Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

マルウエア動的解析に於ける自動分類手法の研究

畑上英毅[†]　橋本正樹[†]　堀合啓一[†]　田中英彦[†]

動的解析に於ける自動分類手法について，マルウエアを実行して得られるプロセスの起動情報，レジストリの改ざん情報，通信パケットの内容などといった動的な挙動から得られる情報の他に，マルウエアの静的な情報をマルウエア解析時のパラメータとして扱うことで，検知率の向上を試みた．
従来の動的解析に於ける自動分類は，動的解析より得られた挙動情報に重きを置いて自動分類を行ってきたが，マルウエアそのものから得られる静的な情報を追加することで，従来の自動分類手法に比べ概ね 5 ポイント程の一致率の向上が確認できた．これによって，市販の製品では，マルウエアを検出できない場合でも，蓄積した既存のマルウエアの挙動と類似したマルウエアを抽出し，約 75%以上の精度でその科名を自動的に提示可能となった．

Research of malware classification by dynamic analysis.

Hideki Hatagami[†]　Masaki Hashimoto[†]　Keiichi Horiai[†]
and Hedehiko Tanaka[†]

This paper describes the improvement techniques of the detection rate for dynamic malware analysis, by using static information of malwares as the parameter in automatic classification. This technique improve the detection rate by 5point when compared against automatic classifications so far that focus the behavior information provided by dynamic analysis and can guess the malware name that cannot be detected by the commercial antivirus products at the probability of 75%.

1. はじめに

近年のマルウエアは開発ツールの流通や難読化・暗号化，ポリモーフィック型などの一般化よって，非常に多くの種類が出現し，パタンに頼る検出だけでは困難となりつつあるといわれている．

特に，マルウエアに感染した PC で構成されたボットネットによるスパムメールの大量送信や DDoS 攻撃，情報の奪取などの不正行為が問題となっている．この対策の一環として,総務省と経済産業省によるボット対策のための連携プロジェクト CCC（Cyber Clean Center）が 2006 年 12 月に設立され，運用を開始している．CCC の発表によると，捕獲したマルウエアの 10～20%程度は，捕獲の段階ではマルウエアを検出できない．また，先行研究［1］で構築した定点観測によるマルウエアの捕獲の状況でも，同様の傾向が見られ，捕獲後しばらくしてマルウエア対策製品のパタンが更新されると，ある種類の亜種として検出されることが多い．

「市販の最新のマルウエア対策製品を使ってスキャンを行い，その結果マルウエアが検出されなかった」ということだけでは，必ずしも安心できない状況となりつつある．このため従来の様に，マルウエア対策をセキュリティベンダーに全面的に依存することができ難い状況となりつつあり今後は，自分の組織をターゲットとしたマルウエアについては，解析の一部を自ら実施せざるを得ない可能性がある．

このような背景から，本研究では，先行研究[1]で提案されているマルウエアの自動解析システムに於ける自動分類について、マルウエアを実行して得られるプロセスの起動情報，レジストリの改ざん情報，通信パケットの内容などの他に，マルウエアの静的な情報をマルウエア解析時のパラメータとして扱うことで，分類一致率の向上を試みた．[a]

その結果，全体を通して概ね 5 ポイント程の一致率の向上を確認した．これによって，市販の製品では，マルウエアを検出できない場合でも，蓄積した既存のマルウエアの挙動と類似したマルウエアを抽出し，約 75%以上の精度でその科名を自動的に提示できるようになった. 本提案の手法は, 先行研究［1］の特徴である計算量の少なく，且つ分類精度も高い特徴を犠牲にすることなく，さらに高い分類精度を示すことが可能となった．

以下，次章では，自動解析システムに於ける自動分類手法について述べ，第 3 章では，一致率向上の検討，第 4 章で実験及び実験結果，第 5 章まとめと今後の課題について述べる．

2. 自動解析に於ける自動分類手法

先行研究では，定点観測によってマルウエアを捕獲し捕獲したマルウエアを仮想マシン上の Windows 内で実行して，その挙動を自動的に解析するシステムを提案している．

[†] 情報セキュリティ大学院大学
Institute of Information Security.

a) 本論文中で云う分類一致率とは, 定点観測によって捕獲した約 8000 個体のバイナリファイルを個々の検体の名称が不明だったものと仮定し，提案の手法で推定したマルウエアの名称が市販のマルウエア対策製品でのスキャン結果と一致する割合のことである．

