リンクされるシンボル名を用いたLinuxマルウェア検知手法に関する考察

イボット アリジャン† 大 山 恵 弘†

1. はじめに

近年，IoTデバイスの普及などによりLinuxを狙ったマルウェアが増加している．増加を続けるLinuxマルウェアに対応するため，我々の研究[1]ではリンクされるシンボル名を用いたLinuxマルウェアの検知手法(以降，単に提案手法と呼ぶ)を提案した．提案手法は10種類以上のアーキテクチャを含むLinuxマルウェアに対して94%の精度で良性ソフトウェアとの分別に成功したが，特徴量として全てのシンボル名を使用した場合と，関数を示すシンボル名のみを使用した場合の間に大きな差が生じた．

本稿では両者の間に生じた差について考察を行い，提案手法がどのような状況においてマルウェアの検出に有効であるかを解明する．

2. 予備知識

2.1 Fuzzy Hash

提案手法では効率的な学習を行うため，シンボル名に対してFuzzy Hashを使用する．Fuzzy HashはContext Triggered Piecewise Hashとも呼ばれ，2006年にJesse Kornblumによって提唱[2]されたハッシュ関数である．Fuzzy HashはMD5などの暗号学的ハッシュ関数とは異なり，似たような入力に対して似たようなハッシュ値を出力する．そのためFuzzy Hashによって出力されたハッシュ値間の類似度を比較することで，Fuzzy Hashの入力に使われたデータの類似度を推定することが出来る．

2.2 ランダムフォレスト

提案手法では機械学習アルゴリズムとしてランダムフォレストを使用する．ランダムフォレストは決定木を利用する機械学習アルゴリズムである．乱数を利用することで少しずつ異なる決定木を複数集め，その平均を取ることで過剰適合を減らす事ができるアンサンブル学習法の一種でもある．

3. 提案手法の概要

提案手法（**図1**）ではまず訓練データとして良性・悪性が既知のELFファイルから，リンクされるシンボル名の抽出を行う．次に抽出したシンボル名をFuzzy Hashでハッシュ化し，ランダムフォレストを用いてマルウェア検知器の訓練を行う．その後，未知のファイルに対しても同様にシンボル名のFuzzy Hash値を取得することにより，訓練済のマルウェア検知器で良性・悪性の分類を行うことができる．

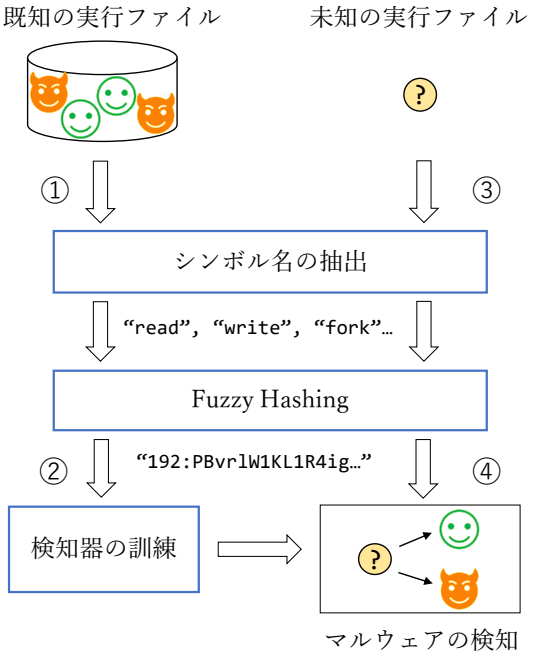

図1 提案手法の概要

4. 課題

我々は提案手法を評価するためにシンボル名の取得方法を複数用意し，比較実験を行った．実験の結果，**表1**で示されたように関数を示すシンボルのみを使用した場合は，全てのシンボルを使用した場合と比べてprecisionが上昇し，recallが大きく低下していた．しかし我々がこの実験結果を公表した際，両者の間で生じた差については詳細な調査・考察に至らなかった．

本稿では両者の間に生じた差について考察を行う．

† 筑波大学
University of Tsukuba

表 1　提案手法の検知精度

入力	accuracy	precision	recall	F-measure
全シンボル	0.941	0.952	0.928	0.940
関数シンボル	0.919	0.985	0.851	0.913

表 2　良性・悪性ファイル間におけるシンボル名の類似度

	全シンボル	関数シンボル	変化率
悪性 × 悪性	5.172	5.219	100.1%
良性 × 良性	3.771	2.093	55.5%
悪性 × 良性	0.047	0.090	191.5%

