自動表層解析システムにおける FFRI Dataset scriptsの利用と拡張

荒井 晃平[1,a)]　小林 良太郎[1]

概要：近年，マルウェアの種類や攻撃は増加傾向にあり，これらのマルウェアへの対策として，解析により作成されたシグネチャを用いた製品が存在する．このシグネチャの作成において，他の解析手法と比較して環境の構築が容易である表層解析が用いられることがあるが，現状として手動での解析では限界である．表層解析により得られる情報は，マルウェア分類に関する研究に多く用いられており，有益な情報である．一方で，解析の経験や知識が少ない人が，表層情報から攻撃手法の概要を把握することは容易ではない．そのため，FFRI セキュリティが提供するスクリプトを拡張し，複数のツールと組み合わせることで，表層情報の取得と明瞭化を実現する．本論文では，FFRI Dataset scripts を利用し，マルウェアの詳細な表層情報の取得と明瞭化に対応した自動表層解析システムを提案する．

キーワード：表層解析, マルウェア, 自動解析

Usage and Extension of FFRI Dataset Script in Automated Surface Analysis System

Kohei Arai[1,a)]　Ryotaro Kobayashi[1]

Abstract: In recent years, the number of malware types and attacks has been increasing, and there are products that use signatures created by analysis as a countermeasure against these malware. In creating these signatures, surface analysis is sometimes used because it is easier to construct an environment than other analysis methods. While the information obtained from surface analysis is used in many malware classification studies and is useful, it is not easy for those with limited analysis experience and knowledge to obtain an overview of the attack methods from the surface information. Therefore, we extend the scripts provided by FFRI Security and combine them with several tools to obtain and clarify the surface information. In this paper, we propose an automatic surface analysis system using FFRI Dataset scripts for obtaining and clarifying detailed surface information of malware.

Keywords: Surface Analysis, Malware, Automated Analysis

1. はじめに

近年，マルウェアの種類や攻撃は増加傾向にあり，対策としてマルウェア用のセキュリティソフトウェアが一般的に用いられている．これらのマルウェア対策用ソフトウェアの多くは，既存のシグネチャとのパターンマッチングによりマルウェアを検知するため，既知のマルウェアにのみ対応可能である．未知のマルウェアに対しては，ホワイトリストによる制御や，機械学習や AI といった次世代型製品での検知により対応可能である．これらの検知機構に必要なシグネチャの作成において，表層解析が用いられることがある．表層解析とは，マルウェアを実行することなくファイルの情報を取得する手法であり，ファイルのメタデータやハッシュ値，含まれる文字列といった表層情報の取得が可能である．先行研究として，コンピュータセキュ

[1] 工学院大学
Kogakuin University, Shinjuku, Tokyo 163-8677, Japan
[a)] em24003@ns.kogakuin.ac.jp

リティシンポジウム 2023 で発表したシステム [1] では，複数の UNIX コマンドを用いることで，信頼性の高い表層情報の取得を自動化した．しかし，難読化による影響や PE ファイルに対する詳細な解析については未対応である．そこで，表層情報のデータセットとして MWS Dataset に含まれる，FFRI Dataset を作成するスクリプト [2] を使用することで，既存の表層解析システムの発展を行う．FFRI セキュリティが提供するスクリプトでは，PE 形式のファイルに対する詳細な解析が可能であるが，難読化による影響を考慮していないため，他のツールで難読化の解除を行う必要がある．また，2023 年以降に提供されているスクリプトでは，PE 形式ではないファイルに対する解析が可能であるため，FFRI セキュリティが提供する 2023 年版以降のスクリプトを使用するが，解析対象ファイルの指定が必要であり，半自動のシステムである．そこで本論文では，FFRI Datset scripts を拡張し，複数のツールと組み合わせることで，マルウェアのより詳細な表層情報の取得に対応した表層解析の自動化を実現する．また, 本論文の検証に使用したマルウェア検体の入手先は，YNU IoT honeypot Dataset-1 等である [3],[4].