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

本研究では，先行研究[1]での解析結果及び分類手法を利用し，解析の時点で，マルウエア対策製品では，マルウエアとして検出されない対象に対し，挙動の類似性を自動的に算出して，マルウエアの名称を自動的に推定する手法の改良である．これを具体的に実現するためのマルウエアの挙動の解析環境の実装，挙動として取得するデータの種類，挙動の数値化と類似性の判定手法について述べる．

2.1 挙動解析環境

近年のマルウエアの中に，仮想マシンやデバッガの存在を検出して，自身の解析を妨害する種類の存在が知られている．先行研究[1]ではこのようなマルウエアの実行環境の違いが，解析結果に与える影響についても検討できる解析環境を実現している．

ネットワーク環境はLinuxのカーネル･パケット･フィルタに使用されているiptablesの機能を利用して構成されている．模擬ネットワークには，マルウエアを実行する感染PC（Victim PC）（OS:WindowsXP)の他，制御PC(Control PC)と，解析対象のファイルを指定し,解析結果を閲覧するための利用者端末(User console)が接続され,模擬DNS, IRC，SMTP，SMB，HTTPの各サーバ群は実際には制御PCの中に実装がなされている．ここで，IRCサーバが利用する標準的なポート番号は，6667/TCPであるが，マルウエアがIRCサーバとの通信に利用するポート番号を意図的に変更している場合が多い.そこで，模擬環境の中で，マルウエアとIRCサーバとの通信の観測確率を高くするため，iptablesのスクリプトを利用して，複数のTCPポートを模擬環境の中のIRCサーバが待受けている6667へリダイレクトしている．これによって，6667/TCP以外のポートを使って，IRCサーバへログインするマルウエアについても，その挙動を観測できる仕組みとなっている．

VictimPCの部分は，仮想マシンを利用する場合と利用しない場合とで異なった実装となる．ネットワークを模擬環境で構成する利点は，解析を安全に実行できる点に加え，挙動解析の再現性を確保し易い点にある．

仮に感染拡大の防止など外部への影響を緩和する対策を行った上で，インターネットへ接続した状態で解析を行う場合では，マルウエアを実行する時期・時間帯によって通信の相手先の状態の影響を受ける可能性があり，マルウエアの種類とは必ずしも直結しない要素で，観測できる挙動が変化する可能性がある．

インターネットへ接続しないことによって，ポットのHerderからの指令等を観測できないという欠点もあるが，マルウエアを実行した直後の数分間の挙動を自動的に解析し，マルウエアの種類を特定するための環境としては，模擬環境の方が適していると考えられる．また，Windows内の挙動については文献［2，3，4］のようにAPI CALLの情報を解析するのではなく，マルウエアを実行する前の状態と実行後の状態を記録したログを比較し,それらの差分をマルウエアの挙動を示す情報として抽出している．これによって，デバッガの存在を検出してその挙動を変化させるマルウエアへの対策としている．

またマルウエアを実行するWindows OSの種類としてWindows XP及びWindows 2000の場合の比較，OSの脆弱性対策のためのサービスパック適用（SP0,SP2）の有無による比較ができる環境を構築している．

以上のような環境で取得できる情報はAV製品ベンダーなどが公開している，マルウエアの特徴（レジストリの改ざん箇所と内容，マルウエアが作成・削除・改ざんするファイルの名称，起動または停止されるプロセスやサービスの名称，ルートキットの埋め込み，発生する通信のポート番号，通信のあて先，IRCサーバへログインする際のユーザ名やパスワードなど）とほぼ同等である．

2.2 マルウエアの実行で取得する情報

次に前節で示した動的挙動解析の結果から，図1にその一例を示す各項目を文字列の情報として抽出し，これをBDB(Behavior Data Base)として蓄積する．

BDBには，複数の種類の情報を含むが，それぞれの種類の情報を太字で下線付きのアルファベットの一文字で表すこととする．（例：R は、レジストリの改ざん情報）BDBには，各マルウエアに対する複数のAV製品によるスキャン結果を含むが，解析対象のファイルからマルウエアが検出されなかった場合には，Unknown としている．分類にあたり，同種類のマルウエアの挙動は類似していることを仮定している．

