6X-06

APIコール列とLZWを用いたマルウェア亜種分類の一検討

浅沼 和希 † 甲斐 博 ‡ 森井 昌克 §
愛媛大学 † 愛媛大学 ‡ 神戸大学 §

1. まえがき

MacAfee Labs の 2020 年 11 月の脅威レポート [1] によると，発見されるマルウェアのほとんどは既存のマルウェアの亜種である．このため発見されるマルウェアがどのマルウェアファミリに属するかを高精度で分類することはセキュリティ対策を行う上で重要である．

マルウェアファミリの分類を行う手法として，Paragraph Vector を用いる方法が先行研究 [2] で提案されている．先行研究では，マルウェアの挙動を監視して得られた動的解析結果のうち，API コールを使用する．Paragraph Vector によって，API コール列から特徴ベクトルを作成し，これに教師データを付加したものを SVM およびランダムフォレストで学習させ，それぞれのマルウェアファミリについての分類器を作成する．これに対してテストデータを入力し，マルウェアの亜種判定を行う．

本研究では API コールの繰り返しに着目した特徴を用いることを考える．具体的には，辞書式圧縮アルゴリズム LZW の辞書を特徴として用いる．API コール列と辞書を使って特徴ベクトルを作成し，マルウェア亜種分類を行う手法について検討する．

2. 提案手法

本研究で提案するマルウェアの亜種分類手法を示す．

まず，アルゴリズム 1 に示される LZW アルゴリズム [3] を用いて検体の API コール列を圧縮し，作成される辞書 string table を得る．

次に，API コール列に，得られた辞書を付加したものを文書として Paragraph Vector を作成する．Paragraph Vector のモデルは，図 1 に示される PV-DBOW(Distributed Bag of Words Model of Paragraph Vector) モデルを用いる．このモデルでは，Paragraph ID に対応するベクトルが入力され，文中の語 $(w_{t-k}, ..., w_{t+k})$ の出現確率が最も高くなるように Deep Learning によって学習される [4].

最後に，得られた Paragraph Vector にラベルを付け，SVM で教師付き学習を行い学習器を作成する．作成した学習器を用いてマルウェアの亜種を分類する．

アルゴリズム 1 LZW アルゴリズム

```
Initialize table to contain single-character strings.
Read first input character → prefix string ω
Step: Read next input character K
    if no such K (input exhausted): code(ω)→output; EXIT
    if ωK exists in string table: ωK → ω; repeat Step.
    else ωK not in string table: code(ω)→output;
                                 ωK →string table;
                                 K → ω; repeat Step;
```

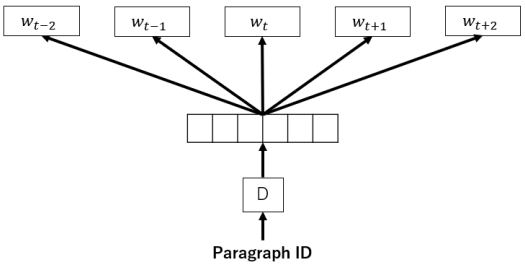

図 1 PV-DBOW モデルの構造

3. 提案手法による分類の実験

3.1 使用するデータセット

本研究では，FFRI 社が収集したマルウェアの動的解析結果である FFRI Dataset2013-2016[5] を用いる．Kaspersky の命名規則に則り，API コールが取得されている検体のうち，亜種の数が 100 検体以上ある 22 ファミリを対象とした．

3.2 実験方法

前節で対象となった検体に対して，動的解析の結果から API 名を抽出し，検体ごとに API コール列を作成する．作成した API コール列に LZW 辞書を付加したものを Paragraph Vector によってベクトル化する．学習器はそれぞれのマルウェアファミリに対し亜種であるかないかを判定させる二値分類を行う．そのため，亜種判定したいファミリとそれ以外の 21 ファミリの検体の比率が 21:1 になるように Paragraph Vector を抽出し，亜種判定したいファミリの学習器用データセットを作成する．これにより，ファミリ毎に 22 個のデータセットが作成される．

それぞれのデータセット内の検体のベクトルに，教師データとして亜種であるかないかのラベルを付け，SVM で学習させる．学習が終わった学習器に対してテストデータを入力し，亜種であるかないかの二値分類を行うことで分類精度を測定

A consideration of malware classification by API call sequences and the LZW algorithm
† Kazuki Asanuma, Ehime University
‡ Hiroshi Kai, Ehime University
§ Masakatu Morii, Kobe University

する．本研究では，4 分割交差検証により分類精度を測定した．

3.3 実験結果

実験の結果として，それぞれのファミリの分類精度および全体の分類精度の平均値を表 1 に示す．先行研究の分類精度の平均値 81.94% よりも高い分類精度を確認できた．