本論文の構成は以下の通りである．まず第 2 章で関連研究について述べる．第 3 章で提案システムについて述べ，第 4 章で提案システムでの検証内容及びその結果について述べる．それをもとに第 5 章で結論及び今後の課題について述べる．

2. 関連研究

本論文では，マルウェアに対する自動表層解析システムを提案する．本研究では，FFRI Dataset script と複数の解析ツールを用いて表層情報を自動で取得するが，本章では，自動表層解析システムの先行研究について述べ，得られる表層情報をもとにどのような既存研究が存在するのかについて述べる．最後に本研究の位置付けについて述べる．

2.1 自動表層解析システムに関する先行研究

先行研究では，UNIX コマンドを用いた表層解析を自動化することで，自動表層解析システムを実現した [1]．解析環境には仮想環境として VirtualBox を使用し，解析対象の検体を区別せず UNIX コマンドを適用することで，解析結果を出力した．それにより，仮想環境によるオーバーヘッドが大きいことや，PE 形式のファイルへの詳細な情報の取得，難読化による影響が課題であった．

2.2 表層情報を用いた研究

岡山らは，マルウェアが引き起こす将来的な被害やそれらへの対策をまとめたものを最終進行度と定義し，短い動的解析ログと表層情報を組み合わせることで，最終進行度の推定を行った [5]．動的解析ログ，表層情報から抽出したものそれぞれから単語を抽出し，特徴ベクトルに変換し分類することで最終進行度ラベルを作成した．その結果，86〜87%という精度を得た．さらに作成した分類器に対して回帰を用いることで，精度が最大で 2.5%向上した．また，各検体の動的解析ログを全て用いて分類した場合と同程度の精度を得るのに必要なログの行数を，97.8%削減した．ここでは，MWS Dataset の一部として提供される Soliton Dataset 2020 を使用した．

2.3 FFRI Dataset を用いた研究

茂木は，fuzzy hash や peHash といったハッシュ値と PE 表層情報を組み合わせた分類器を作成した [6]．これらのハッシュ値と PE 表層情報を組み合わせることで，PE 表層情報単体で作成した分類器の 97〜98%という精度から約 1 ポイント向上した．実験結果から，マルウェア検知における表層情報の有用性，表層情報とハッシュ値のそれぞれを用いた分類器を組み合わせることで，それぞれ単体での性能より向上することが明らかとなった．Mimura は，検体に含まれる文字列を用いて，検体の分類器を作成した [7]．また，悪性と良性の検体それぞれで時系列を考慮した実験を行い，検体に含まれる文字列が PE ファイルの検出に効果的であった．最終的に，複数の言語モデルを組み合わせて作成した分類器において，0.981 という精度を得た．

2.4 本研究の位置付け

本研究は，マルウェアの表層情報を自動で取得することを目的として，FFRI Dataset scripts を利用する．

2.1 節で述べた先行研究 [1] では，表層解析システムの課題点，得られる表層情報の課題点がそれぞれ明らかとなった．2.2 節で述べた既存研究 [5] は，マルウェアの表層情報を用いたマルウェアの分類に関する研究であり，マルウェアの分類器における表層情報の有用性を示している．また，2.3 節で述べた既存研究 [6],[7] は，FFRI Dataset を用いたマルウェアの分類に関する研究であり，マルウェアの分類器における FFRI Dataset の有用性を示している．これらのことから，FFRI Dataset-scripts を利用した表層解析システムは，マルウェアの分類に対して有用であり，本システムを自動化することで，日々増加するマルウェアへの対応を可能にする．

3. 提案システム

本論文では，FFRI Dataset-scripts を利用した自動表層解析システムを提案する．FFRI Dataset-scripts で得られる表層情報は，FFRI Dataset の内容と同等である [8]．本システムでは，検体ファイルに対するスクリプトでの解析実行から，解析結果の取得までを自動で行う．FFRI Dataset-scripts では，PE 形式のファイルに対する詳細な解析が可能であり，PE 形式特有のハッシュ値や PE 形式