2.3 挙動の数値化と類似性の判定

本節では，2.1 で述べたマルウエアの動的挙動の自動解析システムから得られる情報を利用して，マルウエアの分類を行う手法を説明する．

最初に，この分類の手順を述べ，マルウエアの挙動を表す文字列情報を2値のカテゴリ・データとして数値化する手法について述べる．

BDBの各項目を，本論文ではBDBの「要素」と呼ぶ．

BDBの各要素は不定長の文字列で表現された複数個のレコードで構成されている．この BDB の各要素を利用して，分類を行うがこのマルウエアの自動分類処理の全体ブロック構成を図2に示す．
また同図1におけるトップNリストの一例を R，M，T について表1に示す．
ここでトップNリストとは，マルウエアを実行して観測した挙動の各要素に出現する文字列の出現頻度の高い順からN番目までのリストを意味している．

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

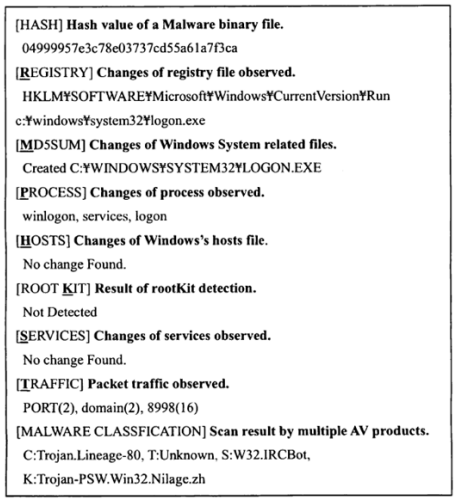

図 1　BDB(Behavior Data Base)の一例

分類の各ステップは次のとおりである．

[STEP-1］　マルウエアを実行し，ファイルの改ざん，プロセスやサービスの起動，トラフィックの発生など，マルウエアの実行に伴って顕在化する動的挙動を記録したデータベース（BDB）を生成する．

[STEP-2］　マルウエアを市販の AV 製品でスキャンし，判明したマルウエアの名称を BDB へ加える．（検出されない場合は Unknown とする）

[STEP-3］　STEP-1 で作成した BDB から要素毎の出現頻度リスト（表 1 における Top N リスト）を生成する．（このステップは，BDB の初期生成時と大幅な更新時のみ実行）

[STEP-4]　Top N リストを使って，BDB の各要素をカテゴリ・データへ変換し，これらを結合して PDB（PDB: Profile Data Base）へ追加する．BDB は複数レコード不定長の文字列の情報であるが，これを固定長のカテゴリ・データへ変換し，複数のレコードを 1 レコードに纏めて PDB を生成する．

[STEP-5]　STEP-3 で生成した PDB を利用し，分類対象の検体と検体自身を除く PDB 内の全てのマルウエア個体間のハミング距離を算出する．

[STEP-6］　距離が最短となるマルウエアを求めてこれを検体の種類の候補として出力する．　ここで，　距離が最短のマルウエアが複数種類存在する場合には，　種類毎の個体数が多い種類を候補として出力する．

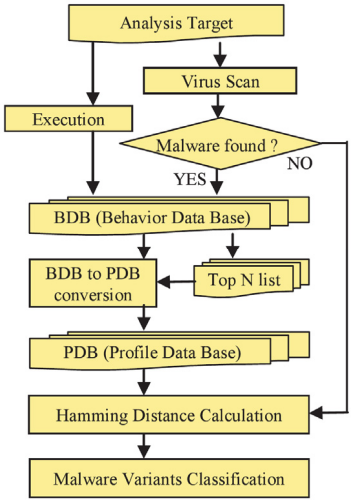

図 2　マルウエア分類処理の全体構成

表 1　R，M，T に関する Top N（N=５）リストの一例

N	R:REGIDTRY(Value)	M:MD5SUM	T:PORT
1	%win%system32¥browseui.dll	%win%system32¥vcmgcd32.dll	80
2	%win%system32¥shdocvw.dll	%win%system32¥spoolsvc.exe	65520
3	%win%system32¥explorer.exe	%win%system32¥lssas.exe	25
4	%win%system32¥urlmon.dll	%win%system32¥algs.exe	1863
5	%win%explorer.exe	%win%system32¥winiogon.exe	8585

