平成 27 年度 修士論文

エキスパートによるマルウェア解析レポートと動的解析ログの相関分析

Correlating Experts' Malware Analysis Reports and Dynamic Malware Analysis Logs

指導教員　森 達哉 准教授

早稲田大学基幹理工学研究科情報理工・情報通信専攻

学籍番号 5114F077-2

藤野　朗稚

2016 年 2 月 1 日

概要

アンチウィルスベンダーは日々大量のマルウェアを解析し，その結果はマルウェア解析レポートとしてデータベースに蓄積されている． 一般にマルウェア解析レポートは自然言語で記述されており，マルウェアがアクセスするファイルやレジストリ，通信先，関連するマルウェア種別等の情報が記載されている． それらの情報は挙動の対象に関する簡易的なもので，実際に使用される API や引数等に関する詳細な情報は記載されていない． また，解析レポートはマルウェア種別毎に独立しているため，同じ挙動を持つ他の種別を調べることは難しい． 本論文では，マルウェア解析のエキスパートによって作成された解析レポートと動的解析ログを対応付けるデータベースの作成を狙いとする． このデータベースを使うことにより，動的解析ログから既存の悪性挙動を自動的に検出可能となること，解析レポートのドラフトを自動生成可能となることが期待される．実データを用いた解析の結果，異なるマルウェア種別や種別不明のマルウェアからも共通する悪性挙動が検出可能であることが明らかになった．また，解析レポートに記述されていない挙動を行うマルウェアが多くを占めることが明らかになった．

目次

第 1 章　序論　11

1.1　はじめに　11

1.2　提案　12

1.3　貢献　12

1.4　論文の構成　13

第 2 章　分析データ　15

2.1　FFRI Dataset の概要　15

2.2　Microsoft 社の解析レポートについて　20

2.2.1　Summary　20

2.2.2　What to do now　20

2.2.3　Technical information　20

2.2.4　Symptoms　23

2.3　Microsoft 社の解析レポートの収集　23

第 3 章　提案手法　25

3.1　提案手法の概要　25

3.2　解析レポートに対する前処理　26

3.3　悪性挙動の抽出方法　29

3.3.1　ファイル操作系の分類，抽出方法　30

3.3.2　レジストリ操作系の分類，抽出方法　31

3.3.3　ネットワーク操作系の分類，抽出方法　32

3.3.4　ミューテックス操作系の分類，抽出方法　33

3.3.5　正規表現への変換方法　33

3.3.6　悪性挙動の検出　34

3.3.7　検出結果の出力　35

第 4 章　結果　39

4.1 解析レポートから抽出した悪性挙動について 39
4.2 動的解析ログから検出した悪性挙動について 39
4.2.1 正解に対するカバレッジについて 42
4.2.2 未知のマルウェア検体について 45
4.2.3 解析レポートの草案の自動生成 46

第 5 章 議論 57
5.1 制限事項 57
5.1.1 悪性挙動の抽出 57
5.1.2 悪性挙動の検出 58
5.2 今後の展望 59

第 6 章 関連研究 61

第 7 章 まとめ 63

第 8 章 研究業績 65

参考文献 67

謝辞 71

付録 A 解析レポートから抽出した悪性挙動の抜粋 73

図目次

2.1 Virustotal の検査結果の例 16
2.2 API コールの例..... 16
2.3 マルウェアが行った通信に関する情報の例..... 17
2.4 Win32/Vobfus [7] における Technical information の構造の例 21
2.5 解析レポートの書式の例 23

3.1 提案手法の概要図 25

4.1 1 検体に含まれる悪性挙動数の CDF 41
4.2 1 検体に含まれる悪性挙動数の CDF ($0 \leq x \leq 100$)..... 41
4.3 1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2013) 43
4.4 1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2014) 43
4.5 1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2015) 44
4.6 正解レポートに対するカバレッジの CDF (FFRI Dataset 2013)..... 44
4.7 正解レポートに対するカバレッジの CDF (FFRI Dataset 2014)..... 45
4.8 正解レポートに対するカバレッジの CDF (FFRI Dataset 2015)..... 45
4.9 未知検体の悪性挙動数の CDF 46
4.10 既知検体の悪性挙動数の CDF 46
4.11 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (1 ページ目)..... 47
4.12 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (2 ページ目)..... 48
4.13 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (3 ページ目)..... 49
4.14 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (4 ページ目)..... 50
4.15 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (5 ページ目)..... 51
4.16 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (6 ページ目)..... 52
4.17 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (7 ページ目)..... 53
4.18 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (8 ページ目)..... 54
4.19 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (9 ページ目)..... 55
4.20 Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (10 ページ目) ... 56

A.1 ファイル操作系悪性挙動の抜粋 73
A.2 レジストリ操作系悪性挙動の抜粋 74
A.3 ネットワーク操作系悪性挙動の抜粋 75
A.4 ミューテックス操作系悪性挙動の抜粋 76

表目次

2.1 カテゴリごとの API の呼び出し回数 (FFRI Dataset 2013) 18
2.2 カテゴリごとの API の呼び出し回数 (FFRI Dataset 2014) 18
2.3 カテゴリごとの API の呼び出し回数 (FFRI Dataset 2015) 19
2.4 Microsoft 社におけるマルウェア種別に関する分布 19
2.5 解析レポートで使われる項目名の出現回数 22
2.6 収集した解析レポートのマルウェア種別について 24

3.1 同義語の意味と表記の組み合わせ一覧 28
3.2 補足説明時に使用される接頭辞一覧 29
3.3 レポートの書式と記述内容 29
3.4 Microsoft 社の解析レポートで使用されている環境変数 [23]，[24] 31
3.5 ファイルの拡張子の一部 36
3.6 レジストリ操作系悪性挙動の記述方法 37
3.7 トップレベルドメイン一覧 [26] 37
3.8 抽象的表現で指定される文字列の型の抜粋 37
3.9 Behavior API MAP 38
3.10 悪性挙動の検索時に必要な情報 38

4.1 抽象的表現で指定される文字列の型の抜粋 39
4.2 各 FFRI Dataset における悪性挙動の検出数 40
4.3 各データセットにおける API 呼び出し回数 40
4.4 FFRI Dataset 2015 において悪性挙動を検出できなかったマルウェア種別上位 30 に関するまとめ 42
4.5 各データセットにおけるカバレッジ 43
4.6 既知，未知検体における悪性挙動検出数 45

第 1 章　序論

1.1　はじめに

マルウェアの解析方法は静的解析と動的解析の 2 つに大別される．静的解析はマルウェアの実行ファイルを解析するためより深く解析することができるが，難読化のようなマルウェア解析対策技術が施されている場合は対処できない．一方，動的解析は実際の感染者の環境と近い状況でマルウェアを動作させ，マルウェアが実際に行う挙動を解析する．そのため，難読化のようなマルウェア解析対策技術の影響を受けない．そのような理由から，マルウェア解析対策技術に対抗するためには動的解析を用いる必要がある．

AV-TEST [1] の調査では 2015 年末時点において，一日に検出される新種のマルウェアは約 39 万種，今までに発見されたマルウェア種別の総数は約 4 億 7 千万種であると報告されている [2]．各アンチウィルスベンダーは (以下，ベンダーと記述する) このように大量生産されるマルウェアの解析を日夜行っている．そのマルウェアの解析結果 (以下，マルウェア解析レポートと記述する) は各ベンダーのデータベースに蓄積されている．そして，一部のベンダーではマルウェア解析のレポートをインターネット上で公開している．公開されているレポートはベンダーに属するマルウェア解析のエキスパートが作成したものであり，マルウェアの悪意ある行動 (以下，悪性挙動と記述する) に関する詳細情報が記載されている．具体的にはマルウェアの概要，悪性挙動に関連するファイル，レジストリ，ネットワーク通信などの情報，他のマルウェアとの関係，感染時の対処方法，レポートの作成者等が記録されている．一般に解析レポートはマルウェアの種別毎に作成されるため，解析が行われたマルウェア種別の数と比例する大量の解析レポートが各ベンダーのデータベースに蓄積されている．インターネット上で公開されている解析レポートは自然言語で記述されており，実際に使用された Application Programming Interface (以下，API と記述する) や API に与えられた引数，挙動の順序などの詳細な情報を知ることはできない．また，解析レポートはそれぞれが独立したドキュメントとなっているため，種別間の類似点や相違点を調べることは難しい．

これらの解析レポートは，新たにマルウェア解析を行う上での重要な手掛かりとなり得る．重要な手掛かりとなり得ると考える理由は以下の 2 つが挙げられる．

- マルウェア解析のエキスパートが作成したものであるため，マルウェアの挙動に関する確度の高い詳細情報が得られる．

- ベンダーは日夜解析を続けているため，蓄積されている解析レポートからは膨大な量のマルウェアに関する情報を得られる．

また，解析レポートに含まれる主な情報はマルウェアが動的に行う挙動に関するものであるため，前述した動的解析と親和性が高いと考えられる．

1.2 提案

本論文では, そのような背景にもとづきエキスパートによる解析レポートと動的解析ログの相関分析を行う. そして，分析結果を利用した悪性挙動データベース (以下，悪性挙動 DB と記述する) の作成を目指す．悪性挙動 DB では，解析レポートから抽出した悪性挙動と実際にマルウェアが使用する API 名や引数値を紐付ける．悪性挙動 DB の作成手順を以下に示す．

1. 解析レポートから悪性挙動を抽出する
2. 抽出した悪性挙動から種類，挙動の対象，挙動によって引き起こされる脅威，実際に使用される API 名や引数値などを定義する
3. 定義した情報をデータベースに保存する

悪性挙動 DB を用いることで，動的解析を行う際にデータベース内の挙動を既知の悪性挙動として自動検出することが出来る．また，解析レポート中の挙動の説明文と実際の挙動を結びつけることで，解析レポートの作成を自動化することも可能になる．それ以外にもマルウェアの悪性挙動を自動検出することができることを利用して研究用データセットのラベリングにも利用することができる．

本研究では以下の 2 つを分析対象にする．

- 8,640 のマルウェア検体に対して Cuckoo Sandbox を適用して収集したログ
- Microsoft 社 [3] のマルウェア種別 1,678 種に関する解析レポート

上記 1 社の解析レポートから悪性挙動とその説明文を抽出し，それらの情報を元に悪性挙動を定義する．そして，定義した悪性挙動をデータベースに保存する． 次に，データベースを用いて動的解析ログから悪性挙動を検出し，解析レポートの草案を作成する． また， これらの処理の全てを自動化する．

1.3 貢献

本研究の主要な貢献は下記のとおりである．

- マルウェア解析レポートから悪性挙動を自動的に抽出する方法を開発した．
- マルウェア動的解析ログから悪性挙動を自動検出できることを示した．
- マルウェア動的解析ログからマルウェア解析レポートの草案を自動生成する方法を開発した．

これらの貢献により，マルウェア解析時に要する時間的コストの削減が期待される．また，マルウェアに関する研究を行っている研究者の補助ツールとして利用されることが期待できる．

1.4 論文の構成

本論文の構成は以下の通りである．はじめに 2 章では本論文の分析対象について述べる．つづいて 3 章では提案手法を紹介し，4 章で分析結果を述べる．5 章では本論文の制限事項と今後の展望を述べる．6 章では関連研究について述べ，7 章にて本論文のまとめを述べる．最後に 8 章で本論文の著者の業績を紹介する．

第 2 章　分析データ

本章では，分析に利用するデータの詳細と，解析レポートの収集方法を示す．

2.1　FFRI Dataset の概要

本研究では， MWS の研究用データセット [4] の一部として FFRI 社が提供している FFRI Dataset 2013 と FFRI Dataset 2014, FFRI Dataset 2015 を用いる．以下に FFRI Dataset 2013, 2014, 2015 の主な特徴を記す．

共通する特徴: データは JSON 形式で保存されている動的解析ログである．動的解析に使用したマルウェア検体は PE 形式かつ実行可能で，1 検体あたりの実行時間は 90 秒となっている．

FFRI Dataset 2013: 動的解析には Cuckoo Sandbox が用いられている．マルウェア検体の収集期間は 2012 年 9 月から 2013 年 3 月までとなっており，解析対象となる検体数は 2,644 検体となっている．

FFRI Dataset 2014: 動的解析には Cuckoo Sandbox と FFRI yarai analyzer Professional が用いられている．本論文では FFRI yarai analyzer Professional のログを使用しない．マルウェア検体の収集期間は 2014 年 1 月から 2014 年 4 月までの期間となっており，解析対象となる検体数は 3,000 検体となっている．

FFRI Dataset 2015: 動的解析には Cuckoo Sandbox が用いられている．マルウェア検体の収集期間は 2015 年 1 月から 2015 年 4 月までの期間となっており，解析対象となる検体数 FFRI Dataset 2014 と同様に 3,000 検体となっている．

以下に Cuckoo Sandbox のログの詳細を述べる．ログに記録されている情報は静的な情報と動的な情報の 2 つに大別される．静的な情報では，マルウェア検体のハッシュ値やファイル名などのファイル情報，実行ファイル中に出現する文字列， VirusTotal [5] によるマルウェア検体の検査結果が記録されている．VirusTotal の例を図 2.1 に示す．動的な情報では，実行時の API 呼び出しや，マルウェアが行った通信に関する情報，実行時に生成したファイル情報，アクセスしたファイルやレジストリに関する概要が記録されている．API 呼び出しは時系列順に記録されており，各 API の関数名や引数値，カテゴリ，実行の成否などの詳細な情報が記録されている．その例を図 2.2 に示す．マルウェアが行った通信に関する情報の例は図 2.3 に示す．前述

```
"virustotal": {
    "scan_id": "ff3ebac6e2e4bc2cd95e1148f318df7829bcd9533cd2248ec487b8c8eec96de9-1352932528",
    "sha1": "c8677d117c4d294dce3a523f69184a808e8d0e74",
    "resource": "276fba8f53b143e92b942831072bb51d",
    "response_code": 1,
    "scan_date": "2012-11-14 22:35:28",
    "permalink": "https://www.virustotal.com/file/ff3ebac6e2e4bc2cd95e1148f318df7829bcd9533c
        d2248ec487b8c8eec96de9/analysis/1352932528/",
    "verbose_msg": "Scan finished, scan information embedded in this object",
    "sha256": "ff3ebac6e2e4bc2cd95e1148f318df7829bcd9533cd2248ec487b8c8eec96de9",
    "positives": 36,
    "total": 44,
    "md5": "276fba8f53b143e92b942831072bb51d",
    "scans": {
        "Microsoft": {
            "detected": true,
            "version": "1.8904",
            "result": "Worm:Win32/Vobfus.GZ",
            "update": "20121114"},
        "Symantec": {
            "detected": true,
            "version": "20121.2.1.2",
            "result": "W32.Changeup!gen20",
            "update": "20121114"},
        ...
    }
},
```


図 2.1　Virustotal の検査結果の例


```
"category": "system",
"status": "FAILURE",
"return": "0xc0000135",
"timestamp": "2013-02-28 12:03:55,656",
"thread_id": "432",
"repeated": 1,
"api": "LdrGetDllHandle",
"arguments":
    [
        {
            "name": "FileName",
            "value": "C:\\WINDOWS\\system32\\rpcss.dll"
        },
        {
        "name": "ModuleHandle",
        "value": "0x00000000"
        }
    ]
```


図 2.2　API コールの例

した VirusTotal の検査結果には 1 検体あたり最大で 51 種類のアンチウイルスエンジン (以下，AV エンジンと記述する) による検査結果が含まれている．本論文では Cuckoo Sandbox のログ内の API コールログ， VirusTotal の検査結果，通信の概要を用いる．