にのみ対応するツールでのパッカー情報がそれぞれ得られる．スクリプトでは，バイナリに含まれる文字列の取得に strings コマンドを使用しているが，オプションを指定せずに文字列の情報を出力している．PE 形式のファイルは，Unicode が含まれることが多く，オプションの指定が必要である．しかし，検体が難読化されている場合は，strings コマンドによる解析において重要な情報を得にくいといった課題が存在する．また，スクリプトはマルウェアの分類へ用いることを前提としているため，検体の解析結果の明瞭さという観点から見ると機能の追加が必要である．これらのことから本システムでは，strings コマンドにオプションを指定した解析の追加，解析結果の明瞭化を目的とした Stella[9]，Capa[10] の追加，難読化への対応を目的とした代表的なパッカーの解除をそれぞれ適用する．しかし，Capa による解析は処理時間が膨大であるため，各解析の実行可否を選択可能にすることで対応する．本システムは，解析対象となる検体 zip ファイルを実行指示マシンに配置し，解析開始のスクリプトを実行することで，表層解析結果が出力されるという流れである．表層解析は主に，解析準備，FFRI Dataset-scripts による解析，テキストファイルに対する解析，難読化への対応，Stella による解析，Capa による解析という 6 段階に分けられ，各解析環境の構築には仮想環境として Docker[11] を使用している．本システムの概要図を**図 1** に示す．

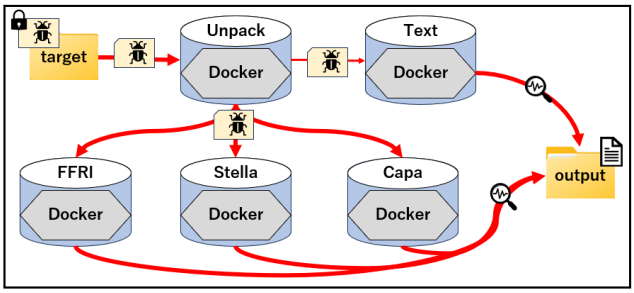

図 1　システム概要図
Fig. 1　System Overview Chart

3.1　解析準備

一般的に，検体は zip ファイルで提供されているため，解凍して扱う必要がある．また，FFRI Dataset-scripts では，解析対象ファイルのパス，ラベル，日付をそれぞれ csv ファイルに記述する必要がある．そのため，解析スクリプトを実行すると最初に検体 zip ファイルを解凍し，ファイルのパスなどの必要事項を csv ファイルに出力する．この際，テキストファイルが含まれる場合は，別のフォルダに移動させ，区別する．

3.2　FFRI Dataset-scripts による解析

解析準備が完了すると，csv ファイルに記載されているファイルを対象として，FFRI Dataset-scripts による解析を実行する．本システムで対象とする検体は，インターネット上に存在する全てのマルウェアとするため，PE 形式のファイル以外も対象とし，スキーマのバージョンは 2024 とする．解析には仮想環境として Docker を使用しており，解析結果のファイル形式は json である．strings コマンドによる解析では，オプションの指定を独自に追加することで，PE 形式に含まれる Unicode に対応する．バイナリデータ以外への解析を実行すると，エラーが出る場合があるため，テキストファイルに対する解析は別のスクリプトで実行する．また，PE 形式のファイル以外には Lief による解析が適用されず，検体 zip 内でのファイル名が解析結果に出力されないため，ファイル名の出力を追加することで対応する．解析の追加として Magika を実装する [12]，Magika は，Google 社が提供する AI モデルを利用したファイル形式識別ツールで，他の識別ツールよりもスコアが高いことが示されている [13]．

3.3　テキストファイルに対する解析

FFRI Dataset-scripts による解析が完了すると，解析準備で分けたテキストファイルに対する解析を実行する．解析内容は，ハッシュ値 (MD5, SHA-256, SHA-1, ssdeep) の算出，文字列の抽出 (strings)，ファイル情報の取得 (file, die, stat) である．解析には仮想環境として Docker を使用しており，解析結果のファイル形式は json である．

