケーションズ株式会社
108-8118 東京都港区 芝浦三丁目 4-1
{kiyo.yamamoto, k.hiraki, t.sudou}@ntt.com

あらまし シグネチャ検知されない未知のマルウェアへの対策として，サンドボックスによる動的解析の導入が広まっている．他方，近年の Windows に限らない OS の利用増加に伴い，非 Windows を狙ったマルウェアも増加の傾向にある．そのため非 Windows に対してもサンドボックスによる対策の必要性が高まってきている．そこで本稿では，Windows を狙ったマルウェアのサンドボックス内での典型的な挙動が，OS X のマルウェア検出に利用可能か検証実験を行った．その結果，Windows においてマルウェア検出への有効性が確かめられた挙動が，OS X を狙ったマルウェアにおいても同様に検知可能なことがわかった．

A Note of Malware Detections in non-Windows using a Sandbox

Kiyohito Yamamoto†　　Kousuke Hiraki†　　Toshiaki Sudou†

†NTT Communications Corporation.
3-4-1 Shibaura, Minato-ku, Tokyo 108-8118, JAPAN
{kiyo.yamamoto, k.hiraki, t.sudou}@ntt.com

Abstract Dynamic analysis on sandbox has become popular in order to detect unknown malwares, which couldn't be detected by signatures. On the other hands, malwares act on non-Windows OS have also been widespreading with the increasing utilization of non-Windows OS. In this paper, to keep up with those recent trend, we extracted some typical behaviors of malwares act on Windows, also extracted from those act on OS X, and compared. As a result, we found typical behaviors of Windows malwares are also detected on OS X malwares.

1　はじめに

標的型攻撃に代表されるサイバー攻撃では，シグネチャでは検知されない未知のマルウェアがしばしば利用される．シグネチャ検知が困難となっている要因の一つとして，マルウェアのプログラムコードの難読化によるシグネチャ作成の高度化，それに伴う作成の遅延が挙げられる．そのためシグネチャによらないマルウェア検出の方法が必要となっており，その一つにサンドボックスによる動的解析の利用がある．サンドボックスを利用することでコードの難読化に依らない，仮想環境におけるファイルの実行結果によるマルウェア検出が可能となる．そのため近年では従来のアンチウィルスソフトに加え，サンドボックスの導入が広まっている．

他方，近年 Windows に限らない OS，例えば Android や iOS，OS X や Linux の利用が増加している．この背景に伴い，Windows 以

外の OS を狙ったマルウェアも増加している．企業内においても Windows 以外の OS の利用が増加しているため，サンドボックスを利用した動的解析によるセキュリティ対策の必要性は今後高まる一方であると予想される．

そこで本稿では，Windows を狙ったマルウェアのサンドボックス内の挙動からマルウェア検出に有効と考えられる特徴量を抽出，その有効性を示した後に，同様の特徴量が OS X を狙ったマルウェアの検出にも利用可能か，その実験を行った結果および考察を示す．

2 サンドボックスによる動的解析

Windows を狙うマルウェアでは多くの場合，Windows API コールが利用される．そのためサンドボックス利用によるマルウェアの検出では，呼出される API コールは注目すべき挙動となる．

2.1 サンドボックスの利用における課題

API コールは，正常ファイルおよびマルウェアの双方において利用される．そのため API コールの挙動をもとにした検知ではマルウェアの検出が可能となる一方で，その検知が過検知であるといった問題を引き起こし得る．特に企業内利用における過検知は対応稼動等による業務支障をきたすため，正しくマルウェアのみを検出することが重要となる．

2.2 API コールを利用した悪性判定の関連研究

従来より正常ファイルとの API コール の挙動の逸脱性およびマルウェアとの挙動の類似性に着目し，マルウェアを検出しようとする研究がある．

市田らは過検知を起こしたファイルの分析と合わせ，API コールの Sleep 関数と FileDelete 関数に着目している．Sleep 関数については 2PID 以上での呼出の有無が検出精度の向上に有効であることが示されている [3]．

青木らは API コールの N-gram を特徴量に，正常ファイルとマルウェアを学習データとした決定木による悪性判定を行っている．そこでは，API コールの連鎖の数が検出精度の向上に必ずしも有効ではないことが示されている [5]．

3 非 Windows 系マルウェア

近年 Windows に限らない OS，Android や iOS，OS X や Linux 等の利用が増加している．この背景に伴い，Windows 以外の OS を狙ったマルウェアも増加している．

3.1 OS X を狙ったマルウェア

Mac の OS X は高い安全性を内臓していると言われている [1]．しかしながら近年では，Mac の利用者数の増加に伴い OS X を狙ったマルウェアが増加している．代表的なマルウェアとして，例えば以下のマルウェアが挙げられる [2]．