動的な情報に含まれる API のカテゴリごとの呼び出し回数を各データセット毎にまとめたものを表 2.1，2.2，2.3 に示す．各表から レジストリの操作に使用される registry，ファイルの操作に使用される filesystem の API の割合が大きいことがわかる．具体的には，FFRI Dataset 2013 では 32.72 %，2014 では 54.07 %，2015 では 64.77 % である．詳しくは後述するが，解

```
"network": {
    "udp": [
        {
            "dport": 53,
            "src": "111.22.33.251",
            "dst": "111.22.33.1",
            "sport": 1033
        }
    ],
    "http": [
        {
            "body": "",
            "uri": "http://5.199.175.164/content/offers/default.aspx",
            "user-agent": "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)",
            "method": "GET",
            "host": "5.199.175.164",
            "version": "1.1",
            "path": "/content/offers/default.aspx",
            "data": "GET /content/offers/default.aspx HTTP/1.1\r\nAccept: */*\r\nCookie:
                pb_session=OxH17xSrWNHxFg7hUcya8zPx4sIbnBDE_ISyLtZe1O_jReLddz6YUQ3txyfSL
                F34yIEOwnbh1mKPqunGsRAz96S9RW3NMCiNkfyUDzdA2QSpxaox0TfC4Y6akY2zzNHVQgmRw
                aTLowXYUGMYEPLFKTWnU5XIx2GB25SEg6lf9xI; WC_PERSISTENT=nSDXZjYGxf6Ni4v2Zb
                UpRw9fuzkhMg1S6G7W9N3BMAUlqSApMq89Z4YtaWEb2dBki8iK7jtPJzy3YsMeVUvjQud4CH
                TYMsS8Q_P93SrPGq8b2yYIzrO3j7wp9uZXl1ndvfmOnfmxZscCh4sIhjakhpuxcARg-Ukwvp
                lnxFpLr; imgshck=zp1=z3VsO8fJI8-U9yTBBc8rdKVLdyDCM8jADhydO7n8eujnfBcTkCi
                QaSKns8f4GuRRT_OMaP2BVtp4BrjBI3aB0tytLGya8bMquehYp7ATZphl1VcaBzm5wpsYcWW
                JDSEf7FWS0Jq4QNyOF_Q1G4kCqG70LMHlZYEXhSA-EDCDFzoQ82jr-xJAx3VVF-Gj8KM7yJH
                Q50BA&page=Vmbwl5RtumFQLSAcpWl8B2L5vZNlGaBmUu1NpiJZrm9mW1; clogid=pAWoXO
                XhO6j_Jz; member_session=lc0=eT2pst3m3q6XZ4cXes\r\nUser-Agent: Mozilla/4
                .0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)\r\nHost: 5.199.175.164\r\
                nConnection: Close\r\nCache-Control: no-cache\r\nPragma: no-cache\r\n\r\n",
            "port": 80
        },
    ],
    "smtp": [
        {
            "raw": "EHLO server\r\nEHLO server\r\nAUTH LOGIN\r\nAUTH LOGIN\r\nYm9pZn
                JlbmQ1Ng==Ym9pZnJlbmQ1Ng==\r\n\r\ncmZoZnJla3oycmZoZnJla3oy\r\n\r\n",
            "dst": "94.100.177.1"
        }
    ],
    "tcp": [
        {
            "dport": 80,
            "src": "111.22.33.251",
            "dst": "5.199.175.164",
            "sport": 1034
        },
    ],
    "hosts": [
        "111.22.33.251",
        "111.22.33.1",
        "5.199.175.164",
        "111.22.33.255"
    ],
    "dns": [
        {
            "type": "A",
            "request": "intohave.com",
            "answers": [{
                    "data": "29.172.39.109",
                    "type": "A"
                }]}],
    "domains": [{
            "ip": "29.172.39.109",
            "domain": "intohave.com"
        }],}
```


図 2.3　マルウェアが行った通信に関する情報の例

析レポートから抽出される悪性挙動の大部分はファイル操作とレジストリ操作に関するものであるため，本論文では主に filesystem と registry の API を扱う．

表 2.1　カテゴリごとの API の呼び出し回数 (FFRI Dataset 2013)

カテゴリ	呼び出された回数 (割合)
system	807,960 (47.43%)
registry	380,413 (22.33%)
process	233,080 (13.68%)
filesystem	177,039 (10.39%)
device	62,122 (3.64%)
synchronization	12,267 (0.72%)
sleep	8,153 (0.47%)
hooking	5,697 (0.33%)
windows	4,965 (0.29%)
threading	4,750 (0.27%)
network	3,465 (0.20%)
services	2,743 (0.16%)
socket	609 (0.03%)

表 2.2　カテゴリごとの API の呼び出し回数 (FFRI Dataset 2014)

カテゴリ	呼び出された回数 (割合)
filesystem	10,172,592 (29.08%)
registry	8,743,954 (24.99%)
system	7,817,701 (22.35%)
process	3,624,065 (10.36%)
misc	3,208,983 (9.17%)
socket	443,762 (1.26%)
windows	430,130 (1.22%)
synchronization	301,751 (0.86%)
threading	141,947 (0.40%)
services	41,135 (0.11%)
device	25,923 (0.07%)
network	25,789 (0.07%)
hooking	483 (0.00%)

本論文では解析レポートを収集する際に VirusTotal の検査結果から得られる Microsoft 社のマルウェア種別名を用いる．FFRI Dataset 全体における Microsoft 社のマルウェア種別に関する情報を表 2.4 にまとめる．表中の種別名が判明していない検体とは，図 2.1 中のベンダーの検査結果で種別名が記載されていない検体，動的解析ログ中に VirusTotal の検査結果それ自体が

表 2.3　カテゴリごとの API の呼び出し回数 (FFRI Dataset 2015)

カテゴリ	呼び出された回数 (割合)
registry	1,624,371 (45.92%)
filesystem	666,804 (18.85%)
process	598,425 (16.92%)
system	266,723 (7.54%)
misc	170,397 (4.81%)
synchronization	76,363 (2.15%)
socket	46,751 (1.32%)
windows	31,060 (0.87%)
threading	25,367 (0.71%)
device	13,904 (0.39%)
anomaly	11,989 (0.33%)
network	2,639 (0.07%)
services	1,493 (0.04%)
hooking	479 (0.01%)

保存されていない検体を指す．本論文ではマルウェア種別名が判明している検体を既知の検体，判明していない検体を未知の検体とする．

表 2.4　Microsoft 社におけるマルウェア種別に関する分布

	Microsoft
マルウェア種別数	1,299
種別名が判明している検体数	4,967
種別名が判明していない検体数	3,673

Microsoft 社のマルウェア種別名の命名規則を以下に示す [6].

```
Type:Platform/Family.Variant[!Information]
```

Type はマルウェアが端末に与えうる脅威を表しており，Backdoor，Trojan，Worm，PWS などがある．Platform はマルウェアが対象とするプラットフォームやプログラミング言語，ファイルフォーマットを表しており， Win32，JS， Java などがある．Family は同じ脅威をもつマルウェアのグループ (以下，マルウェアファミリーと記述する) を表しており， Vobfus，Zbot などがある．このファミリー名は AV ベンダー間で異なるものを使用している場合がある．Variant はファミリーの亜種名を表しており， AE や AF のような意味を持たない文字列が与えられる．Information は追加情報を表している．本論文では上記の命名規則の内，Platform と Family のみで表されたマルウェア種別名を原種と呼ぶ．例えば， Worm:Win32/Vobfus.F というマルウェア種別名が存在する場合，その原種は Win32/Vobfus となる．この原種は後述する解析レポート

の収集で使用する．

2.2 Microsoft 社の解析レポートについて

本研究で悪性挙動の抽出に用いる Microsoft 社の解析レポートについて以下に述べる．解析レポートは Microsoft 社のウェブサイト上で公開されており，URL は以下に示す生成規則に従っている．

```
http://www.microsoft.com/security/portal/threat/encyclopedia/
entry.aspx?Name="マルウェア種別名"
```

本論文で使用する Microsoft 社の解析レポートは上記の URL と FFRI Dataset から得られる Microsoft 社のマルウェア種別名を使用して収集する．

解析レポートの詳細を以下に示す．解析レポートは HTML で記述されており，レポートごとに異なるマルウェア解析者によって作成される．解析レポート中には必ずマルウェア種別名，解析レポートが最初に作成された年月日，解析レポートが最後に編集された年月日が記載されている．解析レポートの記述内容は "Summary"，"What to do now"，"Technical information"，"Symptoms" の４つから構成される．以下にそれぞれの記述内容を示す．

2.2.1 Summary

マルウェアの概要が記述されている．具体的には，マルウェアがどのような経路で感染するのか，マルウェアが引き起こす脅威，他のマルウェアとの関係などが簡潔に記述されている．

2.2.2 What to do now

マルウェアに感染した際にとるべき行動について書かれている．具体的には，マルウェアを検知，削除するために利用すべきソフトウェア，マルウェアに書き換えられた設定項目の修正方法，リムーバブルドライブがマルウェア感染しているかどうかの確認方法などが記述されている．

2.2.3 Technical information

マルウェアの悪性挙動に関する詳細情報が示されている．Technical information の構造の例を 2.4 に示す．

図 2.4 で示すように Technical information は id が tab-link-3C となっている div タグで表されている．Microsoft 社の解析レポートは不定期で更新されており，ここで示した id や構造に変化が生じる場合がある．図 2.4 からわかるように Technical information ではマルウェアの悪性挙動によっていくつかの項目に分けて説明される．この場合では Installation，Spread via，

```
<div id="tab-link-3C">
    <h3>Threat behavior</h3>
        <p>
            Vobfus is often downloaded by other malware,
            and also downloads other malware itself, including:
        </p>
        <ul>
            <li>Win32/Beebone</li>
            <li>Win32/Fareit</li>
            <li>Win32/Zbot</li>
        </ul>
    <h5>Installation</h5>
        <p>
            In the wild, we have observed variants of Vobfus being
            downloaded by variants of Win32/Beebone.
        </p>
        <p>
            This threat creates a mutex named "A" to mark its infection,
            and to make sure that only a single copy of its process is
            running on your PC at any one time.
        </p>
        <p>
            It then drops a copy of itself in the
            "C:\Documents and Settings\<user>" folder
            using a random file name, for example:
        </p>
        <p>C:\documents and settings\Administrator\zkyip.exe.exe</p>
        ...
    <h5>Spreads via...</h5>
        <p>Network and removable drives</p>
        <p>
            The worm copies itself to the root directory of network
            and removable drives using "rcx<hexadecimal number>.tmp",
            then renames this TMP file to any of the following:
        </p>
        <ul>
            <li>passwords.exe</li>
            <li>porn.exe</li>
            <li>secret.exe</li>
            <li>sexy.exe</li>
            <li>subst.exe</li>
            <li>system.exe</li>
        </ul>
        ...
    <h5>Payload</h5>
        <p>Changes PC settings</p>
        <p>
            Worm:Win32/Vobfus changes the following registry entries
            to prevent you from changing how hidden files and folders
            are displayed in Windows Explorer:
        </p>
        <p>
            In subkey: HKCU\Software\Microsoft\Windows\CurrentVersion
                    \Explorer\Advanced<br>
            Sets value: "ShowSuperHidden"<br>
            With data: "0"
        </p>
        <p>Downloads and runs other malware</p>
        <p>
            Worm:Win32/Vobfus tries to connect to a remote host to receive
            encrypted commands that, when decrypted, specify the following:
        </p>
        <p><URL to download><Save as file name></p>
        ...
</div>
```

図 2.4　Win32/Vobfus [7] における Technical information の構造の例

Payload の 3 項目に分けて説明がされている．解析レポート中で使われる項目名の出現回数 Top 5 を表 2.5 に示す．この表中の割合は今回収集した 1685 の解析レポートの内，Technical information 内に悪性挙動の記述が存在する 973 の解析レポートに対する値となっている．

表 2.5　解析レポートで使われる項目名の出現回数

項目名	出現レポート数 (割合)
Payload	668 (68.65)
Installation	663 (68.13)
Spreads via...	212 (21.78)
Additional information	139 (14.28)
Related encyclopedia entries	18 (1.84)

表 2.5 で示した項目の概要を以下に示す．

Installation: 感染時に行われる挙動に関する情報が記載される．具体的には，PC 起動時のマルウェアを自動実行するための設定変更やマルウェア自身のコピーを作成する際のファイル名，コピー先のディレクトリ名などである．

Spread via: 感染活動の挙動や感染方法に関する情報が記載される．具体的には，リムーバブルドライブへ自身のコピーを作成する際のファイル名，感染活動に使われるスパムのタイトルや本文などである．

Payload: メインの挙動に関する情報が記載される．具体的には，設定変更される項目，生成するファイル，通信先アドレスなどが示される．

Additional information: 他の項目で説明されなかった追加情報が記載される．具体的には，マルウェア動作時に現れるメッセージボックスの説明と画像，マルウェアが作成するミューテックス，マルウェアが検出する自動解析環境についてなどが記載される．

Related encyclopedia entries: 関連のある他の種別の解析レポートへのリンクが列挙されている．例えば，マルウェア種別 Win32/Sality[8] の解析レポートでは Trojan:WinNT/Sality[9], TrojanSpy:Win32/Keatep.B[10]，Virus:Win32/Sality.AM[11]，Virus:Win32/Sality.G[12], Virus:Win32/Sality.G.dll[13], Virus:Win32/Sality.AT[14], Virus:Win32/Sality.AU[15], Win32/Bagle[16], Worm:Win32/Bagle.IF@mm[17]，Worm:Win32/Sality.AU[18] へのリンクが記載されている．

Threat behavior では前述した項目に関係なく，図 2.5 で示す形式のいずれかで悪性挙動についての記載がされている．書式 1 では悪性挙動についてや他のマルウェアとの関係など様々な記述がされる．書式 2 では悪性挙動の説明とその対象となるファイル，ディレクトリ，プロセス，通信先ドメイン，フックされる API，関連するマルウェアなどが記載される．多くの場合で説明文の語尾が ":" となっており，それ以降に各情報が列挙される．また，挙動そのものについてだけでなく，挙動の目的やレポート内で使用された抽象的な表現の例なども記載される．本論文では <random>, <user name>, <product name> などの一意に定めることができない表

```
書式 1
    悪性挙動の説明文など

書式 2
    悪性挙動の説明文など:
    ・要素 1
    ...
    ・要素 X

書式 3
    レジストリ操作に関する説明文:
    In subkey:  "対象のレジストリキー"
    Sets value: "追加，変更されるレジストリエントリ名"
    With data:  "追加，変更されるレジストリエントリ値"

書式 4
    スパムに関する説明文:
    Subject: "メールタイトル"
    Attachment: "添付ファイル"
```

図 2.5　解析レポートの書式の例

現のことを抽象的な表現としている．書式 3 ではレジストリの設定変更に関する情報が記載される．図 2.5 は一例であり，エントリ名やエントリ値に関する記述がないもの，1 つのレジストリキーに対して複数のエントリ名，エントリ値の組が記載されるものなど複数の記述方法が存在する．書式 4 ではマルウェア感染に用いられるスパムメールに関する情報が記載される．書式 3 と書式 4 は書式 2 の要素として階層的に使用されることもある．

作成者が異なる解析レポートには図 2.5 で示す書式や HTML タグの使い方が異なるものが存在する．また，人為的なミスによってスペルミスが含まれるレポートも存在する．

2.2.4　Symptoms

マルウェア感染時に見られる設定やファイルについて書かれている．具体的には，マルウェアが変更するレジストリの設定項目とその設定値，マルウェアが生成するファイルなどについて書いてある．この項目の内容は Technical information で述べられた内容の一部を抜粋したものとなっている．

本論文では，解析レポートから悪性挙動を抽出する際に Technical information のみを使用する．これは悪性挙動の詳細について記載されているのは Technical information のみであり，他の項目はマルウェアへの対処方法や Technical information の抜粋，まとめとなっているからである．

2.3　Microsoft 社の解析レポートの収集

本節では，悪性挙動の抽出に用いる Microsoft 社の解析レポートの収集方法を示す． 2.2 節で述べたように解析レポートへの URL は以下の生成規則に従っている．

```
http://www.microsoft.com/security/portal/threat/encyclopedia/
entry.aspx?Name="マルウェア種別名"
```

この生成規則と FFRI Dataset から得られた 1,299 種のマルウェア種別名を用いて解析レポートを収集する．収集の手順を以下に示す．

1. マルウェア種別名と URL の生成規則から URL のリストを作成する
2. リストの中から未処理の種別名の URL を一つ選択する
3. 選択した URL を用いて解析レポートを収集する
4. 収集した解析レポートから他の解析レポートへのリンクを抽出する
5. 抽出したリンクの中から未収集のマルウェア種別名のものをリストに追加する
6. 3 ～ 6 の処理を繰り返す
7. FFRI Dataset から抽出したマルウェア種別の原種について 2 ～ 6 の処理を行う

本論文では上記のように，解析レポート中に出現する他の種別に関する解析レポートも収集の対象としている．これはマルウェア種別によっては解析レポートの Technical information で悪性挙動について述べられず，同一の悪性挙動を持つ他の解析レポートへのリンクのみが記載されている場合があるからである．また，本論文では原種の解析レポートも収集の対象としている．これは原種の解析レポートにはファミリーに共通する挙動が記載されており有用だからである．

上記の方法によって得られた解析レポートの数についてまとめたものを表 2.6 に示す．前述したように FFRI Dataset から抽出したマルウェア種別数は 1,299 で，そこから生成した原種名は 546 であった．この数値と表 2.6 から，FFRI Dataset で使われているマルウェア種別の 1.39% に関しては種別名が存在するのに解析レポートが存在しないこと，本論文で定義した原種に関しては 76.2% で解析レポートが存在していないことがわかる．これは本論文で定義した原種は特定のマルウェアファミリーでのみ存在するということを示している．