2.4 BDB から PDB への変換

マルウエアの挙動の類似性をハミング距離で測定するには，不定長の文字列で表現された BDB から数値で表現した PDB へ変換が必要となる．このため，図 1 で示した BDB 要素の R，M，P，S，K については，BDB 内に記録されたファイル名やプロセス名などの文字列を，要素 T については，観測したパケットのポート番号についてそれぞれ出現頻度の高いトップ N のリストを参照して，カテゴリ・データへ変換する方式としている．

すなわち，BDB 内の各「要素」に記録された文字列等に注目し，出現頻度の多い文字列のトップ N を｛0｜1｝の 2 値へ対応させてカテゴリ･データへ変換する．

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

例えば，レジストリの変化については，R=｛r1,r2,r3,…r_n,r_n+1｝ただし，r_i=｛0|1｝とする方式である．

ここでr_n+1は,文字列がトップN番目以内に入らなかった場合に1をセットする「その他」のカテゴリである．

この BDB から PDB の変換は，BDB の各要素をパラメータとして関数 F を使って PDB へ変換する過程と考えることができる．

この方法では，関数の選び方すなわち，利用する要素の種類，その組合せ，及びトップ N の値（＝カテゴリの種類数）などの影響で PDB が変化し分類の精度が変化するが，本研究では，これについて着目し，第 4 章で述べる実験で検証した．

先行研究[1]における PDB の一例を図 3 に示す．図 3 の各行を PDB の構成要素と呼ぶ．ここで，PDB の構成要素数を｜PDB｜と表現すると，｜PDB｜＝分類対象のマルウエアの個体数である．

```
---hash_value_of_malware_file  Categorized_Malware_Profile_Data ------
00c71d5f5b2931c670270722105dc1f1 1 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0
00d858f0b4ce5cba9d94fc63c3850c2e 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0
00f548639b071fb2f5bf3a7b715f8e39 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
00fa223d7bca367f3f78e6d273eafa0a 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
01020e2280a6a90f91c8079ba4b01991 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0
0103960520dca7d430f398eee95832bb 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0
```

図 3　PDB（Profile Data Base）の一例

3. 一致率向上の検討

本章では，前章にて説明した分類手法に基づいて，マルウエア分類の一致率向上に向けて，分類要素の再検討と新たな要素の発掘を行う．

はじめに，現在分類に用いられているマルウエアを動的に解析して得られた結果に基づく要素について確認を行い，一致率向上への検討を行う．

3.1 分類要素の再検討

本論文では，先行研究[1]で示されている分類要素に着目し，その組み合わせや新たな要素の発掘によって一致率の向上を試みた．はじめに，一致率向上のために現在分類に用いられているマルウエアを動的に解析して得られた結果に基づく要素について述べる．

先行研究[1]によれば，マルウエアを実行した前後で，各要素において何らかの挙動の変化が観測された割合を表 2 に示す．

表 2　要素毎で挙動を観測できた割合

S	H	K	R	P	M	T
3%	17%	19%	77%	96%	67%	82%

この表から，P，R，M，T の各要素については，挙動の変化が観測された割合が多いことから,これらの項目が分類の有力なキーとして利用できる可能性を示している．

以下，これらの要素を挙動の主要素と呼ぶ．また，主要素以外の S，H，K については，マルウエアの実行前後で挙動の変化を観測できた割合が少ないことから，これらの要素単独の情報だけで，マルウエアの分類を行うことは困難と考えられる．

ただし，主要素と組み合わせることによって，分類の精度を向上できる可能性があるので，これらについてはそれぞれの要素についての変化の有無を｛0｜1｝の 1 ビットで表現した．以下，これらの要素を補助要素とよぶ．

これまでに，マルウエアの実行に伴って顕在化した情報（マルウエアを実行して得られるプロセスの起動情報，レジストリの改ざん情報，通信パケットの内容など）について確認してきたが，その他に，マルウエアの静的な情報をマルウエア解析時のパラメータとして扱うことができれば，検知率の向上が見込める可能性がある．

今までの動的な情報に加えて，ファイルそのものからマルウエアの特徴を抽出できれば，分類精度の向上を期待できる可能性がある．

先行研究[1]では，マルウエアの動的挙動を中心とした分類の提案であるが，ファイルそのものから得られる情報を補助的に利用して，分類精度の向上が期待できるとの報告がなされている．