3.4　難読化への対応

検体に対して難読化が施されている場合，strings コマンドによる文字列の抽出を行う際に影響が出てしまい，正常に文字列を得ることができない．そのため，事前にパッカーを特定し，解除することでパッキング処理に対応する．本システムでは，代表的なパッカーによる解除に対応する [14]．パッカーの解除を行うことでハッシュ値が変化するため，解除後に FFRI Dataset-scripts，Stella，Capa による各解析，テキストファイルに対する解析をそれぞれ行う．

3.5　Stella による解析

これまで行ってきた解析は，マルウェアの分類を目的とした解析結果の出力であるため，解析者が結果を確認した際に得られる情報は不明瞭である．そのため，Stella を使用することで，抽出した文字列から予測される攻撃手法を明瞭化する．Stella は PE 形式のファイル向けではあるが，strings 結果から各関数名を抽出するため，本システムでは PE 形式のファイル以外にも適用する．また，Stella には FLOSS という難読化された文字列に対応した，文字列抽出ツールを使用する解析が存在するが，本システムでは難読化の解除を別のスクリプトで適用するため使用していな

い．解析には仮想環境として Docker を使用しており，解析結果のファイル形式は text である．ここでの解析は解析結果の明瞭化を目的するため，パッカーの検出を行った後に解析を実行する．これにより，解析の経験や知識が少ない人でも，攻撃手法の把握が可能である．

3.6 Capa による解析

Stella と同様に，Capa を使用することで検体情報の可視化が可能である．Capa は，実行形式のファイルに対してケーパビリティの検知を行うツールである．ここでのケーパビリティとは，実行形式のファイルが保持する機能であり，その機能を ATT&CK[15] に基づき出力する．ATT&CK とは，実際の攻撃を技術や手法といった観点で分類したナレッジベースである．Capa による解析で得られる情報は，表層情報の明瞭化を実現すると同時に，機械学習データとしても有用であるため，解析には仮想環境として Docker を使用しており，解析結果のファイル形式は text と json である．これにより，解析の経験や知識が少ない人でも，攻撃手法の把握が可能である．Capa はパッキングされている検体に対しては解析が不可能であるため，パッカーの検出を行った後に解析を実行する．

3.7 研究倫理

ここでは，提案システムにおける安全性への対応について述べる．まず検体とするマルウェアファイルの扱いについて，マルウェアの入手，実行指示マシンへの配置の過程での誤実行防止を目的として，鍵付き zip ファイルとして扱っている．また，解析環境に Docker を使用することで，環境の仮想化を実現している．実行指示マシンおよび解析環境は，インターネットへの接続を行っておらず，ネットワーク設定は本システム内での通信のみとしている．そして解析結果の扱いについて，マルウェアを再現するための十分な情報は記載されていないが，手掛かりとなる情報が記載されている．そのため，第 4 章で述べる検証結果について，論文情報が第三者によって悪用される危険性を考慮し，解析結果の一部のみを記載している．

4. 提案システムの検証

ここでは，前章までで述べてきた本システムに対して，マルウェアを投入して解析を実行することでシステムの有用性を検証する．解析によって得られた表層情報と，解析実行開始から終了までの解析処理時間について述べる．

4.1 システム構築環境

本システムで使用しているマシン及び仮想環境は以下の通りである．

- 実行指示マシン
  - ホスト OS: Ubuntu-22.04-amd64
    - CPU: Intel Core i3-1115G4
    - メモリ: DDR4-3200 32GB
  - 仮想環境: Ubuntu-22.04-amd64

4.2 FFRI Dataset-scripts による解析結果

FFRI Dataset-scripts による解析によって得られた表層情報は, 以下の通りである．追加した解析以外は FFRI Dataset の内容と同等であるため，一部を記載する．