表 2.6　収集した解析レポートのマルウェア種別について

種別名の由来	種別数
FFRI Dataset から抽出したもの	1,281
FFRI Dataset から抽出したものの原種	130
レポート内のリンクから抽出したもの	267
計	1,678

第 3 章　提案手法

本章では，提案手法の概要，2.3 節で収集した解析レポートに対する前処理方法，レポートからの悪性挙動の抽出方法，動的解析ログから悪性挙動を検出する方法，各マルウェア検体に関する解析レポートの草案作成方法を示す．

3.1　提案手法の概要

提案手法の概要を図 3.1 に示す．

図 3.1 中の構成要素の詳細を以下に示す．解析レポートは 2.3 節で収集したものであり，図 3.1 中の 1 つの点が 1 つの解析レポートを表している．悪性挙動 DB は解析レポートから抽出した悪性挙動に情報を付加し，再定義したものを保存しているデータベースである．保存する情報は悪性挙動で使用される API の種類と引数，その悪性挙動をもつマルウェア種別一覧，

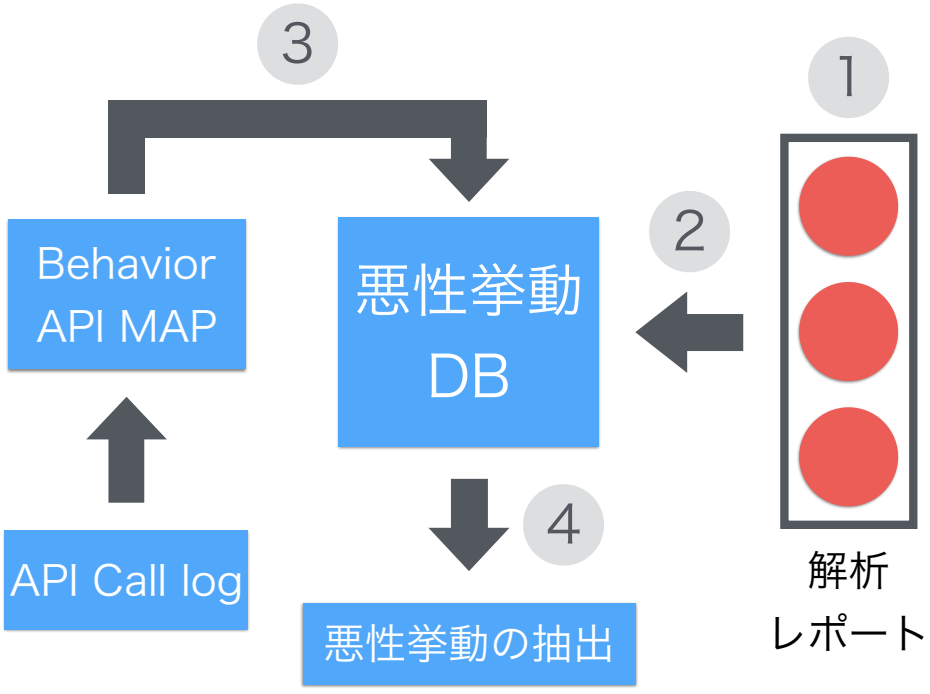

図 3.1　提案手法の概要図

悪性挙動の説明文となっている．その内，付加する情報は悪性挙動で使用される API の種類，その悪性挙動をもつマルウェア種別一覧の 2 つとなっている．DB に保存する悪性挙動の引数は悪性挙動のカテゴリによって決定される．悪性挙動のカテゴリはファイル操作系，レジストリ操作系，ネットワーク系の 3 種としている．Behavior API MAP は悪性挙動 DB に保存されている悪性挙動と実際にマルウェア検体で使用される API を対応付けるためのものである． 例えば，Co pyFile [19]，CopyFile2 [20] などの API が使用された場合，カテゴリがファイル操作系であること，コピー先のファイル名を検索に使用する引数として悪性挙動 DB に送信する．API Call log は動的解析ログの中の API の呼び出しが記載されている部分である．図 3.1 中の 1 ～ 4 についてを以下に示す．

1. 解析レポートに対して前処理を行う
2. 解析レポートから悪性挙動と挙動の説明文を抽出する
3. API コールに対応する悪性挙動を Behavior API MAP を介して悪性挙動 DB に問い合わせる
4. API コールに対応する悪性挙動が存在する場合，その悪性挙動と挙動の説明文を結果に追加する

それぞれの詳細は後述の各節で述べる．

3.2 解析レポートに対する前処理

本論文では，作成者ごとの解析レポートの差異を除去するために前処理を行う．前処理の手順を以下に示す．

1. Technical information を抽出する
2. 抽出したデータをプレーンテキストへの変換する
3. 同義語の表記の統一する
4. テキストの整形を行う

それぞれについての説明を以下に示す．

Technical information の抽出: 2.2 節で述べたように Microsoft 社の解析レポートは "Summary"，"What to do now"，"Technical information"，"Symptoms" の 4 つから構成され，"Technical information" 内で悪性挙動に関する記述がされる．そのため，本論文では Technical information のみを使用する．Technical information は HTML で記述されている解析レポート内の id が tab-link-3C で ある div タグ以下に記述されているため，この部分を抽出する．

プレーンテキストへの変換: 前述したとおり Microsoft 社の解析レポートはウェブ上で公開されているため，HTML で記述されている．HTML の状態ではレポート作成者ごとの HTML タグの使用方法による差異や HTML タグによる文字の装飾等を考慮して処理を行わなければな

らないため，処理が複雑になってしまう．本論文では処理の簡略化のために HTML 形式の解析レポートをプレーンテキストへ変換する．プレーンテキストへの変換では Block-Level Elements [21] と Inline Elements [22] といったタグの種類を考慮して再帰的に処理を行う．抽出した Technical information の子要素の先頭から順に以下の処理を適用する．以下の説明では，出力するテキストを RESULT，HTML の要素はテキスト，コメント，入れ子になっている HTML タグの 3 種の子要素から構成されているとする．

1. 要素がコメントである場合，コメントを削除する
2. 要素がテキストである場合，テキストを RESULT の最後尾に連結する
3. 要素が HTML タグである場合，その子要素の先頭から順に 2 以降の処理を行う
4. 要素が Block-level Elements，li，br のいずれかに該当する場合は RESULT の最後尾に改行文字を追加する

一般的に HTML タグの li はリスト構造を表すときに使用される．このタグを使用したテキストは実際に画面上に表示する際に自動で改行される．HTML タグを削除し，プレーンテキストへ変換する際には，リスト構造を維持するためにテキストの末尾に改行を挿入する必要がある．また，br タグは画面上に表示する際に改行を明示的に行うために使用される．li タグと同様にプレーンテキストへの変換後も画面表示時の構造を維持するために br タグを改行文字で置換する．Block-level Elements はパラグラフやリストなどのまとまりを構成する HTML タグのグループである．このグループに属するタグを使用した場合，li タグや br タグと同様に画面上に表示する際に自動で改行される．そのため，テキストの構造を維持するために末尾に改行文字を挿入する．

同義語の表記の統一: 解析レポートによっては，同じ悪性挙動について記述する場合であっても表記が異なるものがある．例えば，図 2.5 の書式 3 を "In Subkey:" で書き始めているレポートや "To Subkey:" で書き始めているレポートがある．このような差異はレポートの作成者の違いによって生じるものである．本論文では，悪性挙動の抽出作業の簡略化のために上記のような表記を統一し，書式の差異を除去することで抽出時の処理を簡略化する．統一する際に用いる同義語の意味と表記の組み合わせ一覧を表 3.1 に示す．

テキストの整形: 解析レポートは図 2.5 で示したように 4 つの書式で記述されている．その内の書式 2，3 は悪性挙動の説明文と悪性挙動に関係するリストの要素で構成されており，対象となるファイル，ディレクトリ，プロセス，通信先ドメイン，フックされる API，関連するマルウェア，レジストリキーなどが要素として記述される．そして，それらの記述の文末には悪性挙動を補足説明するような文が付与されている場合がある．補足説明では前述した抽象的表現が取りうる値の中の一例などが示される．補足説明の例を以下に示す．

表 3.1　同義語の意味と表記の組み合わせ一覧

意味	表記
Registry key	In subkey:, In subkeys:, Under subkey:, Under subkeys:, To subkey:, To subkeys:, Within subkey:, Within subkeys: The subkey:, The subkeys:
Registry entry name	Sets value:, Sets values:, Modify Registry value:, Adding the value: Adding registry value:, With value:, With values:, Under value: Under values:, Changes value:, Modifies value:, Modify value:
Registry entry value	With data:, To data:, From data:
Partition	<drive>, <drive:>, <targeted drive>, <drive root>, <drive letter>
HKEY_LOCAL_MACHINE	HKEY_LOCAL_MACHINE, HKLM
HKEY_USER	HKEY_USER, HKU
HKEY_CURRENT_USER	HKEY_CURRENT_USER, HKCU
HKEY_CLASSES_ROOT	HKEY_CLASSES_ROOT, HKCR
HKEY_CURRENT_CONFIG	HKEY_CURRENT_CONFIG, HKCC
HKEY_PERFORMANCE_DATA	HKEY_PERFORMANCE_DATA, HKPD
Space	U+0020 (通常のスペース), U+00A0 (改行禁止のスペース)
Dowble quote	半角ダブルクォート, 全角左ダブルクォート, 全角右ダブルクォート
Single quote	半角シングルクォート, 全角左シングルクォート 全角右シングルクォート
Comma	半角カンマ, 全角カンマ
Period	半角ピリオド, 全角ピリオド, <dot>

This malware is dropped by other malware as a DLL file with the following file name format:
<random characters>.cpl (for example, "kxxxacvv.cpl", "qrejtdcd.cpl")

It might also add the following registry entry to store some of its configuration data
or settings, like its path name, unique ID, and user agent string.
In subkey: HKLM\SOFTWARE\<8-digit hexadecimal number>, for example, AFB117A7
Sets value: "1"
With data: "%LOCALAPPDATA%\KB<random number>\KB<random number>.exe"

本論文ではこれらの情報を悪性挙動の抽出，検出時に使用しないため事前に削除する．補足説明文で使用される接頭辞を表 3.2 に示す．実際の処理では表 3.2 を判定する正規表現を用いることでリストの各要素から補足説明文の除去を行う．表 3.2 の接頭辞を判定する正規表現を以下に示す．

```
(,|\.)? *( ?- |\( ?for \( ?e\.g|\( ?not |such as|where |which |\( ?example|, or).*
```

表 3.2　補足説明時に使用される接頭辞一覧

接頭辞
", for example"
" (for example"
", (e.g"
", such as"
" - "
", where "
", which "
", example"

3.3　悪性挙動の抽出方法

本節では Microsoft 社の解析レポートから悪性挙動を抽出する方法を示す．2.2 節で述べたように解析レポートは図 2.5 で示した書式 1 ～ 4 のいずれかを用いて述べられている．2.2 節で述べた書式の説明の内，悪性挙動についてまとめたものを 3.3 に示す．

表 3.3　レポートの書式と記述内容

書式	記述内容
書式 1	悪性挙動全般
書式 2	ファイル操作，通信先ドメイン，ミューテックス操作， プロセス，フックされる API，ダウンロードする種別
書式 3	レジストリ操作
書式 4	スパムのタイトル，添付ファイル

このうち，書式 1 は文の構成が多様であり，全てのレポートから同じように情報を抽出するのは難しいため，本論文では使用しない．また，書式 4 で表されるスパムの情報はマルウェアの感染経路の一つであり，マルウェアの挙動ではないため，本論文では使用しない．そのため，本論文で使用する書式は書式 2 と書式 3 の 2 つとなる．また，本論文では書式 2 の内，ファイル操作，通信先ドメイン，ミューテックス 操作に関する情報のみを用いる．つまり，本論文で抽出する悪性挙動はファイル操作，レジストリ操作，通信先ドメイン，ミューテックス操作の 4 種である．こうした理由を以下に示す．

- Microsoft 社の解析レポート中の書式 2, 3 に記述されている情報はファイル，レジストリ，通信に関するものが大部分を占めていること．
- 表 2.1，2.2，2.3 で示したように API の呼び出しは registry，filesystem の 2 種が約 5 割

を占めていること．

- 動的解析ログにはマルウェアが行った通信に関する詳細な記録が保存されていること．
- 上記 4 種は抽出の自動化が比較的容易であること．

上記の 2 で示している約 5 割という数値は registry，filesystem の操作に使用された system の API である "NtClose" も考慮した数値となっている．

悪性挙動を抽出する際には，まず前処理を適用したデータを図 2.5 の書式 2，3 で示す形，つまり，説明文とリストの要素からなるブロックに分割する．ブロックには必ずリストの要素が含まれるが，説明文は含まれないこともある．これは書式 2，3 で示した説明文の書式に従わない，つまり，文末に ":" がなく，書式 1 と判別ができない説明文が存在するためである．本論文ではそれらの説明文を抽出しない．次に，ブロックに分割した悪性挙動に関する記述をファイル，レジストリ，ネットワーク，ミューテックスのカテゴリに分類し，それぞれに適した方法で悪性挙動を抽出する．

以下にそれぞれの分類，抽出方法と抽出時に適用する正規表現への変換方法を示す．

3.3.1 ファイル操作系の分類，抽出方法

ブロックの各要素がファイルパス，ファイル名である場合はファイル操作系に分類する．ファイルパス，ファイル名であるかどうかの判別には以下の基準を用いる．

- 文献 [23]，[24] で示されている環境変数で開始している文字列
- a ～ z，<targeted drive> のいずれかと ":\" で開始している文字列
- "." とファイルの拡張子で終了し，レジストリキーの接頭辞で開始しない文字列

文献 [23]，[24] で示されている環境変数の一覧を表 3.4 に示す．a ～ z，<targeted drive> のいずれかと ":\" で開始している文字列は，Windows がインストールされている C ドライブや USB ドライブなどの外部記憶装置を示すファイルパスを判別するために用いる．ファイルの拡張子は a ～ z で始まる 19,018 個のものを判定の対象にしている．19,018 個の拡張子一覧は文献 [25] を参考に作成した．19,018 個の拡張子の一部を表 3.5 に示す．

上記の基準でファイル操作系に分類した悪性挙動の抽出は以下の手順で行う．

1. アルファベットを小文字に統一する
2. ファイルパスの区切り文字 "\" でパスを分割する
3. 分割した各要素が抽象的表現を含むか確認し，含む場合は正規表現に変換する
4. 各要素が入れ子になるようにデータベースに保存する
5. 末尾の要素を保存する際には挙動の説明文のリストを "description"，挙動を持つマルウェア種別名のリストを "malware" を作成し，挙動の説明文とマルウェア種別名を保存する
6. 既に各要素がデータベースに保存されている場合は，挙動の説明文とマルウェア種別名の追加のみを行う

表 3.4　Microsoft 社の解析レポートで使用されている環境変数 [23]，[24]

表記	Windows XP, 2000 での実際のパス	Windows 10, 8.1, 7, Vista での実際のパス
%ALLUSERSPROFILE%	C:\Documents and Settings\All Users	Referred to as %ProgramData%
%APPDATA%	C:\Documents and Settings\<user name>\ Application Data	C:\Users\<user name>\AppData\Roaming
<commonappdata>	C:\Documents and Settings\All Users\ Application Data	C:\ProgramData
%CommonProgramFiles%	C:\Program Files\Common Files	C:\Program Files\Common Files
%HOMEPATH%	\Documents and Settings\<user name>	\Users\<user name>
%LOCALAPPDATA%	C:\Documents and Settings\<user name>\ Local Settings\Application Data	C:\Users\<user name>\AppData\Local
%ProgramData%	Referred to as %ALLUSERSPROFILE%	C:\ProgramData
%ProgramFiles%	C:\Program Files (x86)	C:\Program Files (x86)
%ProgramW6432%	N/A	N/A
%PUBLIC%	N/A	C:\Users\Public
$recycle.bin	Windows recycle bin (hidden location)	Windows recycle bin (hidden location)
<start menu>	C:\Documents and Settings\<user name>\ Start Menu C:\Users\<user name>\Start Menu	C:\Users\<user name>\AppData\Roaming\ Microsoft\Windows\Start Menu
<startup folder>	C:\Documents and Settings\<user name>\ Start Menu\Programs\Startup	C:\Users\<user name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
%SystemDrive%	C:	C:
<system folder>	C:\WinNT\System32	C:\Windows\System32
%SystemRoot%	C:\Windows	C:\Windows
%TEMP%	C:\DOCUME 1\<user name>\LOCALS 1\Temp	C:\Users\<user name>\AppData\Local\Temp
%USERPROFILE%	C:\Documents and Settings\<user name>	C:\Users\<user name>
%windir%	C:\Windows	C:\Windows