1. XLSCmd

   キーロガーやリバースシェルを行うトロイの木馬．

2. WireLurker

   侵入先のコンピュータを介して，接続された iOS デバイスに拡散し情報を盗み取るトロイの木馬．

3. iWorm

   バックドア仕掛けるトロイの木馬．

3.2 非 Windows を狙った未知のマルウェアへの対策

Windows 以外の OS の利用が増加している今，非 Windows の OS における未知のマルウェア対策の必要性は今後高まる一方であると予想される．そこで本稿では，非 Windows の OS として OS X を取り上げ，Windows におけるマルウェアの動的解析の結果から得られた悪性挙動の知見を OS X における動的解析の類似挙動への利用を検討する．

4　マルウェア検出のための特徴量

本節では，マルウェアと正常系ファイルとの挙動をもとにマルウェア検出に有効であると思われる特徴量について述べる．尚，本稿において正常系ファイルとは，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と雑誌等で紹介されているフリーソフトを示すとする．

4.1　Sleep 挙動

マルウェアは自身が解析されることを回避するために Sleep 関数を呼出し，自動解析のタイムアウトを狙うことがある．他方，正常系ファイルのインストーラー等ではユーザーインタラクションを求めるために，待機状態にて Sleep 関数を呼出し続けることがある．市田らによると 2PID 以上での Sleep 関数の呼出の有無がマルウェア検出に有効であることが示されている [3]．

4.2　自身の自動起動設定

マルウェアは，コンピュータの再起動後にも活動を継続するために自身の自動起動設定を行うことがある．他方，様々な正常系ファイルにおいても自動起動設定が行われるが，多くの場合ユーザーの許可をとった後にその設定が行われる．そのためサンドボックスというユーザーインタラクションがない環境においては，正常系のファイルにおいて自身の自動起動設定がされるものは少ないと考えられる．

4.3　外部通信

マルウェアは，侵入先のコンピュータより外部へ情報流出させるために外部通信を行うことがある．他方，正常系ファイルのインストーラー等では配信元サイトへのコールバック通信を行うことがある．またマルウェアか否かはグレーであるが，アドウェアについても広告コンテンツへのサイトにアクセスすることがある．

4.4　ポート開放

バックドアといったマルウェアは，侵入先のコンピュータに外部よりアクセスするために管理者やユーザーが空けていないポート番号を開けることがある．他方，正常系ファイルのビデオメッセンジャー等のアプリケーションにおいて外部からの接続要求に応えるためにポートを開放することがある．

5　特徴量の評価実験

本節では，前節にて示した特徴量がマルウェアと正常系ファイルとの識別に有効であるかについて Windows, OS X それぞれについて評価を行う．

5.1　実験データ

サンドボックスにおける動的解析の対象となるファイルの挙動ログとして，以下のデータを用いる．

1. Windows7 動作

	種類数
マルウェア	2974 (5974)
正常系ファイル	728

マルウェアとして，独自に取得したマルウェアが 2974 種類と，FFRI Dataset 2015 よりサンドボックス Cuckoo の挙動ログとして得た 3000 種類を実験データとして扱う [6]．括弧外の値は独自取得マルウェアを，括弧内の値は FFRI Dataset 2015 のマルウェアとを合わせた値を表す．

また正常系ファイルは，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と雑誌等で紹介されているフリーソフトを合わせ独自に取得した 728 種類を実験データとして扱う．

2. OS X 動作

マルウェアとして，独自に取得した 10 種類を実験データとして扱う．

	種類数
マルウェア	10

5.2 評価方法

各実験データのファイルの挙動に対して，前節にて述べた特徴量が検知されるか否かを調べた．さらに本稿では，マルウェアと正常系ファイルとを正しく識別することに目的があるため，特徴量の有効性の評価指標として検出率と過検知率をそれぞれ算出した．検出率は，特徴量の検知に基づいてマルウェアを検出できた割合を表す．また過検知率は，正常系ファイルにおいて特徴量が検知されてしまった割合を表す．サンドボックス利用においては過検知が大きな課題となっているため，過検知率を小さく抑えらながら高い検出率が得られた特徴量が有効な特徴量と言える．

表 1: Sleep 挙動の実験結果

Windows7 動作

	マルウェア	正常系ファイル
特徴量検知有	1268 (1382)	10
特徴量検知無	1706 (4592)	718

検出率 [%]	42.6 (23.1)
過検知率 [%]	1.3

OS X 動作

	マルウェア
特徴量検知有	1
特徴量検知無	9

5.3 実験結果

実験データに対して，特徴量を適用した結果を示す．尚，表のマルウェアの項目の括弧外の値は独自取得マルウェアに対する検知結果に限った値を，括弧内の値は FFRI Dataset 2015 における検知結果を加えた値を表している．

5.3.1 Sleep 挙動

実験結果を表 1 に示す．

過検知率が 1%代に抑えられており，有効な特徴量の一つであることがわかる．本特徴量により検出された FFRI Dataset 2015 のマルウェアの Kaspersky 社による種別としては，Trojan-Ransom が全体の検知の 17%を占めた．OS X についてはマルウェア WireLurker において，Sleep 挙動が検知された．