- 検体 A
  - ”file_name”: ”276cdbXXXXXc72a4.exe”
  - ”trid”: {”(.EXE)GenericCILExecutable (.NET,Mono,etc.) (73123/4/13)”: ”71.1%”, }
  - ”magika”: {”ct_label”: ”pebin”, ”score”: ”1.0”, ”group”: ”executable”, }
  - ”strings_plus”: [, ”http://”, ”/connect_bot.php”, ”data=”, ”update”, ”/receive_bot.php”,]
- 検体 B
  - ”file_name”: ”dfd2aXXXXX7491c.elf”
  - ”trid”: {”(.)ELFExecutableandLinkableformat (Linux) (4022/12)”: ”50.1%”, }
  - ”magika”: {”ct_label”: ”elf”, ”score”: ”1.0”, ”group”: ”executable”, }
  - ”strings_plus”: null

4.3 Stella による解析結果

Stella による解析によって得られる情報の一部は以下の通りである．

- 検体 B
  - md5sum : 4aa7bXXXXX74e02
  - sha256sum: dfd2aXXXXX7491c
  - device 関連: POST /ctrlt/DeviceUpgrade_1 HTTP/1.1 Authorization: Digest username=”dslf-config”,
  - このマルウェアに関連する http: POST /ctrlt/DeviceUpgrade_1 HTTP/1.1, POST /cgi-bin/ViewLog.asp HTTP/1.1
- 検体 C
  - sha1sum: 04b5dXXXXX19191
  - xxd: 4d5a 9000 0300 0000 0400 0000 ffff 0000 MZ.....
  - File 関連: SetFileSecurityW, SHFileOperationW, SHGetFileInfoW
  - Blacklist 検索結果:
    shell やコマンドプロンプトが使われる可能性があります。 - ShellExecuteExW,
    ファイルの作成、移動、削除が行われる可能性があります。 - MoveFileW, MoveFileExW

4.4 Capa による解析結果

Capa での検体 D に対する解析によって得られる情報の一部を**図 2** に示す.

MBC Objective	MBC Behavior
FILE SYSTEM	Read File [C0051]
PROCESS	Allocate Thread Local Storage [C0040] Terminate Process [C0018] Terminate Thread [C0039]

図 2 検体 D に対する Capa での解析結果
Fig. 2 Results of Capa Analysis for Sample D

4.5 テキストファイルに対する解析結果

テキストファイルに対する解析によって得られる情報の一部は以下の通りである.

- 検体 E
  - ”sha256”: ”42b9eXXXXXf6869”
  - ”file”: {”type”: ”ASCII text”, ”info”: ”with very long lines (61017)”}
  - ”strings”: [”set ”Systiojbbw=\Wocrgvnlvju.png””, ”set ”Abovsvqpzu=tem32\WindowsPowerShel””]
- 検体 F
  - ”md5”: ”abc26XXXXXea1ad”
  - ”ssdeep”: {”block_size”: ”24576”, ”hash1”: ”l3XOFUJGP50sYdpm/mg97QFfJDHOVa0/H2/3I1EnB”, ”hash2”: ”pOkVCDUVJ4uz”}
  - ”stat”: {”file_type”: ”regular file”, ”file_size”: ”1044136”, ”block_size”: ”512”}

4.6 パッカー解除前後での解析結果の比較

検体のパッカー解除前後では解析結果が異なる．特に，ハッシュ値は別の値となり，可読可能な文字列も異なるため，解析は別の検体として解析を実行することになる．ここでは，パッカーの検知とアンパック処理をログとして出力することで，アンパック前後のファイルを比較可能としている．パッカー解除前後での，FFRI Dataset-scripts と Stella の解析結果の比較は以下の通りである.