3.3.2　レジストリ操作系の分類，抽出方法

ブロックの各要素がレジストリに関連するものであるかどうかの判別には以下の基準を用いる．

- "In subkey:" で開始している文字列
- "Sets value:" で開始している文字列
- "With data:" で開始している文字列

3.2 節で適用した同義語の統一により，上記以外の接頭辞で開始するレジストリ操作系に関する記述は存在しない．そのため，これらの接頭辞をレジストリ操作系の判別に用いている．

レジストリ操作系の記述は，図 2.5 で示した書式の内，書式 3 はレジストリ操作に関する説明文と対象のレジストリキー，エントリ名，エントリ値から構成されている．そして，2.2 節で述べたように図 2.5 は一例であり，エントリ名やエントリ値に関する記述がないもの，1 つのレジストリキーに対して複数のエントリ名，エントリ値の組が記載されるものなど複数の記述方

法が存在する．記述方法をまとめたものを表 3.6 に示す．表で示すように，解析レポート中で出現するレジストリキー，エントリ名，エントリ値の並びは最初にレジストリキーを X 個記述した後にエントリ名，値の組を Y 個記述する場合，エントリ名，値の組を X 個記述した後にレジストリキーを Y 個記述する場合の 2 通りの記述方法に場合分けできる．また，これらの 2 通りではエントリ名，値の順序が逆になる場合があるため，計 4 通りの記述方法が存在することになる．ただし，同一リスト内であればエントリ名とエントリ値の順序は常に一定である．上記の X と Y は 0 以上の整数である．本論文で悪性挙動の抽出に用いるのは X，Y のそれぞれが 1 以上のもののみである．レジストリキーを s，エントリ名を v，エントリ値を d とした場合の抽出に用いる記述方法を表す正規表現は以下の通りである．

(s+(vd)+|s+(dv)+|(dv)+s+|(vd)+s+)

上記以外の正規表現 s+ や (dv)+，(vd)+ で表されるレジストリ関連の記述は抽出に用いない．上記の正規表現で選定した悪性挙動の抽出は以下の手順で行う．

1. アルファベットを小文字に統一する
2. レジストリキーの区切り文字 "\" で分割する
3. 分割した各要素が抽象的表現を含むか確認し，含む場合は正規表現に変換する
4. 各要素が入れ子になるようにデータベースに保存する
5. 末尾の要素を保存する際には挙動の説明文のリストを "description"，挙動を持つマルウェア種別名のリストを "malware" を作成し，挙動の説明文とマルウェア種別名を保存する
6. 既に各要素がデータベースに保存されている場合は，挙動の説明文とマルウェア種別名の追加のみを行う

3.3.3 ネットワーク操作系の分類，抽出方法

ブロックの各要素がネットワーク操作系であるかどうかの判別には以下の基準を用いる．

- "/" で区切った先頭の要素が IP アドレスである文字列
- "/" で区切った先頭の要素が "." とトップレベルドメインで終わる文字列

IP アドレスかどうかの判定には以下の正規表現を用いる．

[0-9]{1,3}(.[0-9]{1,3}){3})

トップレベルドメインの判定には表 3.7 に示すものを使用する．表 3.7 は文献 [**?**] を参考にして作成した．

上記の基準で選定した悪性挙動の抽出は以下の手順で行う．

1. アルファベットを小文字に統一する
2. "https://"，"http://" が先頭に存在する場合は削除する

3. 分割した各要素が抽象的表現を含むか確認し，含む場合は正規表現に変換する
4. 最も先頭に近い "/" で文字列を分割し，先頭の要素と末尾の要素を入れ子になるようにデータベースに保存する
5. 末尾の要素を保存する際には挙動の説明文のリストを "description"，挙動を持つマルウェア種別名のリストを "malware" を作成し，挙動の説明文とマルウェア種別名を保存する
6. 既に各要素がデータベースに保存されている場合は，挙動の説明文とマルウェア種別名の追加のみを行う

3.3.4 ミューテックス操作系の分類，抽出方法

Microsoft 社の解析レポートでは，レジストリにおける "In subkey:"，"Sets value" のようなミューテックスを指し示す特定の語句は用いられていない．また，ミューテックスではファイルパス (%windir%，%appdata%) やレジストリ (hklm，hkcr)，URL (http://，https://) のようにいくつかの決まった接頭辞で開始することもない．そのため，ブロックの各要素の判別には悪性挙動の説明文を用いる．各要素がミューテックス操作系であると判定する条件を以下に示す．

- 悪性挙動の説明文中に文字列 "mutex" が出現する

悪性挙動を抽出する際には上記 3 種のような処理は適用せず，ブロックの各要素をそのままデータベースに保存する．挙動の説明文のリストとマルウェア種別のリストに関しては上記 3 種と同様に保存する．

3.3.5 正規表現への変換方法

上記 3 種の悪性挙動内に存在する抽象的表現は以下の書式に従っている．

{文字数? (random)? 文字列の型?} | <文字数? (random)? 文字列の型?>

抽象的表現は "<>"，"" で囲まれた文字列で表される．文字数では "1"，"2"，"one"，"two" などの数字や英単語によって指定される．また，文字数が固定ではなく範囲で指定される場合は "1-4" のように "-" が使用される．文字列の型では任意の文字列，10 進数，16 進数，空白，年月日，ポート番号，ドメイン，ハードディスクのシリアル番号，アフィリエイト ID のいずれかが指定される．抽象的表現では文字数や型が指定されずに単に "<random>" と表記される場合もある．これらの規則を利用して抽象的表現を正規表現に変換する方法を以下に示す．

1. "{}"，"<>" で囲まれた抽象的表現を探す
2. 文字数が指定されている場合，"{}" を使用して正規表現の文字数を設定する
3. 文字数が指定されていない場合の文字数は "+" (1 文字以上) とする
4. 文字列の型が指定されている場合，型に適した正規表現を設定する
5. 文字列の型が指定されていない場合，任意の文字列を設定する

6. "{}"，"<>" で囲まれた抽象的表現を正規表現に置き換える

文字列の型によっては "+" 以外の文字数を指定する場合もある．任意の文字列を，ファイルパス，レジストリキーの場合は "[^<>"|?*\\]"，それ以外の場合は "[^<>"|?]" としている．解析レポート内で使用される文字列の型の抜粋を表 3.8 に示す．表 3.8 の年月日に関しては yyyy，mm，dd が順不同で出現することがわかっている．また，各要素の間に "_" や "-" などが挿入されている場合がある．yyyy を "(1|2)[0-9]{3}，それ以外を "[0-9]{2}" にそれぞれ置換することで上記の場合に対応している．

3.3.6 悪性挙動の検出

本節では動的解析ログから悪性挙動を検出する方法を示す．動的解析ログには 2.1 節で述べたように，データセット内には既知検体と未知検体があるが，ここでは全ての検体に対して検出を行うこととする．検出する際にはファイル操作系，レジストリ操作系，ミューテックス操作系は API コールログを，ネットワーク系は動的解析ログ中の図 2.3 で示した項目を調べる．

ネットワーク系悪性挙動の検出では悪性挙動 DB に保存されているドメイン，IP アドレスが図 2.3 で示す "hosts"，"domains" に含まれているかを調べる．本論文では通信する目的やドメインが悪性なものであるかは調べない．

レジストリ操作系悪性挙動では，実際に特定のレジストリキー，エントリ名に対してエントリ値の変更を行った場合だけでなく，現在のエントリ値の確認を行った場合も検出している．これは既に設定を変更済みの環境ではエントリ値の変更を行う必要がなく，確認だけが行われると考えたためである．

ファイル操作系，レジストリ操作系，ミューテックス操作系の検出には Behavior API MAP を用いる．Behavior API MAP で結びつけている カテゴリと API を表 3.9 に示す．悪性挙動の検出時は，呼び出された API を使った悪性挙動がないかを Behavior API MAP を介して悪性挙動 DB に問い合わせる．問い合わせ時に必要な情報がカテゴリごとに存在する．その必要な情報の一覧を表 3.10 に示す．表 3.10 に示されている情報は API の引数として与えられる場合がほとんどである．例外として，reg_set_api，reg_query_api，reg_query_key_api で必要となるレジストリキーだけは引数で与えられない．レジストリ操作系の API は以下の手順で処理を行う．

1. reg_open_api (reg_create_api) で指定したレジストリキーを開く
2. reg_set_api (reg_query_api，reg_query_key_api) は引数で与えられた 1 の API で得たハンドル，エントリ名，エントリ値を用いて処理を行う
3. close_handle_api でレジストリキーを閉じる

レジストリキーが明示的に引数として与えられるのは reg_open_api と reg_create_api だけである．それ以外のレジストリ操作系の API は reg_open_api と reg_create_api の実行時に得られたハンドルを介してレジストリキーへの処理を行う．また， reg_open_api や reg_create_api が入れ子で使用されることがある．この場合，それぞれが開いたレジストリキーを連結したもの

がレジストリキーのフルパスとなる．このレジストリキーのフルパスとハンドルの組を保存しておくことで reg_set_api，reg_query_api，reg_query_key_api で必要となるレジストリキーを特定する．

3.3.7 検出結果の出力

本節では解析レポートの草案の自動生成する方法について述べる．本論文における解析レポートの草案は，3.3.6 節で検出した悪性挙動に関する情報をまとめ，Markdown 形式で出力したものを指す．草案には，検査したマルウェアのファイル名，検査したマルウェアの種別名，その種別がもつ悪性挙動一覧，実際に検出した悪性挙動一覧，それらの挙動を持つマルウェア一覧，それらの悪性挙動の検出に用いた正規表現のパターン一覧を出力する．

生成する手順を以下に示す．

1. マルウェア検体の種別名を取得する
2. 検出された悪性挙動の内，同じ正規表現のパターンによって検出された挙動をまとめる
3. それぞれのパターンが持つ説明文のリストから最適なものを 1 つ選択する
4. それぞれのパターンの説明文，検出された挙動一覧，そのパターンを持つマルウェア種別一覧を列挙する
5. マルウェアが既知の検体である場合，その種別がもつ悪性挙動の一覧を列挙する

上記の説明文の選択方法の基準を以下に示す．

- 既知検体である場合，自身の種別から抽出した説明文を選択
- 自身の説明文が複数存在する場合，説明文の文字数が最も少ないものを選択する
- 自身の種別から抽出した説明文が存在しない場合，説明文の文字数が最も少ないものを選択する

表 3.5　ファイルの拡張子の一部

拡張子
a, a$v, aa, aaa, aab, aac, aad, aae, aaf, aai, aam, aao, aap, aapkg, aar, aas, aat, aatrend, aaui, aaw, aawdef, aax, ab, ab$, aba, abap, abb
abbu, abc, abcd, abcddb, abcdg, abcdmr, abcdp, abd, abdata, abe, abf, abg, abi, abicollab, abk, abkprj, abl, abm, abn, abo, abp, abr, abs, b, b!k
b&w, ba, ba_, bab, babl, babyluvsavedgame, bac, back, backpack, backup, backupdb, bad, bada, badongo, baf, bafl, bag, bah, bai, bak, bakx, bak
bal, ballhallasavedgame, bam, bamboopaper, baml, bamp, ban, banana, band, bank, bap, bar, baroc, bas, basebinary, baseconfig, baseproj, basex
bash, bash_history, bash_login, bash_logout, bash_profile, bashrc, basin, bat, c, c#, c++, c–, c-map, c_, c__, ca, ca_, caa, cab, cabal
cac, caccrt, cache, cachedump, caction, cad, cadc, cadpac, cadrg, cae, caf, caff, cag, cah, cai, caj, cak, cakemaniasavedgame, cakewalkstudioware
cakewalkwindowlayout, cal, calb, calca, calendarlocks, calib, calibre, calibz, cals, calx, cam, cam-ft, cam-gmc, cam-pkg, camelsounds, caml
camm, camp, camproj, d, da, da$, da_, daa, daap, dab, dac, dacpac, dad, dadx, dae, daemonscript, daf, dag, dah, dai, dal, dam, dan, dao, dap
dapf, daproj, daq, dar, dart, dartclip, das, daschema, dash, dat, dat-shm, dat-wal, dat_mcr, dat_new, dat_old, dat_tureg_old, data, database
datacontract, datasource, datx, dau, dav, daw, dawg, dax, day, daz, e, ea, eac, eaf, eaglerc, eal, eam, eap, ear, eas, easm, easmx, eax, eaz, eb
eba, ebaml, ebax, ebb, ebc, ebcdic, ebd, ebdl, ebev, ebf, ebh, ebi, ebj, ebk, ebkproj, ebktml, ebl, ebm, ebmd, ebmp, ebn, ebo, ebp, ebq, ebr
ebs, ebu, ebuild, ebv, ebw, ebx, ebz, ec_, ecab, ecb, f, f#+a, f#+d, f+db, fa, fab, fabbproject, fabpfpkg, fac, face, fact, factorypath, fad
fadein, fader, fae, faff, fag, fah, failed, failed-conv, failurerequests, fairytreasuresavedgame, fak, fal, fam, familyx, familyxs, fan, fap, faq
far, farrun, fas, fasl, fasta, fastresume, fat, fatesavedgame, fateundiscoveredrealmssavedgame, fau, fav, favorite, favoritemetadata, fax, fay
faz, fb, fb!, fba, g, gab, gac, gad, gadgeprj, gadget, gae, gaf, gai, gal, galapagosavedgame, galaxy, gallery, gallerycollection, galleryitem, galsave
gam, gambas, game, gamedata, gameproj, gamout, gan, gap, gar, garmin, gas, gasx, gat, gau, gax, gb, gba, gba.ds, gbap, gbaskin, gbb, gbc, gbck
gbcskin, gbd, gbf, gbh, gbhs, gbi, gbk, gbl, gblorb, gbm, gbn, h, h!, h++, h–, h_, h__, ha, haf, hairy, hak, hal, hald, halog, ham, hamachi
haml, han, handlebars, hang, hap, happyfish, har, harddisk, harx, has, hash, hashes, hat, hathdl, hav, hax, hay, haz, hb, hbc, hbe, hbf, hbi, hbin
hbk, hbl, hbm, hbn, hbp, hbr, hbs, hbx, hby, hbz, hc, i, ia, iaa, iab, iad, iaf, iai, iak, ial, iam, ian, iao, iap, iar, ias, iascii, iax, ib, ib_mon
iba, ibadr, ibak, ibank, ibatemplate, ibb, ibc, ibcd, ibch, ibd, ibdat, ibe, ibf, ibg, ibi, ibk, ibl, ibm, ibn, ibo, ibook, ibooks, ibp, ibplugin
ibq, ibr, ibre, ibro, ibs, ibt, ibv, j, ja, jaas, jab, jacksum, jacl, jad, jade, jaf, jag, jai, jak, jam, jammix, jan, jap, jar, jar.pack, jardesc
jas, jascproject, jasper, jav, java, javajet, jaw, jax, jbc, jbf, jbg, jbi, jbig, jbk, jbl, jbm, jbr, jbs, jbt, jbw, jbx, jbz, jc, jc!, jc_, jcb
jcc, jcd, jcf, jck, jcl, k, ka, kab, kac, kad, kaf, kahl, kal, kam, kan, kap, kar, kas, kat, kb, kbb, kbcf, kbcls, kbd, kbed, kbf, kbgi, kbi, kbl
kbm, kbn, kbp, kbr, kbrf, kbs, kbt, kbtf, kburl, kbx, kcd, kces, kcey, kcf, kch, kci, kcl, kcm, kct, kda, kdb, kdbx, kdc, kdd, kde, kdelnk, l, la
la_, laa, lab, label, labels, labx, laccdb, lad, lae, laf, lai, lak, lam, lamp, lan, lanalyser, land, landsat, lang, langz, lansat, lao, lap, lar
las, lasso, last, lastbuildstate, lastlogin, lat, latex, launch, launchd, launcherex_sourcerecord, lav, lavs, law, lax, lay, layerdiagram, layout
layoutdesigner, laz, lb, lba, lbc, lbd, lbe, m, m$s, m-jpeg, m??, m_p, ma, ma_, maa, mab, mac, maca, macbin, macdraw, macmol, macp, macs
mad, mae, maf, maff, mag, magic, magik, magnet, mahjongtitanssave-ms, mai, mail, mailablistview, mailabview, maildb, mailhost, mailloc
mailstationery, mailtoloc, mailview, main, maj, mak, make, maker, maki, mako, mal, mam, maml, man, managed_manifest, mani, manifest
manifest-desktop, n, n-gage, nab, nabs, nacl, nad, naf, nai, nak, nal, nam, namcan, nameonclipboard, nan, nanflmrkxtns, nanokey_data, nanopad_glob
nanr, nap, napj, naplps, napr, napt, nar, narc, narrative, nas, nat, nav, navionics, navmap, naz, nb, nba, nbak, nbc, nbd, nbe, nbf, nbfc, nbfp
nbfs, nbgr, nbh, nbi, nbib, nbin, nbk, nbkt, nbl, o, oa, oab, oac, oad, oaf, oam, oap, oar, oas, oat, oav, oaw, oaz, ob, ob!, ob_, obak, obb, obc
obd, obf, obg, obi, obj, objectcache, objx, obk, obl, obm, obml, obp, obpack, obr, obs, obt, obv, obw, obx, oby, obyx, obz, oc_, oca, ocamlmakefile
ocb, occ, ocd, ocdc, ocdf, p, pa, paa, paal, pab, pac, pack, pack.gz, package, pact, pad, padl, pae, paf, paf.exe, pag, page, page.security, pages
pages-tef, pai, paint, paj, pak, pal, palm, paltalk, pam, pamp, pan, pando, pandora, panic, panl, pano, pao, pap, papa, papers, paq, par, param
parm, part, partial, partimg, partimg.gz, partitions, pas, pat, q&a, qa, qaa, qac, qad, qaf, qag, qal, qam, qap, qar, qat, qb, qba, qbb, qbd
qbe, qbf, qbi, qbk, qbl, qbm, qbmb, qbmd, qbo, qbp, qbquery, qbr, qbs, qbt, qbw, qbx, qby, qbz, qc, qc-emod, qca, qcall, qcc, qcd, qcf, qch, qci
qcif, qck, qcm, qcn, qcow, qcp, qcs, r, r*ch, ra, ra_, raa, rab, rac, rad, radius, rae, raes, raf, rag, rai, rak, rake, rakefile, ral, ram, ramd
ran, rao, rap, rapc, rar, rarenx, rargb, rarx, ras, raskinlicense, raskinplace, rast, rat, ratDVD, rav, ravi, raw, rawa, rawpkt, rax, ray, razersynapse
rb, rba, rbb, rbc, rbd, rbdf, rbf, rbg, s, s#+a, sa_, saa, saas, sab, sabl, sabs, sacd, sad, saf, safariextz, safe, sag, saga, sagem, sah, sai
saif, saj, sak, sal, sam, sami, sample, samrc, san, sap, sapssd, sar, sas, sasf, sass, sat, sav, save, save_multiplayer, saved, saveddeck, savedsearch
saver, savf, saw, sax, say, saz, sb, sb_, sba, sbb, t, tab, tm, ta_, taac, taact, taasmt, tab, tableContent, tabletprefs, tabletprâĂŃefs, tabula-buttons
tabula-doc, tabula-docstyle, tac, tacls, tacmpt, tacont, tad, taf, tag, tags, taguit, tah, tai, tajima, tak, tal, talf, talk, tam, tan, tao, tap
tapesrcmdl, tar, tar-gz, tar-lzma, tar-z, tar.gz, tar.xz, tar.z, taraa, tarc, tarct, tardist, targa, targets, tas, task, u, uae, ual, uap, uaq
uas, uav, uax, ub, ubb, ubd, ubf, ubi, ubifs, ubj, ubk, ubl, ubm, ubot, ubox, ubr, ubs, ubx, uc, uc_, ucb, uccapilog, ucd, uce, ucf, uci, ucl, ucls
ucm, ucn, ucp, ucr, ucs, ucsettings, uct, ucv, ucx, ud, uda, udb, udc, udcx, udeb, udf, udg, v, vExporter, vShaderBin, vab, vac, vad, vaf, vag, vai
val, vala, valu, vam, vap, vaporcd, vapplauncher, var, vas, vat, vault, vb, vb_, vb_aq, vba, vbadoc, vbak, vbb, vbc, vbd, vbdproj, vbe, vbe_aq
vbee, vbf, vbg, vbhtml, vbi, vbk, vbl, vbm, vbn, vbo, vbox, vbox-extpack, vbox-prev, vboxlm, vboxsave, vbp, vbproj, vbr, w, w??, wa, wa!, wa_
waa, wab, wab , wac, wacomprefs, wacomxs, wad, wadcfg, wae, waf, waff, wag, wagame, wai, waj, wal, wam, wand, wap, wapt, war, warc, warning
warz, was, wat, wav, wavc, wave, wavm, wax, wa , wb, wb?, wb_, wba, wbb, wbc, wbcat, wbd, wbdp, wbf, wbfs, wbk, wbl, x, x-ms-wma, x-ms-wmv
x-musepack, x-png, x_b, x_n, x_t, xa, xaa, xab, xac, xad, xadd, xadml, xaf, xaiml, xam, xaml, xamlx, xan, xann, xap, xapk, xappl, xar, xarc
xas, xav, xavc, xba, xbap, xbb, xbc, xbd, xbe, xbel, xbf, xbin, xbk, xbl, xblr, xbm, xbml, xbmml, xbmp, xbn, xbrl, xbsav, xbt, y, yab, yacc, yadpcm
yaf, yah, yajl, yak, yal, yam, yaml, yaodl, yar, yast, yaws, ybc, ybhtm, ybk, ybm, ybrar, ybrtf, ybtxt, ycbcr, ycbcra, ycc, ychat, ycm, ycp, yct
ydb, ydc, ydd, ydk, ydl, ydr, ydsp, ydt, yel, yenc, yep, yes, yesterday, yfs, ygf, ygn, ygo, yif, yify, yka, ym, z, za, zaa, zab, zabw, zac, zad
zae, zaf, zak, zaloha, zam, zan, zanebug, zap, zapc, zapd, zar, zargo, zasm, zat, zave, zax, zba, zbb, zbd, zbf, zbi, zbin, zblorb, zbm, zbp, zbr
zbs, zbx, zc, zca, zcache, zcast, zcc, zce, zcfg, zch, zcls, zcn, zcp, zcpml, zct, zcv, zdat