本研究では，ファイル属性情報の手段として，PEiD を用いた情報取得を試みた．

PEiD では File コマンドとは，別の情報が新たに取得できることから，これらの文字列を，カテゴリ・データとし，PDB の一部とすることで，分類精度向上の期待ができる．

3.1.1 File コマンドの利用

UNIX 系 OS に標準で用意されている file コマンドは,ファイルの内容がテキストか，データか，もしくは Windows OS 上で実行可能であるかのなどの判定に利用できる．また，ファイルが圧縮されている場合には，圧縮の形式などの情報も取得できる．

実行可能なファイルの場合には，ターゲットの OS や CPU の種類などの情報を取得できる．捕獲したマルウエアを Windows の環境で実行するに先立って，当該ファイルが Windows 上で実行可能なファイルかどうかを判定するために，この file コマンドを利用することが可能である．

この file コマンドでスキャンした結果，圧縮されたファイルと判定され，そのままでは実行できない場合は,ファイルの解凍後に再度この file コマンドを利用してファイルの調査を行い，解凍後のファイルが実行可能なファイルと判定された場合に，動的

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

解析の対象とする等の使い方ができる．

また，対象としたファイルを，file コマンドでスキャンして得られた情報は，図 4 に示した 13 種類であった．この情報から，Windows OS の実行形式を示す PE，MZ と圧縮形式を表す PEC，UPX のそれぞれの文字列が含まれるか否かをカテゴリ・データとして変換し，PDB の一部とすることによって，分類精度が向上することが判明している．

```
DOS executable (COM)
MS-DOS executable PE for MS Windows (DLL) (GUI) Intel 80386 32-bit
MS-DOS executable PE for MS Windows (DLL) (GUI) Intel 80386 32-bit, Petite compressed
MS-DOS executable PE for MS Windows (DLL) (GUI) Intel 80386 32-bit, UPX compressed
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, PECompact2 compressed
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, Petite compressed
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, RAR self-extracting archive
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, UPX compressed
MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit, UPX compressed, PECompact2 compressed
MS-DOS executable PE for MS Windows (console) Intel 80386
MS-DOS executable, MZ for MS-DOS
MS-DOS executable, PE for MS Windows (GUI) Intel 80386 32-bit
```

図 4　File コマンドによる分類例

3.1.2 PEiD による属性情報の利用

File コマンドと同様にファイルの属性情報を取得する手段として，本論文では PEiD に着目した．PEiD は，GUI ベースで実行ファイルの圧縮形式等の調査が可能なツールで，判定を実行する際に，normal，deep，hardcore の３種類があり，例えば normal で実行した場合の頻度は，次のような情報を得ることができる．図 5

```
Nothing found *
Nothing found [Overlay] *
PECompact 2.x -> Jeremy Collake
Themida 1.8.x.x - 1.9.x.x -> Oreans Technologies [Overlay]
ASProtect 2.1x SKE -> Alexey Solodovnikov [Overlay]
Borland Delphi 6.0 - 7.0
Microsoft Visual C++ 6.0
Themida 1.8.x.x - 1.9.x.x -> Oreans Technologies
Microsoft Visual C++ 6.0 [Overlay]
Microsoft Visual Basic 5.0 / 6.0
kkrunchy 0.23 alpha 2 -> Ryd
ASProtect 2.1x SKE -> Alexey Solodovnikov
PELock 1.0x -> Bartosz Wojcik [Overlay]
Microsoft Visual C++ 3.0
UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo [Overlay]
```

図 5　PEiD による属性情報例

PEiD では File コマンドとは，別の情報が新たに取得できることから，これらの文字列を，カテゴリ・データとし，PDB の一部とすることで，分類精度向上の期待ができる．

4. 実験及び考察

本章では，前章で検討した，一致率向上の為の要素について，実際に自動分類システムに実装し，一致率の算出を行い，また，本方式の有効性を示すため，先行研究[1]で提案されている分類手法との比較を行った．

4.1 実験データ

先行研究[1]にて，行われた実験データは，BDB から PDB へ変換したハッシュ値を異にするマルウエアの個体数が約 8000 であった．

これらの検体は,先行研究で構築された定点観測システムで捕獲したものである．これらのファイルを市販の 3 種類のマルウエア対策製品でスキャンして判明したマルウエアの種類数を表 3 に示す．