5. 考　　察

5.1 シンボル名の類似度

良性・悪性ファイル間におけるシンボル名の類似度を**表 2** に記載する．これはファイルを良性・悪性に分別し，各ファイルのシンボル名の Fuzzy Hash 値からシンボル名の類似度について測定したものである．

マルウェア間における類似度は全シンボル・関数シンボルを使用した場合の間に大きな差は見られなかった．マルウェアには亜種が存在し，同じ関数群を使用するファイルが複数存在するためと考えられる．一方で良性ソフトウェア間における類似度は，関数シンボルを使用した場合と全シンボルを使用した場合で大きく異なった．良性ソフトウェアには stdin などの共通した関数以外のシンボルが複数存在する一方，同じような動作をする亜種が少ないため，関数シンボルのみを使用させることにより類似度が減少したと考えられる．

5.2 決定木アルゴリズム

決定木アルゴリズムは各ノードにおいて，次のノード内に存在するクラスの純度が最も高くなるように生成されてゆく．次のノードの純度が変わらなくなった場合，現在のノード内で最多なクラスのリーフとなる．

よって全シンボルを使用した場合に生成される決定木（**図 2**）は良性・悪性ともに類似度が高いため，分類に失敗する総数は少ないものの，良性・悪性ともに分類ミスが発生したと考えられる．一方で関数シンボルのみを使用した場合に生成される決定木（**図 3**）は，全シンボルを使用した場合より良性ソフトウェアの類似度が下がるため，分類に失敗する総数が増えたと思われる．ただし分類を諦めてリーフとなったノードには良性ソフトウェアが多く残るため，良性ソフトウェアの分類ミスは減少したと推測することができる．

6. お わ り に

本稿ではリンクされるシンボル名を用いた Linux マルウェアの検知手法において，全てのシンボルを使用した場合と関数シンボルのみを使用した場合の間に生じた差について考察を行った．その結果，良性ソフトウェアとは異なりマルウェアには同じ関数群を使用する亜種が多く存在することが原因だと推測された．よって，この手法は亜種が多いマルウェアの検出に有効である一方，新種などの亜種が少ないマルウェアに弱いと考えられる．亜種が少ないマルウェアを検出できる手法については今後の研究課題としたい．

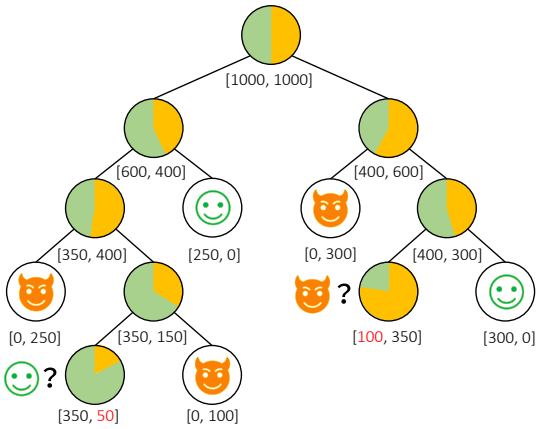

図 2　全シンボルを使用した場合に生成される決定木のイメージ

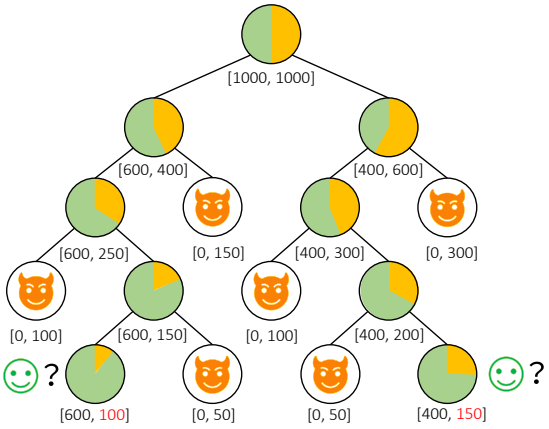

図 3　関数シンボルを使用した場合に生成される決定木のイメージ

謝辞　本研究の一部は JSPS 科研費 17K00179 の助成を受けたものである．

参　考　文　献

1) イボット アリジャン, 大山 恵弘.: 動的シンボル情報を用いた Linux マルウェアの検出, コンピュータセキュリティシンポジウム 2019 論文集, pp. 1329-1335
2) Jesse Kornblum.: Identifying almost identical files using context triggered piecewise hashing, Digital Investigation: The International Journal of Digital Forensics & Incident Response Volume 3 (2006), pp. 91-97