表 3.6　レジストリ操作系悪性挙動の記述方法

記述方法 1	記述方法 2
In subkey: regkey1	Sets value: value1
...	With data: data1
In subkey: regkeyX	...
Sets value: value1	Sets value: valueX
With data: data1	With data: dataX
...	In subkey: regkey1
Sets value: valueY	...
With data: dataY	In subkey: regkeyY

表 3.7　トップレベルドメイン一覧 [26]

トップレベルドメイン
aero, arpa, asia, biz, cat, com, coop, edu, gov, info, int, jobs, xyz, technology, mil, mobi, museum, name, rocks, red
blue, net, org, pro, rich, tel, travel, wiki, ac, ad, ae, af, ag, ai, al, am, an, ao, aq, ar, as, at, au, aw, ax, az, ba
bb, bd, be, bf, bg, bh, bi, bj, bm, bn, bo, br, bs, bt, bv, bw, by, bz, ca, cc, cd, cf, cg, ch, ci, ck, cl, cm, cn, co
cr, cs, cu, cv, cx, cy, cz, dd, de, dj, dk, dm, do, dz, ec, ee, eg, eh, er, es, et, eu, fi, fj, fk, fm, fo, fr, ga, gb
gd, ge, gf, gg, gh, gi, gl, gm, gn, gp, gq, gr, gs, gt, gu, gw, gy, hk, hm, hn, hr, ht, hu, id, ie, il, im, in, io, iq
ir, is, it, je, jm, jo, jp, ke, kg, kh, ki, km, kn, kp, kr, kw, ky, kz, la, lb, lc, li, lk, lr, ls, lt, lu, lv, ly, ma
mc, md, me, mg, mh, mk, ml, mm, mn, mo, mp, mq, mr, ms, mt, mu, mv, mw, mx, my, mz, na, nc, ne, nf, ng, ni
nl, no, np, nr, nu, nz, om, pa, pe, pf, pg, ph, pk, pl, pm, pn, pr, ps, pt, pw, py, qa, re, ro, rs, ru, rw, sa, sb, sc
sd, se, sg, sh, si, sj, sk, sl, sm, sn, so, sr, st, su, sv, sy, sz, tc, td, tf, tg, th, tj, tk, tl, tm, tn, to, tp, tr
tt, tv, tw, tz, ua, ug, uk, um, us, uy, uz, va, vc, ve, vg, vi, vn, vu, wf, ws, ye, yt, yu, za, zm, zr, zw

表 3.8　抽象的表現で指定される文字列の型の抜粋

意味	表記	正規表現
16 進数	hexadecimal，digit from 0 to f，digit from 0 -> f	[a-fA-F0-9]
10 進数	number，num，digit	[0-9]
英数字	alphanumerical	[a-zA-Z0-9]
空白	many spaces	\s
セキュリティ識別子	sid	S-1-[0-9]+(-[0-9]+)+
ポート番号	port	[0-9]{1,5}
ドメイン	domain	[a-zA-Z0-9]
アフィリエイト ID	affiliate id	[a-zA-Z0-9][a-zA-Z0-9\.\-]+
ハードディスクのシリアル番号	hd serial	[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}
年月日	yyyymmdd	(1|2)[0-9]3[0-9]2[0-9]2
任意の文字列	letter，name，string，char，folder file，value，parameter，location，path	[a-zA-Z0-9]

表 3.9　Behavior API MAP

カテゴリ	API
close_handle_api	RegCloseKey, NtClose
reg_open_api	RegOpenKeyExA, RegOpenKeyExW, NtOpenKey, NtOpenKeyEx
reg_create_api	RegCreateKeyExA, RegCreateKeyExW, NtCreateKey
reg_enumerate_api	NtEnumerateKey
reg_set_api	RegSetValueExA, RegSetValueExW, NtSetValueKey
reg_query_key_api	NtQueryKey
reg_query_api	RegQueryValueExA, RegQueryValueExW, RegQueryInfoKeyExA, RegQueryInfoKeyExW, NtQueryValueKey
reg_del_api	RegDeleteKeyA, RegDeleteKeyW, RegDeleteValueA, RegDeleteValueW, NtDeleteKey, NtDeleteValueKey
open_file_api	NtOpenFile
create_file_api	NtCreateFile, CreateFile, CreateFileA, CreateFileW, CreateFile2, CreateFileTransacted
copy_file_api	CopyFile, CopyFile2, CopyFileA, CopyFileW, CopyFileExA, CopyFileExW
create_dir_api	CreateDirectory, CreateDirectoryA, CreateDirectoryW, CreateDirectoryEx, CreateDirectoryExA, CreateDirectoryExW
mutex_api	NtCreateMutant, NtOpenMutant

表 3.10　悪性挙動の検索時に必要な情報

カテゴリ	必要となる情報
copy_file_api	コピー先のファイル名
open_file_api	開くファイル名
create_file_api	作成するファイル名
create_dir_api	作成するディレクトリ名
reg_open_api	レジストリキー
reg_create_api	レジストリキー
reg_set_api	キー，エントリ名，エントリ値
reg_query_api	キー，エントリ名
mutex_api	ミューテックス名
上記以外	悪性挙動の検索を行わない

第 4 章　結果

本章では解析レポートから抽出した悪性挙動，動的解析ログから検出した悪性挙動，自動生成した解析レポートの草案を示す．

4.1　解析レポートから抽出した悪性挙動について

解析レポートから抽出した悪性挙動数を表 4.1 に示す．

表 4.1　抽象的表現で指定される文字列の型の抜粋

悪性挙動の種類	抽出した悪性挙動数
ファイル操作系	4,416
レジストリ操作系	1,100
ネットワーク系	3,505
ミューテックス操作系	156

表中の数値は正規表現とそれ以外を区別している．例えば，"%allusersprofile%\hoge.exe"，"%allusersprofile%\[<>Ï?*\\]+.exe"，"%allusersprofile%\[<>Ï?*\\]+" のような 3 つの悪性挙動が悪性挙動 DB 内に保存されていた場合，その全てが "%allusersprofile%\hoge.exe" と一致するが，それぞれを異なる悪性挙動として数えている．実際に抽出した悪性挙動の抜粋を図 A.1，図 A.2，図 A.3，図 A.4 のそれぞれに示す．図では子要素を持つ場合は "children"，末尾の要素の場合は "behavior" の以下に情報を保存している．"behavior" 以下には 3.3 節で述べたように悪性挙動の説明文のリスト "description" とその挙動をもつマルウェア種別のリスト "malware" を保存する．"children"，"behavior" の違いにかかわらず正規表現を持つ要素は "REGEXP" 以下に保存する．これは正規表現とそれ以外を区別し，悪性挙動の検出処理の効率化を図るためである．

4.2　動的解析ログから検出した悪性挙動について

データセットの各検体から検出できた悪性挙動数の CDF を図 4.1 に示す．この図の縦軸は全検体に対する割合を，横軸は 1 検体に含まれる悪性挙動数，赤い破線は平均値を表している．

対象となるのは 8,640 検体で，検出された悪性挙動数の最小値は 0，最大値は 836，平均値は 8.37 であった．図 4.1 における曲線の傾きは $0 \leq x \leq 100$ の範囲で大きく変化していることがわかる．図 4.1 の $0 \leq x \leq 100$ の範囲を抽出したものを図 4.2 に示す．図 4.2 から，全検体の 8 割から悪性挙動を抽出できていることがわかる．次に，各データセットから検出できた悪性挙動数の CDF を図 4.3，図 4.4，図 4.5 に示す．そして，それらの最小値，最大値，平均値，悪性挙動を検出できた検体の割合をまとめたものを表 4.2 に示す．

表 4.2　各 FFRI Dataset における悪性挙動の検出数

データセット	最小値	最大値	平均値	悪性挙動を検出できた検体の割合
FFRI Dataset 2013	0	59	5.93	95.87%
FFRI Dataset 2014	1	836	16.16	100%
FFRI Dataset 2015	0	315	2.74	47.77%

図と表からわかるように FFRI Dataset 2013 と 2014 ではほぼ全ての検体から悪性挙動を検出できているのに対し，FFRI Dataset 2015 では 25 % の検体から悪性挙動を検出できていない．つまり，図 4.2 で悪性挙動を検出できなかった 1 割の検体のほとんどが FFRI Dataset 2015 に属するものであることがわかる．以下ではその要因について考察する．

まず，Behavior API MAP が対応している API の呼び出し回数について考える．各データセットにおける API 呼び出しの回数をまとめたものを表 4.3 に示す．この表からわかるように FFRI Dataset 2014 の API 呼び出し回数が最も多く，続いて FFRI Dataset 2015，FFRI Dataset 2013 の順に少なくなっている．表では全 API 呼び出し，BAM が対応している API 呼び出しのどちらにおいても上記の順位になっている．悪性挙動の検出に用いる情報量の大きさから，検出数に関しても同様の順位になるのが妥当であると考えられる．しかし，FFRI Dataset 2015 の検出数は他の 2 つに比べ，大きく劣っている．このことから FFRI Dataset 2015 の検出数が低い要因は API の呼び出し回数に起因するものではないと考えられる．

表 4.3　各データセットにおける API 呼び出し回数

データセット	全 API の呼び出し回数	BAM が対応している API の呼び出し回数
FFRI Dataset 2013	1,703,263	322,703
FFRI Dataset 2014	34,978,215	7,482,885
FFRI Dataset 2015	3,536,765	1,246,389

次に，FFRI Dataset 2015 において悪性挙動を検出できなかったマルウェア種別について考える．FFRI Dataset 2015 において悪性挙動を検出できなかったマルウェア種別を表 4.4 にまとめる．表では検体のマルウェア種別，もしくは原種の解析レポートから抽出した悪性挙動数を正

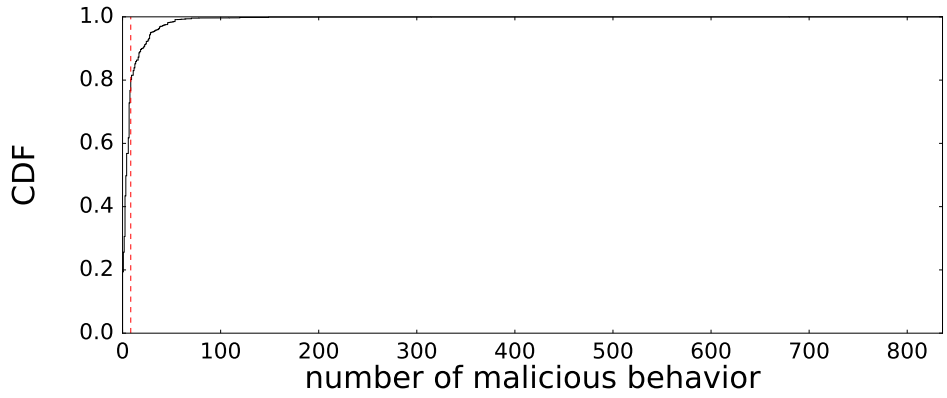

図 4.1　1 検体に含まれる悪性挙動数の CDF

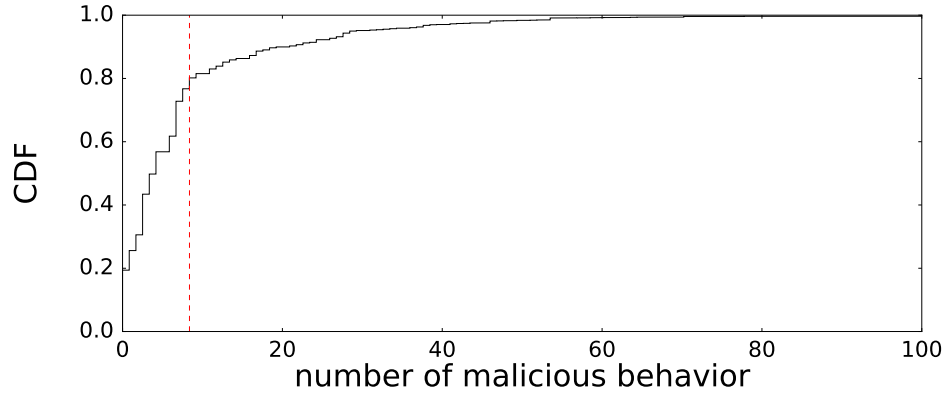

図 4.2　1 検体に含まれる悪性挙動数の CDF ($0 \leq x \leq 100$)