- パッカー解除前
  - FFRI Dataset-scripts
    - ”file_name”: ”1efd3XXXXX17011.exe”
    - sha256: ”1efd3XXXXX17011”
    - ”die”: {”detects”: [{”filetype”: ”PE32”, ”parentfilepart”: ”Header”, ”values”: [{”info”: ”NRV,brute”, ”name”: ”UPX”, ”string”: ”Packer: UPX(3.05)[NRV,brute]”, ”type”: ”Packer”, ”version”: ”3.05”},},}
    - ”strings”: [”UPX0”, ”UPX1”, ”H\tfl”, ”dQ0!”, ”v‘_W”,]
    - ”strings_plus”: [”DEFAULT_ICON”, ”(&P)”, ”(&T)”,]
  - Stella
    - sha256sum: ”1efd3XXXXX17011”
    - START 関連: null
    - SET 関連: 7sET
    - local 関連: null
- パッカー解除後
  - FFRI Dataset-scripts
    - ”file_name”: ”1efd3XXXXX17011_upx”
    - sha256: ”b291aXXXXXabe81”
    - ”die”: {”detects”: [{”filetype”: ”PE32”, ”parentfilepart”: ”Header”, ”values”: [{”info”: ”-”, ”name”: ”Microsoft Visual C/C++”, ”string”: ”Compiler: Microsoft Visual C/C++(6.0)[-]”, ”type”: ”Compiler”, ”version”: ”6.0”},},}
    - strings: [”.text”, ”.rdata”, ”GetWindowRect”, ”SetForegroundWindow”,]
    - ”strings_plus”: [”msctls_updown32”, ”msctls_progress32”, ”Progress1”, ”(&O)”,]
  - Stella
    - sha256sum: ”b291aXXXXXabe81”
    - START 関連: GetStartupInfoA, StartDocA, StartPage, midiStreamRestart
    - SET 関連: SetForegroundWindow, SetPropA, Settings, commdlg_SetRGBColor, SetEndOfFile
    - loacl 関連: GetLocalTime, LocalReAlloc
    - Blacklist 検索結果:
      パソコンを再起動したときに自動起動する可能性があります。- GetStartupInfoA
      C2 通信が行われる可能性があります。- InternetReadFile

4.7 解析処理時間

ここでは，検体数を 1，10，100，200 と増加させた際の処理時間を**表 1** に示す．検体数が 1, 10 の際は，パックされた検体やテキストファイルの存在の有無が解析処理時間に大きく影響するため，3 回の計測それぞれで検体を変更し計測する．また，検体数が 200 の場合の，各スクリプトの処理時間を**表 2** に示す．200 検体の内 5 検体がテキストファイルである．ここでの計測区間は，解析開始のスクリプトを実行した時点から，解析スクリプトが終了した時点である．Capa による解析は処理時間が膨大であるため，これらの解析処理に含まれていない.

表 1　検体数別の解析処理時間
Table 1　Analysis Processing Time by Number of Specimens

検体数	1 回目 (s)	2 回目 (s)	3 回目 (s)
1	4.38	4.00	4.10
10	13.6	13.5	13.7
100	179	178	179
200	345	348	346

表 2　検体数 200 におけるスクリプト別の解析処理時間
Table 2　Analysis Processing Time by Script for 100 Sample

スクリプト	1 回目 (s)	2 回目 (s)	3 回目 (s)
FFRI	264	269	269
Text	1.53	1.51	1.50
Unpack	21.5	22.5	21.2
Stella	48.9	49.1	49.1

4.8　考察

本章での検証結果をもとに考察を述べる．FFRI Dataset-scripts による検体 A の解析結果から，FFRI Dataset と同等の内容に加え，Windows バイナリに対応した strings 結果，Magika によるファイル形式の識別結果それぞれを確認した．strings_plus が追加した strings 情報であり，通常の strings 結果には出力されていない文字列を確認した．このことから，追加した機能から得られる情報は有用である．また，検体 B に関しては，Windows 系で動作するバイナリではないため，strings_plus の情報に null を確認した．Stella による検体 B と検体 C の解析結果から，ハッシュ値などの基本情報を確認した．また，検体に含まれる文字列から予測される検体の動作情報を確認した．これらのことから，Stella による解析結果は明瞭である．テキストファイルに対する解析結果から，検体 E と検体 F の両方で各種情報の出力を確認した．これにより，テキストファイルに対する解析を実現した．そして，パッカー解除前後での解析結果の比較では，FFRI Dataset-scripts による解析結果と Stella による解析を比較している．明確に変化した情報として，ハッシュ値と strings 結果が挙げられる．特に strings 結果では，パッカー解除前では表示されていなかった情報が表示されており，Stella による解析結果が明確に表示された．ここでは一部の解析結果しか記載していないが，strings 結果における解除前後での情報量は，解除前と比較して増加した．このことから，本システムにおけるパッカーの検知および解除機能は，正常に適用されており，有用であるといえる．また，解析処理時間の計測では，処理時間に zip ファイルの解凍などの処理も含むため，100 検体での解析処理時間で 1 検体あたりの平均処理時間を考えると，約 1.8 秒である．これは手動での解析よりもはるかに早いため，自動解析システムとして有用であるといえる．