5.3.2 自身の自動起動設定

実験結果を表 2 に示す．

表 2: 自身の自動起動設定の実験結果

Windows7 動作

	マルウェア	正常系ファイル
特徴量検知有	165 (170)	1
特徴量検知無	2809 (5804)	727

検出率 [%]	5.5 (2.8)
過検知率 [%]	0.1

OS X 動作

	マルウェア
特徴量検知有	3
特徴量検知無	7

過検知率が 1%未満に抑えられており，有効な特徴量の一つであることがわかる．OS X 狙ったマルウェアでは，iWorm, XLSCmd において，自身の自動起動設定の挙動が検知された．

5.3.3 外部通信

実験結果を表 3 に示す．

過検知率が 5%近くやや高いものの，検出率は他と比較して高く有効な特徴量の一つであることがわかる．OS X を狙ったマルウェアでは，WireLurker, iWorm, XLSCmd において，外部通信の挙動が検知された．

表 3: 外部通信の実験結果

Windows7 動作

	マルウェア	正常系ファイル
特徴量検知有	1893 (4892)	32
特徴量検知無	1081 (1082)	696

検出率 [%]	63.6 (81.8)
過検知率 [%]	4.3

OS X 動作

	マルウェア
特徴量検知有	4
特徴量検知無	6

5.3.4 ポート開放

実験結果を表 4 に示す.

表 4: ポート開放の実験結果

Windows7 動作

	マルウェア	正常系ファイル
特徴量検知有	629 (641)	2
特徴量検知無	2345 (5333)	726

検出率 [%]	21.1 (10.7)
過検知率 [%]	0.2

OS X 動作

	マルウェア
特徴量検知有	2
特徴量検知無	8

過検知率が 1%未満に抑えられており，有効な特徴量の一つであることがわかる．OS X を狙ったマルウェアでは，XLSCmd, iWorm において，ポート開放の挙動が検知された.

5.4 考察

4 つの特徴量それぞれがマルウェアの検出に有効であることが確かめられた．一方で，単一の特徴量による検知では過検知となってしまうことも結果よりわかる．過検知の要因としては，3 節でも述べたように特徴量の挙動がマルウェアだけでなく正常ファイルにおいても検知する可能性があることが挙げられる.

過検知を減らす方法としては，複数の特徴量をもとに総合的に判断する方法がある．例えば各特徴量に対してスコアを与え，スコアの合計が設定した閾値を超えた場合に悪性と判定するといった方法が挙げられる．この場合の課題としてスコアリング方法があり，解決策としてはサンドボックスの運用分析官によるスコアリング，もしくは機械学習の応用が考えられる．後者の場合の問題点としては，スコアの意味づけが薄弱となってしまうことが挙げられる．そのため，いずれにせよ運用分析官による調整は避けることはできないと思われる.

また，OS X においてもマルウェアの挙動として Windows のマルウェア同様の挙動が見られた要因として，Windows のマルウェアを参考に OS X のマルウェアが作成されていることがあると考えられる．例えば，OS X を狙った XLSCmd はレポートによると Windows に元来よりあったマルウェアの派生系であることが示唆されている [8]．そのため，現状 OS X においてマルウェアの数は多くないが，Windows のマルウェアの挙動を Mac の挙動とを照らし合わせることでデータ数の少なさを補い，検出精度を向上させることは十分に可能であると思われる.

6 まとめと今後の課題

本稿では，Windows を狙ったマルウェアの挙動にした 4 つの特徴量，Sleep 挙動，自動起動設定，外部通信，ポート開放が Windows におけるマルウェア検出で有効なこと，さらにそれら特徴量が OS X を狙ったマルウェアにおいても同様に検知可能であることを示した．今後の課題としては，OS X のマルウェアや正常系ファ

イルのデータ数を増やした更なる解析，および Android や iOS 等の OS に対しても同様の有効性の有無の検証が挙げられる．さらには，複数の特徴量を合わせた過検知を抑えたマルウェア検出の方法の模索も課題として挙げられる．

参考文献

[1] Apple Inc., http://www.apple.com/jp/osx/what-is/security/

[2] Synack; @patrickwardle, ”Writing Bad @$$ Malware for OS X”, Black Hat 2015.

[3] 市田達也, 須藤年章, 高森覚, ”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”, MWS2014.

[4] 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 コールの分析”, MWS2013.

[5] 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.

[6] 株式会社 FFRI, ”FFRI Dataset 2015”, MWS2015.

[7] 株式会社 FFRI, ”静的情報に基づいたマルウェア判定指標の検討”, Monthly Research, 2014-02-04

[8] Graham Cluley, ”Spyware Gang Ports XLSCmd Malware to Mac OS X from Windows”, http://www.intego.com/mac-security-blog/spyware-xslcmd-malware-os-x/

[9] 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, 2010-12.