解挙動数としている．また，FFRI Dataset 2015 に固有なマルウェア種別の場合には True，固有でない場合は False としている．表から種別不明の検体が最も多いこと，悪性挙動を抽出できていない種別が存在すること，多くの種別において FFRI Dataset 2015 に固有であることがわかる．この表から考えられる FFRI Dataset 2015 の悪性挙動検体数が低い要因を以下に示す．

1. FFRI Dataset 2015 内の種別不明検体の多くが新規のマルウェアであり，解析レポートに含まれる悪性挙動を持たない
2. FFRI Dataset 2015 に含まれるマルウェア種別において悪性挙動の抽出精度が低く，悪性挙動を検出できていない
3. FFRI Dataset 2015 の検体の多くが本論文で対象にしたファイル操作系，レジストリ操作系，ネットワーク系，ミューテックス操作系以外の悪性挙動を持つ
4. 動的解析に対する対抗策が適用されており，悪性挙動と関係のない API 呼び出しが多く含まれている

表 4.4　FFRI Dataset 2015 において悪性挙動を検出できなかったマルウェア種別上位 30 に関するまとめ

種別名	検体数 (割合)	正解挙動数	FFRI Dataset 2015 に固有
種別不明	551 (35.16 %)	0	不明
virtool:win32/ceeinject.gen!kk	106 (6.76 %)	0	False
pws:win32/zbot	69 (4.40 %)	33	False
virtool:win32/obfuscator.wt	66 (4.21 %)	0	False
trojan:win32/dynamer!ac	58 (3.70 %)	0	False
pws:win32/zbot.gen!vm	38 (2.43 %)	29	True
trojan:win32/miuref.f	33 (2.11 %)	3	True
trojandownloader:win32/small.gen!i	26 (1.66 %)	0	True
pws:win32/fareit	23 (1.47 %)	4	False
worm:win32/kasidet.b	18 (1.15 %)	9	True
trojandropper:win32/bunitu.c	15 (0.96 %)	33	True
ransom:win32/critroni.a	14 (0.89 %)	2	True
backdoor:win32/kelihos!rfn	14 (0.89 %)	0	True
trojan:win32/chanitor.a	14 (0.89 %)	4	True
worm:win32/gamarue	13 (0.83 %)	13	False
trojan:win32/ropest.j	13 (0.83 %)	0	True
trojan:win32/neurevt	12 (0.77 %)	23	True
rogue:win32/trapwot	12 (0.77 %)	7	True
trojanspy:msil/golroted.b	12 (0.77 %)	2	True
trojan:win32/enchanim.gen!a	11 (0.70 %)	0	True
backdoor:win32/kelihos	10 (0.64 %)	7	True
trojan:win32/fleercivet.d	10 (0.64 %)	1	True
virtool:win32/autinject.bl	10 (0.64 %)	0	True
ransom:win32/critroni	9 (0.57 %)	2	True
trojan:win32/peals.b!gfc	9 (0.57 %)	0	True
pws:win32/zbot!rfn	9 (0.57 %)	0	True
trojandownloader:win32/dalexis.c	8 (0.51 %)	5	True
trojandownloader:win32/upatre	8 (0.51 %)	3	True
pws:win32/zbot.gen!ci	7 (0.45 %)	29	True

4.2.1　正解に対するカバレッジについて

各データセットの検体から検出した悪性挙動の正解に対するカバー率 (以下，カバレッジと記す) を図 4.6，4.7，4.8 に示す．図は全検体が持つ正解挙動数の平均値である 3.97 以上の正解挙動をもつマルウェア種別に対して算出したものである．検体が属するマルウェア種別と原種から抽出した悪性挙動数を X，その内検出できた悪性挙動数を Y としたときのカバレッジの計算方法を以下に示す．

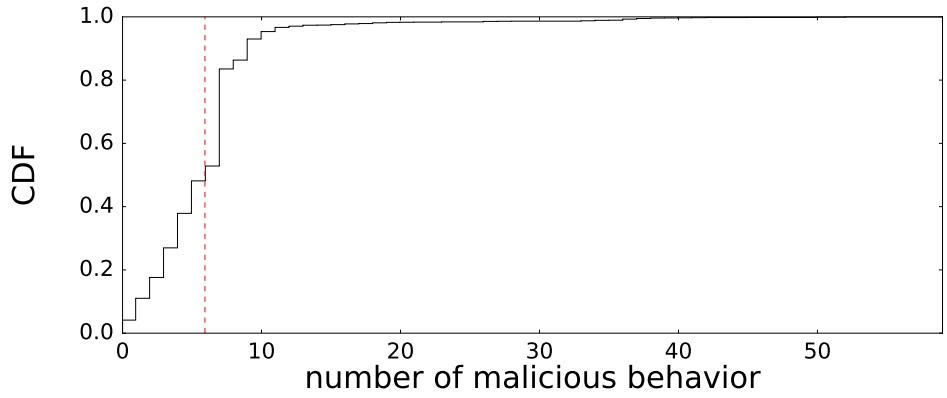

図 4.3　1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2013)

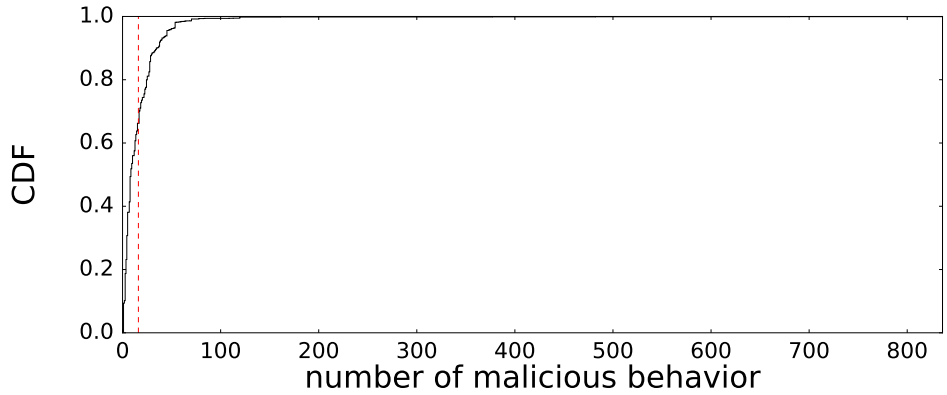

図 4.4　1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2014)

$$\boxed{\text{カバレッジ} = Y \div X}$$

上記の図における最小値，最大値，平均値をまとめたものを表 4.5 に示す．表の数値からわかるようにカバレッジの最大値は 1 ではない．これは，抽出した悪性挙動が正規表現を含む場合，正規表現に一致する挙動のそれぞれを 1 つの悪性挙動として数えているためである．

表 4.5　各データセットにおけるカバレッジ

データセット	最小値	最大値	平均値
FFRI Dataset 2013	0	3.25	0.04636
FFRI Dataset 2014	0	63.33	0.4096
FFRI Dataset 2015	0	2.44	0.1560

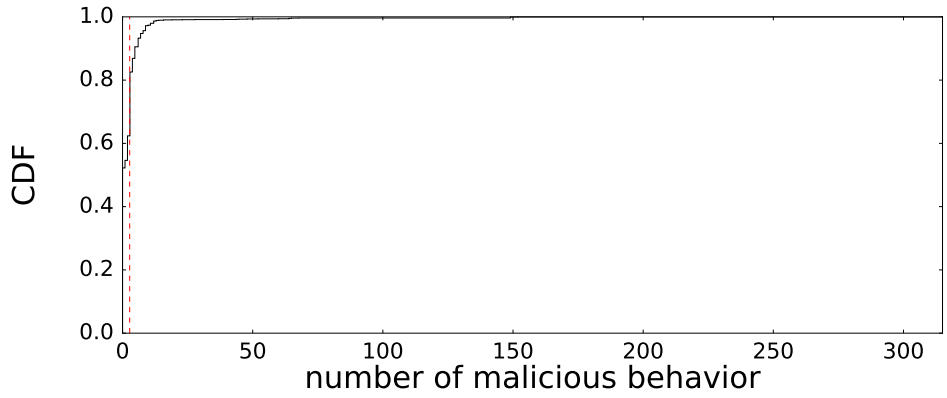

図 4.5　1 検体に含まれる悪性挙動数の CDF (FFRI Dataset 2015)

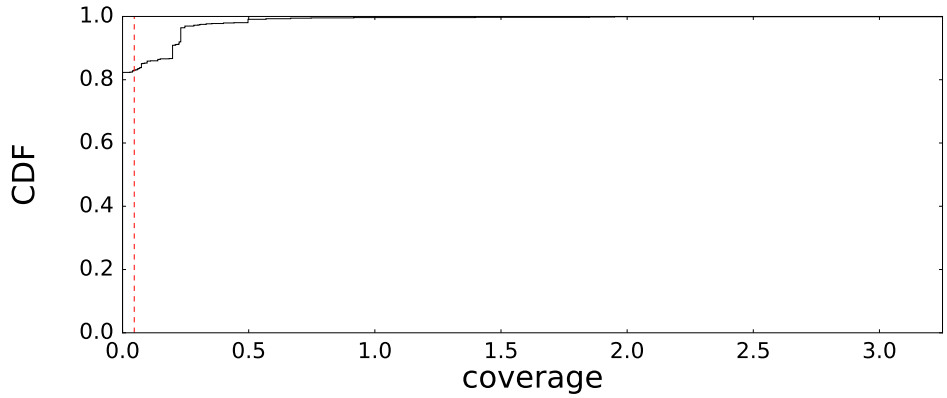

図 4.6　正解レポートに対するカバレッジの CDF (FFRI Dataset 2013)

表からは FFRI Dataset 2014 のカバレッジが高く，FFRI Dataset 2013，2015 ではカバレッジが低いことがわかる．その要因として考えられるものを以下に示す．

要因: 表 4.2 で示したように検出できた悪性挙動数は多い方から順に FFRI Dataset 2014，2013，2015 となっており，2014 と 2013 ではそのほぼ全ての検体から悪性挙動を検出できている．また，表 4.3 で示したように API 呼び出しの回数は多い方から順に FFRI Dataset 2014，2015，2013 となっている．そして，FFRI Dataset 2013 では 2014 の約 5%，2015 では 2014 の約 10% の API 呼び出し回数となっている．これらのことから悪性挙動を検出できるか否かは必ずしも API 呼び出しの回数に依存しないが，正解挙動を検出できるか否かにおいては API 呼び出しの回数に依存すると推測できる．つまり，本論文におけるカバレッジの値は動的解析時に記録した API 呼び出しの回数に大きな影響を受ける．これは，Cuckoo sandbox で実行される 90 秒間で行える挙動には限りがあり，より多くの API 呼び出しを記録していれば種別に特徴的な悪性挙動を観測する確率が高くなるためだと考えられる．

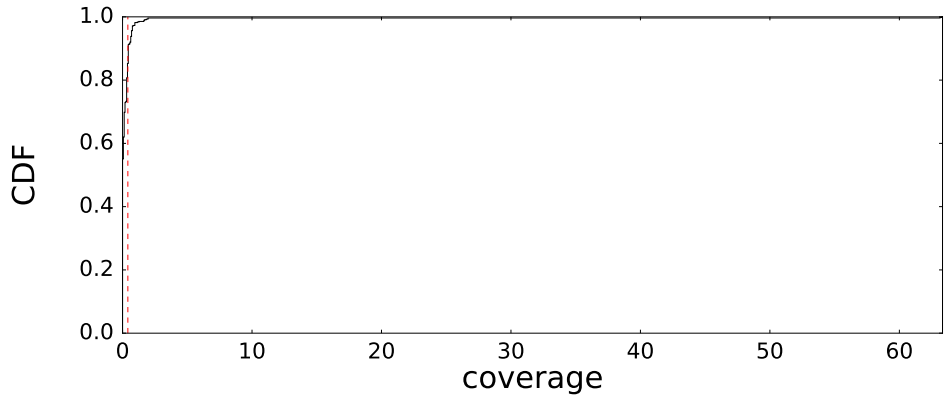

図 4.7　正解レポートに対するカバレッジの CDF (FFRI Dataset 2014)

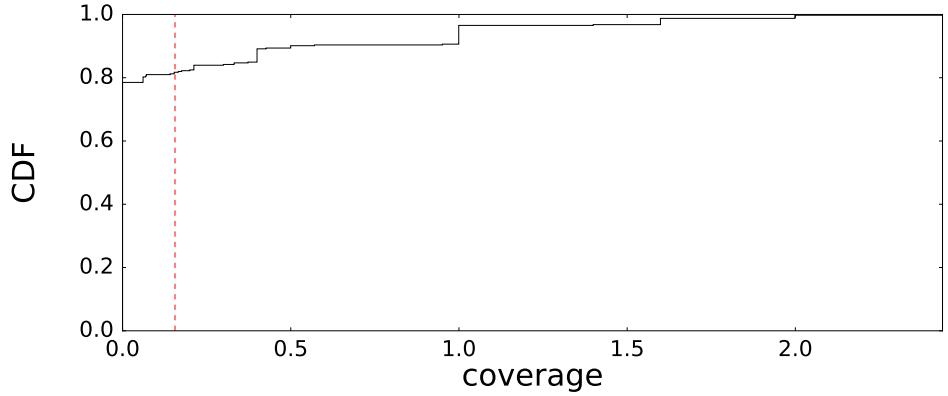

図 4.8　正解レポートに対するカバレッジの CDF (FFRI Dataset 2015)

4.2.2　未知のマルウェア検体について

図 4.1 の中から，未知の検体の結果のみを抽出したものを図 4.9，既知の検体の結果のみを抽出したものを図 4.10 に示す．また，これらの図の数値をまとめたものを表 4.6 に示す．

表 4.6　既知，未知検体における悪性挙動検出数

	最小値	最大値	平均値
既知検体	0	836	6.93
未知検体	0	149	10.33

上記のデータから，既知検体の 78.56%，未知検体の 83.37% から悪性挙動を抽出できていることがわかる．すなわち，本論文で使用したマルウェア検体に関して言えば，悪性挙動を検出