表 3　実験対象のマルウエアの製品毎の種類数

	Trend Micro VirusBuster 2007	kaspersky Internet Securiy 6.0	Symantec Internet Securiy
亜種名	1531	942	100
科名	180	96	66

マルウエアの名称は，科名(Family name)と亜種名(Variant name)の組み合わせとなっているが，この表 3 から，同じ検体を対象としても，分類の種類数が製品によって大きく異なっていることがわかる。文献[5,6]においても，あるマルウエアの個体が製品によって別の種類として分類されている例を指摘している．

先行研究[1]においては、表 3 に示したように，実験対象の検体に対し，4 種類の AV 製品の中では最も多くの種類に分類した，VirusBuster2007 のスキャン結果を，マルウエアの名前として利用している．

4.2 実験手順

最初に、先行研究[1]で用いられている要素にて，一致率の算出を行い，次に本研究で提案する PEiD にて取得したカテゴリ・データを元にした，新たな要素の追加を行った上で再度一致率の算出を行い，先行研究[1]にて提案されている手法との比較を行う．

［STEP-1］先行研究[1]にて提案されている，port100 md50 hostN rootN svcN file のカテゴリ・データを元に一致率の算出を行う．

［STEP-2］本研究で提案する PEiD を用いた新たなカテゴリ・データを追加した，port100 md50 hostN rootN svcN file peid を元に一致率の算出を行う．

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

［STEP-3］STEP-1 STEP-2 の結果を亜種名一致率と科名一致率に分けて比較を行う。以上のステップを同種個体数 1~10 個間各 10 回繰り返す.

4.3 実験結果

一致の割合を算出する際に，名称が完全に一致する場合と，科名までが一致する場合の 2 種類について確認を行った．また，先行研究[1]が提案する手法では，距離が最短となるマルウエアの種類が複数の場合には，PDB の中の種類毎の個体数が多い種類として分類する．このため種類毎の個体数が，分類の精度に影響を与える．

この影響を確認するため，下限を横軸として亜種名一致率をプロットした結果を図 6 に示す．図 6 において，上側のプロットは，提案手法で推定したマルウエアの亜種名までが一致した割合を示し，56%~72%．先行研究[1]の手法では，51%~68%．という結果を得た．

また，科名までの一致では，提案手法が 73%~90%．先行研究[1]の手法では，68%~85%という結果を得た．図 7

4.4 実験結果

以上の実験から分類精度に関して，先行研究では，科名の一致率として 85%程，亜種名の一致率として 85%程の分類精度であった．一方，本研究で提案する手法では，科名の一致率として科名の一致率として 73%程, 亜種名の一致率として 90%程の分類精度を得た．一致率としては概ね 5 ポイント程の一致率の向上が確認できた．

5. まとめと今後の課題

5.1 まとめ

本論文では，動的解析に於ける自動分類手法について，マルウエアを実行して得られるプロセスの起動情報，レジストリの改ざん情報，通信パケットの内容などといった動的な挙動から得られる情報の他に，マルウエアの静的な情報をマルウエア解析時のパラメータとして扱うことで，検知率の向上を試みた．

従来の動的解析に於ける自動分類は，動的解析より得られた挙動情報に重きを置いて自動分類を行ってきたが，マルウエアそのものから得られる静的な情報を追加することで，従来の自動分類手法に比べ概ね 5 ポイント程の一致率の向上が確認できた．これによって，市販の製品では，マルウエアを検出できない場合でも，蓄積した既存のマルウエアの挙動と類似したマルウエアを抽出し，約 75%以上の精度でその科名を自動的に提示できるようになった．本提案の手法は，先行研究[1]の特徴である計算量の少なく，且つ分類精度も高い特徴を犠牲にすることなく，さらに高い分類精度を示すことが可能となった．