表 1　SVM による分類精度

ファミリ名	分類精度 [%]
Trojan.Win32.Waldek	93.68
Trojan.Win32.Agent	73.30
Trojan-Ransom.Win32.Foreign	83.08
Trojan-Dropper.Win32.Injector	83.33
Trojan.Win32.Yakes	79.16
Trojan.Win32.Llac	81.55
Backdoor.Win32.Matsnu	85.76
Trojan-PSW.Win32.Tepfer	76.06
Trojan.Win32.Jorik	91.07
Backdoor.Win32.Androm	75.91
Worm.Win32.WBNA	90.92
Trojan-PSW.Win32.Fareit	80.56
Trojan-Downloader.Win32.Upatre	80.01
Backdoor.Win32.DarkKomet	82.18
Trojan.Win32.Scar	72.49
Packed.Win32.Tpyn	84.93
Trojan-Spy.Win32.Zbot	78.16
Worm.Win32.Vobfus	91.06
Hoax.Win32.ArchSMS	95.21
Trojan.Win32.Kovter	90.66
Downloader.Win32.LMN	86.25
Trojan.Win32.Inject	88.24
平均値	83.80

3.4 考察

マルウェアファミリ Trojan-Downloader.Win32.Upatre の亜種 Trojan-Downloader.Win32.Upatre.fopg について API コール列の一部を図 2 に示す．また，異なるマルウェアファミリ Trojan.Win32.Yakes の亜種 Trojan.Win32.Yakes.olac についても同様に API コール列を図 3 に示す．ここで API コール列に現れる API 名は，紙面の関係上，整数値で表現している．例えば NtAllocateVirtualMemory は 0，NtFreeVirtualMemory は 1 として表現している．図 2 と図 3 を見ると，API コール 2 や 10 が多く繰り返される点において Trojan-Downloader.Win32.Upatre.fopg と Trojan.Win32.Yakes.olac は類似している．実際に，API コール列を Paragraph Vector で特徴ベクトル化し，機械学習で分類した場合，どちらのマルウェアもマルウェアファミリ Trojan-Downloader.Win32.Upatre に分類される．

これに対して，API コール列 +LZW 辞書について，マルウェア Trojan-Downloader.Win32.Upatre.fopg のものを図 4，マルウェア Trojan.Win32.Yakes.olac のものを図 5 に示す．追加した LZW 辞書情報では，マルウェア Trojan.Win32.Yakes.olac でのみ繰り返し呼ばれている API コール 23 が多く現れるため，類似性を下げることができた．実際，機械学習で分類した場合，Trojan.Win32.Yakes.olac はマルウェアファミリ Trojan-Downloader.Win32.Upatre ではないと分類された．このように繰り返しの情報を付加することで，異なるファミリの検体間での類似性を減少させることが，分類精度向上の一因として考えられる．

0 1 0 2 2 2 2 2 2 2 2 2 2
0 1 3 4 5 6 7 8 9 10 10
11 5 5 12 13 14

図 2　Trojan-Downloader.Win32.Upatre.fopg の API コール列

0 1 0 0 ... 10 10 2 2 2 2
2 17 ... 23 23 23 23 10
23 2 2 2 2 2 17 10 23 23

図 3　Trojan.Win32.Yakes.olac の API コール列

0 1 0 2 ... 0 2 2 2 2 2 2
2 ... 10 10 10 10 11 11
5 5 5 5 12 12 13 13 14

図 4　Trojan-Downloader.Win32.Upatre.fopg の API コール列 +LZW 辞書

0 1 0 0 ... 23 23 23 23
23 23 23 10 10 23 23 2
2 2 2 2 2 17 17 10 23

図 5　Trojan.Win32.Yakes.olac の API コール列 +LZW 辞書

4. まとめ

本研究では，API コール列に LZW 辞書情報を付加した Paragraph Vector を特徴量としてマルウェアの亜種を分類する手法を提案した．実験の結果 LZW 辞書を付加しない分類手法よりも高い分類精度が得られることを示した．

精度向上に関してより詳細な解析を行うことが今後の課題である．

参考文献

[1] MacAfee：MacAfee 脅威レポート，https://www.mcafee.com/enterprise/ja-jp/assets/reports/rp-quarterly-threats-nov-2020.pdf (2020)．

[2] 佐藤 拓未，後藤 滋樹，武部 嵩礼：Paragraph Vector を用いたマルウェアの亜種推定法，情報処理学会，コンピュータセキュリティシンポジウム 2016，pp.298-304 (2016)．

[3] Welch, T.A.: A Technique for High-Performance Data Compression, IEEE Computer, Vol.17, No.6, pp.8-19 (1984).

[4] Le, Q.V. and Mikolov, T.: Distributed Representations of sentences and Documents, Proceedings of the 31st International Conference on Machine Learning, pp.1188-1196 (2014).

[5] 寺田真敏，秋山満昭，松木隆宏ほか：マルウェア対策のための研究用データセット MWS Datasets 〜コミュニティへの貢献とその課題〜，情報処理学会，Vol.2020-IFAT-139 No.8 (2020)．