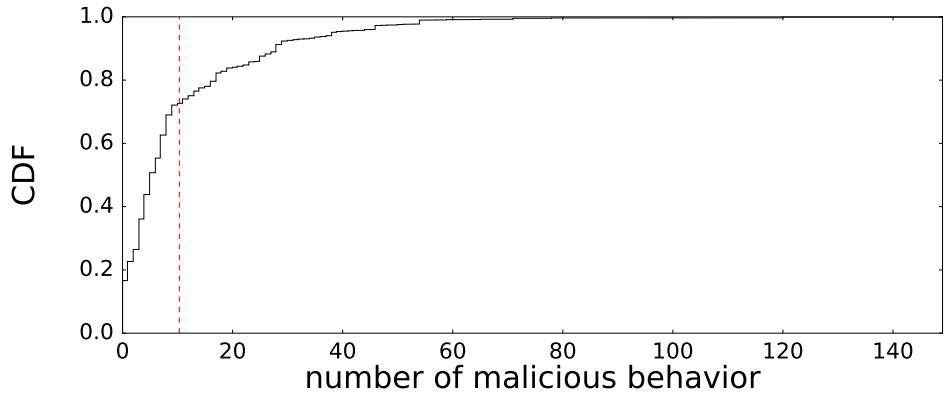

図 4.9　未知検体の悪性挙動数の CDF

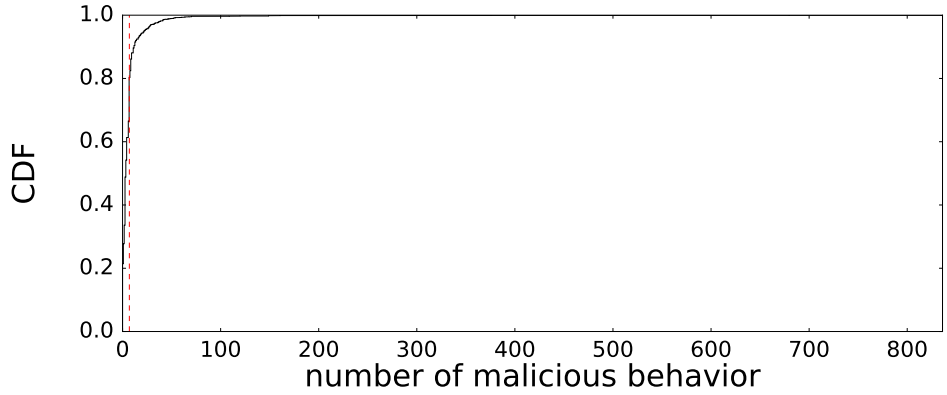

図 4.10　既知検体の悪性挙動数の CDF

できた検体の割合は未知の検体の方が既存の検体よりも大きかったということである．これは以下の 2 つのことを意味している．

意味 1: 未知のマルウェア検体であっても既存の悪性挙動をもっている場合があること

意味 2: 本論文の提案手法はマルウェアの既知，未知に関わらず，適用することができること

4.2.3　解析レポートの草案の自動生成

FFRI Dataset 2013 に属するマルウェア種別名 Worm:Win32/Nuqel.H の検体から 3.3.7 節の手順に従って生成した解析レポートの草案を図 4.11 ～ 4.20 に示す．これらの図から，本論文の提案手法によってマルウェア解析レポートが自動生成可能であることがわかる．

2122.json

Malicious behaviors

When run, it drops copy to varying folder location using random filename. Among possible folder locations are:

- %systemroot%\hinhem.scr

Matching pattern

- %systemroot%[^<>"|?*\]+

This pattern is detected in below malware

- win32/xtrat

This threat can create files on your PC, including:

- %systemroot%\scvhsot.exe

Matching pattern

- %systemroot%[^<>"|?*\]+.exe

This pattern is detected in below malware

- pws:win32/dyzap.m

---

this file drops following files on to the infected computer:

- %systemroot%\system32\advapi32.dll
- %systemroot%\system32\gdi32.dll
- %systemroot%\system32\kernel32.dll
- %systemroot%\system32\msvcrt.dll
- %systemroot%\system32\ntdll.dll
- %systemroot%\system32\ole32.dll
- %systemroot%\system32\oleaut32.dll
- %systemroot%\system32\shell32.dll
- %systemroot%\system32\user32.dll
- %systemroot%\system32\wininet.dll
- %systemroot%\system32\winsock.dll

Matching pattern

- %systemroot%\system32[^<>"|?*\]+.dll

図 4.11　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (1 ページ目)

This pattern is detected in below malware

- trojan:win32/koutodoor.e

---

this file contains an embedded clean CAB file inside the binary. It installs this file on the PC as the following:

- %temp%\2f57_appcompat.txt

Matching pattern

- %temp%[^<>”|?\]+.[^<>”|?\]+

This pattern is detected in below malware

- win32/dalexis

Upon execution, this file drops the following copies of itself with the read-only, system, and hidden file attributes:

- %windir%\hinhem.scr

Matching pattern

- %windir%\hinhem.scr

This pattern is detected in below malware

- worm:win32/nuqel.h

Upon execution, this file drops the following copies of itself with the read-only, system, and hidden file attributes:

- %windir%\scvhsot.exe

Matching pattern

- %windir%\scvhsot.exe

This pattern is detected in below malware

- worm:win32/nuqel.h

This threat drops a copy of itself to a folder with a random file and folder name, such as:

- %windir%\system32\advapi32.dll
- %windir%\system32\gdi32.dll
- %windir%\system32\kernel32.dll

図 4.12　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (2 ページ目)

- %windir%\system32\msctfime.ime
- %windir%\system32\msvcrt.dll
- %windir%\system32\ntdll.dll
- %windir%\system32\ole32.dll
- %windir%\system32\oleaut32.dll
- %windir%\system32\shell32.dll
- %windir%\system32\user32.dll
- %windir%\system32\wininet.dll
- %windir%\system32\winsock.dll

Matching pattern

- %windir%\system32[^<>”|?*\]+

This pattern is detected in below malware

- worm:win32/rebhip.a

Variants can create the following files on your PC:

- %windir%\system32\blastclnnn.exe
- %windir%\system32\scvhsot.exe

Matching pattern

- %windir%\system32[^<>”|?*\]+.exe

This pattern is detected in below malware

- win32/ursnif
- win32/simda

When run, it drops copy to varying folder location using random filename. Among possible folder locations are:

- \msctfime.ime

Matching pattern

- [^<>”|?*\]+

This pattern is detected in below malware

- win32/xtrat

---

this file drops the following components:

- \advapi32.dll

図 4.13　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (3 ページ目)

- \gdi32.dll
- \kernel32.dll
- \msvcrt.dll
- \ntdll.dll
- \ole32.dll
- \oleaut32.dll
- \shell32.dll
- \wininet.dll
- \winsock.dll

Matching pattern

- [^<>"|?*\]+.dll

This pattern is detected in below malware

- trojandownloader:win32/perkesh.gen!a
- trojan:win32/gupboot.a
- trojandropper:win32/vundo.r
- worm:win32/citeary.d
- win32/koutodoor

Upon execution, this file drops the following copies of itself with the read-only, system, and hidden file attributes:

- \blastclnnn.exe

Matching pattern

- \blastclnnn.exe

This pattern is detected in below malware

- worm:win32/nuqel.h

Upon execution, this file drops the following copies of itself with the read-only, system, and hidden file attributes:

- \scvhsot.exe

Matching pattern

- \scvhsot.exe

This pattern is detected in below malware

- worm:win32/nuqel.h

図 4.14　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (4 ページ目)

- \user32.dll

Matching pattern

- \user32.dll

This pattern is detected in below malware

- trojandropper:win32/bamital.g

---

this file creates the following files on an affected computer:

- %systemroot%\scvhsot.exe
- %windir%\scvhsot.exe

Matching pattern

- [^<>"|?\]+[^<>"|?\]+.exe

This pattern is detected in below malware

- trojandownloader:win32/karagany.a
- win32/eyestye

---

this file installs a randomly named copy of itself in any of these paths:

- c:\windows\scvhsot.exe

Matching pattern

- c:[^<>"|?\]+[^<>"|?\]+.exe

This pattern is detected in below malware

- win32/crowti

Disables the Folder Options in Windows Exploirer:

- hkcu\software\microsoft\windows\currentversion\policies\explorer\nofolderoptions___1

Matching pattern

- hkcu\software\microsoft\windows\currentversion\policies\explorer\nofolderoptions___1

This pattern is detected in below malware

- win32/nuqel
- worm:win32/nuqel.ay
- worm:win32/russoturisto.a
- worm:win32/nuqel.h

図 4.15　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (5 ページ目)

Disables Windows Registry Editor:

- hkcu\software\microsoft\windows\currentversion\policies\system\disableregistrytools___1

Matching pattern

- hkcu\software\microsoft\windows\currentversion\policies\system\disableregistrytools___1

This pattern is detected in below malware

- win32/nuqel
- virus:win32/sality.gen!q
- virus:win32/sality.au
- virus:win32/sality.at
- worm:win32/esfury.t
- trojan:win32/startpage.rm
- win32/sality

Disables Task Manager:

- hkcu\software\microsoft\windows\currentversion\policies\system\disabletaskmgr___1

Matching pattern

- hkcu\software\microsoft\windows\currentversion\policies\system\disabletaskmgr___1

This pattern is detected in below malware

- win32/nuqel
- worm:win32/nuqel.ay
- win32/dircrypt
- win32/fakesysdef
- worm:win32/chupik.b
- win32/sality
- trojan:win32/fakesysdef
- virus:win32/sality
- pws:msil/petun.a
- worm:win32/nuqel.h

It changes the following registry entry so that it runs each time you start your PC:

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger___\windows\system32\scvhsot.exe

Matching pattern

- hkcu\software\microsoft\windows\currentversion\run[^<>”|?*\]+_[^<>”|?]+

図 4.16　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (6 ページ目)

This pattern is detected in below malware

- win32/fakevimes
- win32/lethic
- win32/nabucur
- rogue:win32/trapwot
- trojan:win32/lethic.b
- win32/trapwot
- win32/rebhip
- worm:win32/rebhip
- trojandownloader:win32/neojit.a
- win32/eyestye

It modifies the system registry so that it runs every time Windows starts:

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger___\scvhsot.exe

Matching pattern

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger___\scvhsot.exe

This pattern is detected in below malware

- worm:win32/nuqel.h

It changes the following registry entries so that it runs each time you start your PC:

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger___\scvhsot.exe

Matching pattern

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger___[^<>”|?]+

This pattern is detected in below malware

- win32/nuqel

It modifies the system registry so that it runs every time Windows starts:

- hklm\software\microsoft\windows nt\currentversion\winlogon\shell___explorer.exe scvhsot.exe

Matching pattern

- hklm\software\microsoft\windows nt\currentversion\winlogon\shell___explorer.exe

図 4.17　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (7 ページ目)

scvhsot.exe

This pattern is detected in below malware

- worm:win32/nuqel.h

Removes the limit on how long are scheduled tasks are active when set by the AT command of the Shedule service:

- hklm\system\currentcontrolset\services\schedule\attaskmaxhours___0

Matching pattern

- hklm\system\currentcontrolset\services\schedule\attaskmaxhours___0

This pattern is detected in below malware

- win32/nuqel
- worm:win32/nuqel.av
- worm:win32/nuqel.h

Malware name is “worm:win32/nuqel.h”

This malware has following behaviors

It then copies itself in the root of the found shared drives as the following files:

- new folder.exe
- scvhsot.exe

---

this file can download configuration data from a remote server and save the data as the following file:

- %systemroot%\system32\setting.ini

---

this file copies itself to one of the following folders with read-only, hidden and system attributes:

- %systemroot%\system32

Upon execution, this file drops the following copies of itself with the read-only, system,

図 4.18　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (8 ページ目)

and hidden file attributes:

- %windir%\scvhsot.exe
- %windir%\hinhem.scr
- \scvhsot.exe
- \blastclnnn.exe

---

this file can stop the following processes:

- hijackthis.exe
- bkav2006.exe
- mmc.exe
- cmd.exe
- game_y.exe

Disables the Folder Options in Windows Exploirer:

- hkcu\software\microsoft\windows\currentversion\policies\explorer\nofolderoptions\1

Set to bypass a proxy for Internet connections:

- hkcu\software\microsoft\windows\currentversion\internet settings\zonemap\proxybypass\1

The malware deletes the following registry entries:

- hkcu\software\microsoft\windows\currentversion\run\ieprotection
- hklm\software\microsoft\windows\currentversion\run\bkavfw

Disables Task Manager:

- hkcu\software\microsoft\windows\currentversion\policies\system\disabletaskmgr\1

---

this file makes a number of changes to the affected system's settings in order to facilitate the worm's actions by modifying the following registry entries:

- hklm\system\currentcontrolset\services\schedule\attaskmaxhours\0
- hkcu\software\microsoft\windows\currentversion\explorer\workgroupcrawler\shares\shared\new folder.exe
- hkcu\software\microsoft\windows\currentversion\internet settings\globaluseroffline\0

図 4.19　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (9 ページ目)

It modifies the system registry so that it runs every time Windows starts:

- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger\\scvhsot.exe
- hklm\software\microsoft\windows nt\currentversion\winlogon\shell\explorer.exe scvhsot.exe

It changes the following registry entries so that it runs each time you start your PC:

- hkcu\software\microsoft\windows\currentversion\run\msn messsenger[^<>”|?]+
- hkcu\software\microsoft\windows\currentversion\run\yahoo messengger[^<>”|?]+
- hklm\software\microsoft\windows nt\currentversion\winlogon\shell\explorer.exe [^<>”|?]+

this file enumerates shared drives by checking the values within the following registry subkey:

- hkcu\software\microsoft\windows\currentversion\explorer\workgroupcrawler\shares

Disables Windows Registry Editor:

- hkcu\software\microsoft\windows\currentversion\policies\system\disableregistrytools\1

図 4.20　Worm:Win32/Nuqel.H の検体から生成した解析レポートの草案 (10 ページ目)

第 5 章　議論

本章では本論文に関する制限事項と今後の展望について述べる．

5.1　制限事項

本論文の悪性挙動の抽出，検出における制限事項を以下に示す．

5.1.1　悪性挙動の抽出

本論文では悪性挙動の抽出に Microsoft 社の解析レポートを用いているが，この解析レポートから抽出できる悪性挙動の種別と数に制限がある．それぞれについてを以下に示す．

悪性挙動の種別　本論文ではファイル操作系，レジストリ操作系，ネットワーク系，ミューテックス操作系の 4 種を悪性挙動として抽出している．その内，ファイル操作系，レジストリ操作系，ネットワーク系のそれぞれは特定の書式で記述されるため，悪性挙動の抽出が容易である．上記 3 種以外の悪性挙動は特定の書式が用いられないため抽出が難しい．本論文ではミューテックス操作系の抽出も行っているが，判定方法は簡易的なものであり，全てのミューテックス操作系を抽出できているわけではない．このような理由から Microsoft 社の解析レポートから抽出できる悪性挙動の種別は限られる．この制限は Microsoft 社以外のアンチウィルスベンダーによって作成された解析レポートを使用することで解決できると考えられる．これは，各ベンダーが作成する解析レポートの書式はそれぞれ異なり，上記 3 種以外の悪性挙動でも特定の書式を採用している可能性があるためである．

悪性挙動の数　本論文の提案手法では，抽出可能な悪性挙動数はベンダーの技術力に影響を受ける．Microsoft 社の解析結果によれば，本論文で使用したマルウェア検体の内，4,967 が既知検体，3,673 が未知検体であった．しかし，この既知，未知検体の数はベンダー毎に異なる．例えば，Symantec 社 [27] の解析結果では既知検体が 5,299，未知検体が 3,341 存在することになる．このようにマルウェア検体の解析結果はベンダー毎に異なる．この数値から，ベンダー間では所持しているマルウェア解析技術に差異が存在することが推測できる．この差異とは単純な技術力の高低だけではなく，ベンダーの得意分野などの違いも指す．

この差異の存在は文献 [28] からもわかる．この文献では各アンチウィルスベンダーが解析

した結果と独自に解析した結果を比較し，ベンダーの解析結果がどの程度正しいのかを評価している．この評価結果から，ある特定のマルウェア種別においてのみ突出した性能をもつベンダー，全てにおいて高い解析精度を示すベンダー，全てにおいて低い精度を示すベンダーが存在するとわかる．

マルウェア解析技術の差異は解析レポートが存在するマルウェア種別数や記載される悪性挙動の情報量，精度に反映される．仮に，マルウェア解析の全てにおいて Microsoft 社を上回る X 社が存在する場合，ファイル操作やレジストリ操作，その他の悪性挙動に関する記述の情報量も Microsoft 社を上回ると推測できる．また，Microsoft 社では検出できなかったマルウェア検体に関する解析レポートを記述している可能性も存在する．つまり，X 社から抽出可能な悪性挙動数を 100 % とした場合，Microsoft 社から抽出可能な悪性挙動数は制限されていると考えることが出来る．また，特定分野において Microsoft 社を上回るベンダー Y が存在する場合も同様に，抽出可能な悪性挙動数は制限されていると考えられる．これらの制限は Microsoft 社を上回る X 社を探す，または特定分野において Microsoft 社を上回る複数のアンチウィルスベンダーを選定し，組み合わせて使用することで解決できると考えられる．このとき，ベンダーの選定を行わずに複数のベンダーを使用した場合，抽出する悪性挙動の多くが重複し，1 社のみを使用した場合と比較して抽出数が増えない可能性がある．Microsoft 社以外のアンチウィルスベンダーへの対応は今後の課題とする．

5.1.2 悪性挙動の検出

悪性挙動の検出における制限事項についてを以下に示す．

動的解析システムへの依存 4.2 節で述べたように，悪性挙動を検出できるか否かは API コールの情報量に左右されないが，正解挙動を検出できるか否かは API コールの情報量に左右される．これは，動的解析システムで実行される時間でマルウェアが行える挙動には限りがあり，より多くの API コールを記録していれば種別に特徴的な悪性挙動を観測する確率が高くなるためだと考えられる．実行時間については動的解析システムの設定変更で対応できる．しかし，単純に API コールの情報量が多ければ悪性挙動を検出できるわけではない．マルウェアに動的解析システムへの対抗策が適用されていれば，API コールを記録できていても悪性挙動を記録できていないということが起こる．このように，本論文の検出結果は動的解析システムの性能に大きく依存するという制限がある．