5.　結論・今後の課題

本論文では，FFRI Dataset-scripts を利用した自動表層解析システムについて提案した．FFRI Dataset-scripts に対して Windows バイナリに対応する strings コマンド，Google 社が提供する Magika によるファイル形式解析をそれぞれ追加し，テキストファイルに対する解析を別のスクリプトで実行することで，得られる表層情報の拡張を実現した．Stella と Capa での解析により，得られた表層情報を元に予測される検体の攻撃手法の出力し，得られる表層情報の拡張と明瞭化を実現した．また，パッカーの検知および解除の自動化を行うことで，得られる情報量の増加，解析の効率化を実現した．これらのことから，本研究の目的である表層情報の自動取得および解析結果の明瞭化を実現したといえ，本システムは表層解析システムとして有用であるといえる．

今後の課題として，難読化の解除が挙げられる．本システムにおける難読化の自動解除は UPX のみであるが，マルウェアが使用する難読化手法は多岐にわたっている．そのため，代表的なパッカーの追加のみならず，独自の難読化手法にも対応した難読化の自動解除システムを構築することで，得られる表層情報の増加が期待できると考える．

謝辞　本研究の一部は，JSPS 科研費 23K11108, 23H03396 の支援により行った．

参考文献

[1] 荒井晃平, 小林良太郎, “多種多様なマルウェアに対する自動表層解析システムの提案,” コンピュータセキュリティシンポジウム 2023 論文集, pp. 37-43, 2023.
[2] FFRI Security, Inc., *“FFRI Dataset scripts,”* https://github.com/FFRI/ffridataset-scripts (Accessed 2024-07-19).
[3] Y.M.P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama and C. Rossow, *“IoTPOT: A Novel Honeypot for Revealing Current IoT Threats,”* Journal of Information Processing, vol. 57, no. 4, 2016.
[4] Y.M.P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama and C. Rossow, *“IoTPOT: Analysing the Rise of IoT Compromises,”* 9th USENIX Workshop on Offensive Technologies (USENIX WOOT 2015), 2015.
[5] 岡山あん, 朝倉紗斗至, 中川恒, 押場博光, 市野将嗣, “動的解析ログと表層情報を組み合わせたマルウェア感染活動の最終進行度推定手法,” コンピュータセキュリティシンポジウム 2021 論文集, pp. 1021-1028, 2021.
[6] 茂木裕貴, *“PE 表層情報を用いた機械学習による静的マルウェア検知,”* https://www.ffri.jp/research/index.html (Accessed 2023-12-08).
[7] M. Mimura, *“Evaluation of printable character-based malicious PE file-detection method,”* Internet of Things, Vol. 19, Article 100521, 2022.
[8] FFRI Security, Inc., *“FFRI Dataset,”* https://www.iwsec.org/mws/2023/files/MWS2023_dataset_FFRI.pdf (Accessed 2023-12-11).
[9] *“Stella,”* https://github.com/crum7/Stella (Accessed

2024-07-19).
[10] *"Capa,"* https://github.com/mandiant/capa (Accessed 2024-07-19).
[11] *"Docker,"* https://www.docker.com/(Accessed 2024-07-19).
[12] *"Magika,"* https://github.com/google/magika (Accessed 2024-07-19).
[13] *"Google Open Source Blog,"* https://opensource.googleblog.com/2024/02/magika-ai-powered-fast-and-efficient-file-type-identification.html (Accessed 2024-07-19).
[14] *"UPX,"* https://upx.github.io/(Accessed 2024-07-19).
[15] *"ATT&CK,"* https://attack.mitre.org/(Accessed 2024-07-19).