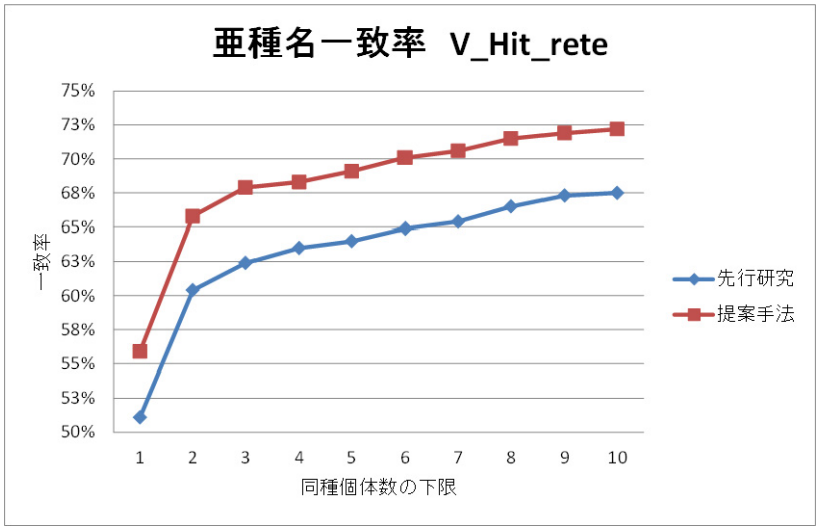

図 6 亜種名一致率

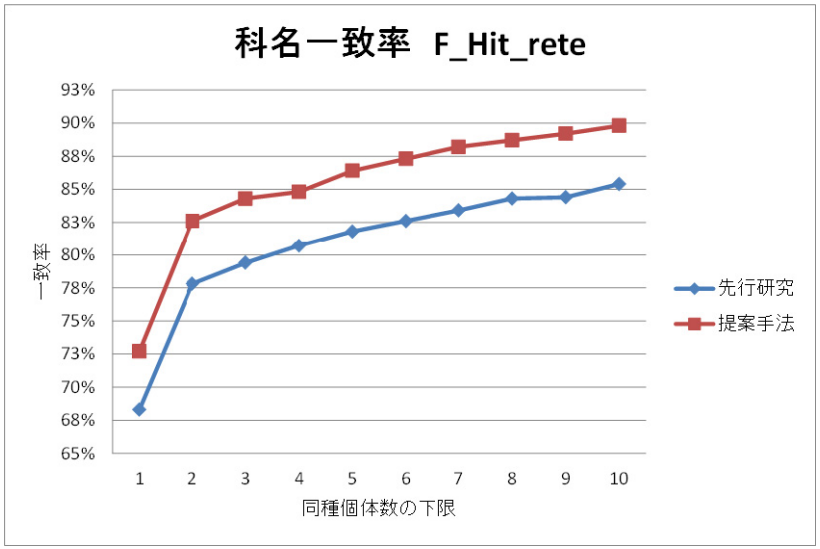

図 7 科名の一致率

Vol.2011-DPS-146 No.51
Vol.2011-CSEC-52 No.51
2011/3/11

5.2 今後の課題

本論文では，動的解析に於ける自動分類手法について，要素の新たな発掘というアプローチで分類精度の向上を試みた．

今後は，要素の定量的な評価を行いさらなる有効性の検討が必要だと考えられる．定量的な評価としては，先行研究でも用いられた ROC 分析や ROC 分析に於ける AUC 値を指標として，要素の検討を行いたい．また，今後とも新たな要素の発掘が期待されるとともに，より多くの検体を確保するために，マルウエアを収集する仕組みについても，検討を行っていく必要がある．また，処理時間を考慮した，新たな分類アルゴリズムの改良も今後の課題である．

参考文献

1) 堀合啓一:マルウェアの自動解析システムと視覚化に関する研究，情報セキュリティ大学院大学博士論文，2008 年度
2) 星澤裕二，太刀川剛，山村元昭，:マルウェア亜種等の分類の自動化，情報処理学会研究報告，2007-CSEC-38，pp271-278，2007
3) Tony Lee，Jigar J.Mody:Behavioral Classification，In Proceedings of EICAR 2006，April 2006
4) Konrad Rieck，Thorsten Holz，Carsten Willems，Patrick Dussel，Pavel Laslov:Learning and Classification of Malware Behavior，DIMVA2008，Detection of Intrusions and Malware，and Vulnerability Assessment，Vol15137/2008，pp108-125
5) UlrichBaye, Christopher Kruegel, Engin Kirda,:TTAnalyze:A Tool for Analyzing Malware,http://iseclab.org/papers/ttanalyze.pdf
6) ITpro「マルウェアの解析対策を無効にする Anti-Anti-Debugging ツールを開発」http://itpro.nikkeibp.co.jp/article/Watcher/20071119/287574/?ST=security&P=1