抽出結果への依存 本論文では悪性挙動の抽出をファイル操作，レジストリ操作，ネットワーク，ミューテックス操作の 4 種に絞っている．そのため，その他の悪性挙動を検出することができない．マルウェア種別によってはこの制限の影響を強く受け，挙動が一つも検出できないというが起こりうる．また，Microsoft 社の解析レポートに含まれる全ての悪性挙動を抽出できているわけではないため，抽出に失敗している悪性挙動については検出できない．

悪性挙動の誤検出 本論文の悪性挙動 DB では保存された悪性挙動の全てを真に悪性であるとして検出を行っている．しかし，悪性挙動 DB に保存されている挙動の中には正規表現を含むものが存在し，それらは悪性ではない挙動を悪性挙動として誤検出している可能性がある．

5.2 今後の展望

本研究では抽出可能な悪性挙動の種別の拡大，正解レポートに対するカバレッジの改善，Microsoft 社以外のベンダーへの対応を今後の課題とする．この課題の解決のために今後は以下の実現を目指す．

- 今回対象外とした書式 1 (自由記述) の利用
- 今回対象外としたプロセス情報等の抽出方法の開発
- 動的解析システムの設定の最適化
- Microsoft 社以外の解析レポートにおける収集，抽出方法の開発

第 6 章　関連研究

本章では関連研究について述べる．マルウェア解析レポートを扱った研究は筆者の知る限りない．ここでは動的解析ログ中を分析した研究，アンチウィルスベンダーによるラベリングの評価を行った研究について述べる．

Ahmed [29] らは API コールログの時系列情報と API の入出力情報を組み合わせることでマルウェアが検出できることを示した．Alazab [30] らは IDA Pro ディスアセンブラによって得られた API コールに n-gram を適用するで正常検体とマルウェア検体を分類できることを示した．Ravi [31] らも同様に API コールに n-gram を適用することで正常検体とマルウェア検体を分類している．

上記の研究は全て API コールの時系列情報を利用した特徴量を作成し，教師あり学習を適用することでマルウェアの分類を行っている．それに対し，文献 [32]， [33] では教師なし学習を用いた研究を行っている．Bayer [32] らは各マルウェア検体から特徴量を抽出し，クラスタリングアルゴリズムを適用することで類似する検体が同じクラスタに分類できることを示した．Li [33] らはマルウェアのクラスタ分析に使用する正解データが分類結果の精度に大きな影響することを示した．この研究では文献 [32] を含む先行研究の各手法を用いて評価を行っている．

Aziz [28] らは，各アンチウィルスベンダーによる解析結果と独自に解析した結果を比較し，アンチウィルスベンダーの解析結果の評価を行っている．この研究では，ベンダーによる解析は常に正しいわけではなく，メジャーなベンダーであっても解析結果に誤りが含まれていることを示した．

第７章　まとめ

本研究では，アンチウィルスベンダーが作成したマルウェアの解析レポートに着目し，レポート内に記述されたマルウェアの悪性な挙動を自動抽出する手法，抽出した挙動を動的解析ログから自動検出する手法，検出した悪性挙動からマルウェアの解析レポートの草案を自動生成する手法を開発した．

悪性挙動の抽出では，1,678 のマルウェア種別に関する解析レポートに対して処理を行い，ファイル操作系，レジストリ操作系，ネットワーク系，ミューテックス操作系の悪性挙動を抽出した．抽出できたそれぞれの悪性挙動の数は 4,416，1,100，3,505，156 であった．悪性挙動の検出では，8,640 のマルウェア検体に対して処理を行ったところ，1 検体当たり 8.37 の悪性挙動が検出された．上記の結果は，既知，未知のマルウェア検体を区別していない．既知，未知の検体を区別した場合，既知検体からは 1 検体当たり 6.93，未知検体からは 1 検体当たり 10.33 の悪性挙動が検出された．これらの結果は，本研究の提案手法がマルウェアの既知，未知に関係なく悪性挙動を検出可能であることを示している．解析レポートの草案の自動生成では，検査した検体のファイル名，検査した検体のマルウェア種別名，その種別がもつ悪性挙動一覧，実際に検出した悪性挙動一覧，それらの挙動を持つマルウェア種別一覧，それらの悪性挙動の検出に用いた正規表現のパターン一覧を含む解析レポートを自動生成可能であることを示した．

本研究の提案手法は解析レポート内の悪性挙動，つまり既知の悪性挙動を抽出し，検出を行うため，新規の悪性挙動は検出できない．また，検出に正規表現を用いる悪性挙動が存在するため，正常な挙動を悪性挙動として誤検出する可能性がある．上記の理由により，本研究の提案手法はマルウェア検出用のシステムに利用することは難しい．本研究の活用方法としては，解析レポートの草案を自動生成することによるマルウェア解析時の時間的コストの削減，実際に行われた悪性挙動を検出できることを利用したマルウェア検体へのラベリングなどが挙げられる．上記の活用により，本研究がマルウェア解析者，研究者の一助となることが期待できる．

第 8 章　研究業績

本論文の著者自身の業績を以下に示す.

国内会議

1. 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"，コンピュータセキュリティシンポジウム 2013
2. 藤野 朗稚，森 達哉，"エキスパートによるマルウェア解析レポートと動的解析ログの相関分析"，コンピュータセキュリティシンポジウム 2015

国際会議

1. Akinori Fujino，Tatsuya Mori， "Analysis of Massive Amount of API Call Logs Collected from Automated Dynamic Malware Analysis Systems"，International Workshop on Security 2014
2. Akinori Fujino，Tatsuya Mori， "Discovering Similar Malware Samples Using API Call Topics"，IEEE Consumer communications & networking conference 2015

受賞

1. MWS 2013 優秀論文賞，2013 年 10 月
2. CCS 2013 学生論文賞，2013 年 10 月

参考文献

[1] AV-TEST. http://www.av-test.org.

[2] AV-TEST. Malware. http://www.av-test.org/en/statistics/malware.

[3] Microsoft. https://www.microsoft.com/en-us.

[4] 神薗雅紀他. マルウェア対策のための研究用データセット ～MWS Datasets 2015～. MWS 2015 http://www.iwsec.org/mws/2015/, Jul 2015.

[5] VirusTotal. http://www.virustotal.com.

[6] Microsoft Malware Protection Center. Naming malware. http://www.microsoft.com/security/portal/mmpc/shared/malwarenaming.aspx.

[7] Microsoft Malware Protection Center. Win32/Vobfus. http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Worm:Win32/Vobfus.

[8] Microsoft Malware Protection Center. Win32/sality. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Win32/Sality.

[9] Microsoft Malware Protection Center. Trojan:WinNT/sality. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Trojan:WinNT/Sality.

[10] Microsoft Malware Protection Center. TrojanSpy:Win32/Keatep.B. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=TrojanSpy:Win32/Keatep.B.

[11] Microsoft Malware Protection Center. Virus:Win32/Sality.AM. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Virus:Win32/Sality.AM.

[12] Microsoft Malware Protection Center. Virus:Win32/Sality.G. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Virus:Win32/Sality.G.

[13] Microsoft Malware Protection Center. Virus:Win32/Sality.G.dll. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?

Name=Virus:Win32/Sality.G.dll.

[14] Microsoft Malware Protection Center. Virus:Win32/Sality.AT. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Virus:Win32/Sality.AT.

[15] Microsoft Malware Protection Center. Virus:Win32/Sality.AU. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Virus:Win32/Sality.AU.

[16] Microsoft Malware Protection Center. Win32/Bagle. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Win32/Bagle.

[17] Microsoft Malware Protection Center. Win32/Bagle. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Worm:Win32/Bagle.IF@mm.

[18] Microsoft Malware Protection Center. Virus:Win32/Sality.AU. https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Worm:Win32/Sality.AU.

[19] Windows Dev Center. CopyFile function. https://msdn.microsoft.com/en-us/library/windows/desktop/aa363851(v=vs.85).aspx.

[20] Windows Dev Center. CopyFile2 function. https://msdn.microsoft.com/en-us/library/windows/desktop/hh449404(v=vs.85).aspx.

[21] Mozilla Developer Network and individual contributors. Block-level elements. https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements.

[22] Mozilla Developer Network and individual contributors. Inline elements. https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements.

[23] Microsoft Malware Protection Center. Common folder variables. https://www.microsoft.com/security/portal/mmpc/shared/variables.aspx.

[24] Microsoft. 認識される環境変数. https://technet.microsoft.com/ja-jp/library/cc749104(v=ws.10).aspx.

[25] File-Extensions.org. File-Extensions.org - File Extension Library. http://www.file-extensions.org/.

[26] Japan Network Information Center. ドメイン名の種類. https://www.nic.ad.jp/ja/dom/types.html.

[27] Symantec. https://www.symantec.com/index.jsp.

[28] Aziz Mohaisen and Omar Alrawi. AV-Meter: An Evaluation of Antivirus Scans and Labels. pp. 112–131, July 2014.

[29] Faraz Ahmed, Haider Hameed, M. Zubair Shafiq, and Muddassar Farooq. Using spatio-temporal information in api calls with machine learning algorithms for malware detection. In *Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence*, AISec '09, pp. 55–62, New York, NY, USA, 2009. ACM.

[30] Mamoun Alazab, Sitalakshmi Venkataraman, and Paul Watters. Towards Understanding Malware Behaviour by the Extraction of API Calls. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pp. 52–59, 2010.

[31] Chandrasekar Ravi and R Manoharan. Malware detection using windows api sequence and machine learning. *International Journal of Computer Applications*, Vol. 43, No. 17, pp. 12–16, April 2012. Published by Foundation of Computer Science, New York, USA.

[32] Ulrich Bayer, Paolo Milani Comparetti, Clemens Hlauschek, Christopher Krgel, and Engin Kirda. Scalable, behavior-based malware clustering. In *NDSS*. The Internet Society, 2009.

[33] Peng Li, Limin Liu, Debin Gao, and Michael K. Reiter. On challenges in evaluating malware clustering. In *Proceedings of 13th International Symposium on Recent Advances in Intrusion Detection*, RAID 2010, pp. 238–255, 2010.

謝辞

貴重なデータセットを提供してくださった株式会社 FFRI の諸氏に感謝します．熱心なご指導をしてくださった森達哉准教授に感謝します．

付録 A　解析レポートから抽出した悪性挙動の抜粋

解析レポートから抽出したファイル操作系，レジストリ操作系，ネットワーク系，ミューテックス操作系の悪性挙動の抜粋を以下に示す．


```
"c:": {
    "behavior": {
        "winstall.exe": {
            "description": [],
            "malware": ["trojandownloader:win32/renos.gen!a"]},
        "xrhkht.gif": {
            "description": [
                "The malware creates the following files on an affected computer:"],
            "malware": ["trojan:win32/startpage.rm"]},
        "~0002ftd.tmp": {
            "description": ["The malware creates the following files on an affected computer:"],
            "malware": ["worm:win32/sillyfdc.o"]}
    },
    "children": {
        "avast free antivirus": {
            "behavior": {
                "avast.exe": {
                    "description": ["this file creates the following files on an affected computer:"],
                    "malware": ["trojandownloader:win32/banload.ahc"]}}},
        "documents and settings": {
            "behavior": {
                "REGEXP": {
                    "[^<>\"|?*\\\\]+": {
                        "description": ["Examples for <DefaultUserPath> are:"],
                        "malware": [
                            "pws:win32/zbot.gen!al",
                            "pws:win32/zbot.gen!ak"
                        ]}},
                "all users": {
                    "description": [],
                    "malware": ["trojandropper:bat/startpage.a"]},
                "cpa.dll": {
                    "description": ["this file creates the following files on an affected computer:"],
                    "malware": ["trojandownloader:win32/netins.a"]},
                "cpa.exe": {
                    "description": ["this file creates the following files on an affected computer:"],
                    "malware": ["trojandownloader:win32/netins.a"]},},},},}
```


図 A.1　ファイル操作系悪性挙動の抜粋

```
"hkcr": {
  "children": {
    "control panel": {
      "children": {
        "desktop": {
          "children": {
            "scrnsave.exe": {
              "behavior": {
                "REGEXP": {
                  "[^<>\"|?*]+": {
                    "description": [
                      "It modifiees multiple system settings to make sure the
                      dropped executable is run. For example:"
                    ],
                      "malware": ["trojandropper:win32/ropest.a"]}}}}}}}}},
      "software": {
        "children": {
          "adwaredisablekey3": {
            "behavior": {
              "631930542": {
                "description": [
                  "Creates a \"marker\" entry in the registry to avoid running
                  this file more than once, as in this example:"
                ],
                "malware": ["win32/busky"]}}},
          "microsoft": {
            "children": {
              "windows": {
                "children": {
                  "currentversion": {
                    "children": {
                      "explorer": {
                        "children": {
                          "advanced": {
                            "children": {
                              "hidden": {
                                "behavior": {
                                  "0": {
                                    "description": [
                                      "Prevents the display of files and folders with
                                        the 'hidden' attribute set:"
                                    ],
                                    "malware": [
                                      "worm:win32/brontok@mm",
                                    "win32/brontok"
                                    ]
                                  }
                                }
                              },
                              "showsuperhidden": {
                                "behavior": {
                                  "0": {
                                    "description": [
                                      "Prevents the display of Windows system files:"
                                    ],
                                    "malware": [
                                      "worm:win32/brontok@mm",
                                      "win32/brontok"
                                    ]}}}}}}}}}}}}}}}}}}}}
```


図 A.2　レジストリ操作系悪性挙動の抜粋

```
"uranus.kei.su": {
    "description": [
        "this file connects to an IRC server, joins a channel and waits for commands.
            In the wild, we have observed the worm contacting
            the following IRC servers using TCP port 9000:"
    ],
    "malware": [
        "worm:win32/dorkbot.am",
        "worm:win32/dorkbot.ar"
    ]
},
"url9.de": {
    "children": {
        "/ajz?facebook.com/id=": {
            "description": ["In the wild, we have seen the following URL links
                    being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        },
        "/aos?fbprofile=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        },
        "/asz?facebook/id=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        }
    }
},
"urly.de": {
    "children": {
        "/43559?profilephoto=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        },
        "/54453?photos=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        },
        "/54453?viewphoto=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        },
        "/7a77e?photo=": {
            "description": ["In the wild, we have seen the following URL links
                being used to direct you to download the worm:"],
            "malware": ["worm:win32/skypii.a"]
        }
    }
}
```

図 A.3　ネットワーク操作系悪性挙動の抜粋

```
"((mutex))": {
    "description": [
        "The threat creates the following mutexes:"
    ],
    "malware": ["win32/xtrat"]
},
"***mutex***": {
    "description": [
        "this file creates the following mutexes, possibly as an infection marker
            to prevent multiple instances running on your computer:",
        "The trojan creates the following mutexes:"
    ],
    "malware": [
        "backdoor:win32/hupigon.cn",
        "trojandropper:win32/agent.fo"
    ]
},
"***mutex***_persist": {
    "description": [
        "this file creates the following mutexes, possibly as an infection marker
            to prevent multiple instances running on your computer:",
        "The trojan creates the following mutexes:"
    ],
    "malware": [
        "backdoor:win32/hupigon.cn",
        "trojandropper:win32/agent.fo"
    ]
},
"1c5f0c6b74194489b807401a853eb5e3": {
    "description": [
        "where <random> is a filename composed of random letters, e.g. abdvfctghz.exe
            or abdvfctghz.dat. The installer may also create the file
            %ProgramFiles%\\MailSkinner\\msbackup.dat.
            The installer program may create the following mutexes:"
    ],
    "malware": ["win32/skintrim"]
},
"2cbe016a-8f28-4e0c-83a6-6079161294d7": {
    "description": [
        "this file creates the following mutexes, possibly as an infection marker
            to prevent multiple instances running on your computer:"
    ],
    "malware": ["backdoor:win32/bifrose.iq"]
},
"6a0i": {
    "description": [
        "This threat can create one or more mutexes on your PC. For example:"
    ],
    "malware": ["win32/emotet"]
},
"6a0m": {
    "description": [
        "This threat can create one or more mutexes on your PC. For example:"
    ],
    "malware": ["win32/emotet"]
},
"6d5ac198": {
    "description": [
        "It creates a mutex to ensure that only one copy of itself is running at startup.
            The mutex name varies, for example:"
    ],
    "malware": ["trojan:win32/hiloti.gen!a"]
}
```


図 A.4　ミューテックス操作系悪性挙動の抜粋