静的解析の強化によるマルウェア自動分類システムの改善

松藤達彦[†]　橋本正樹[†]　堀合啓一[†]　田中英彦[†]

本研究ではマルウェアの自動分類手法について，マルウェアを実行した際に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挙動情報の他に，マルウェアに含まれる静的な情報をマルウェア解析のパラメータとして扱うことで，分類精度の向上を試みた．

先行研究ではパッカー情報を用いて分類精度を向上させたが，提案手法ではその情報からマルウェアをアンパックし，そこから得られる文字列を用いて，自動分類を行った．従来の手法に比べて，科名・亜種名共に 4.5%程度精度の向上が見られた．これによって，ウイルス対策ソフトで検知ができないマルウェアに対しても，マルウェアの挙動や文字列と，既存のデータを比較することで，67%以上の精度でその科名と亜種名を判別することができる．本提案は，先行研究の特徴である短時間で分類を終え，また分類精度も向上させることに成功した．

The improvement of the malware automatic classification system by enforcing the static analysis

Tatsuhiko Matsufuji[†]　Masaki Hashimoto[†]
Keiichi Horiai[†]　and Hidehiko Tanaka[†]

In this research, we attempted to improve the classification accuracy of our automatic classification system for malware, using static information contained in malware as a parameter in the analysis process, in addition to behavioral information such as starting information of malware, tampering information about registries, and contents of network packets.

While previous research has improved the classification accuracy using packer information, automatic classification of proposed method uses the strings obtained by unpacking malwares with packer information.In this way, we found that the gain of run method is 4.5% in accuracy of family and subspecific name.In our method, we can distinct the name of the family and subspecific of malwares with an accuracy of more than 67% by comparing the accumulated data, the behavior of malwares and strings. This system can classify such malware that can not be handled by commercial anti-virus softwares. Besides, it is possible for our method to improve classification accuracy in the time of analysis is comparable with the one of previous research.

1. はじめに

近年のマルウェアは，マルウェアの開発ツールの流出や難読化・暗号化の利用，ポリモーフィック型などによって，マルウェアの数が爆発的に増加すると共に，種類も増加しているため,パターンによる検出だけでは困難になりつつあると言われている．

近年では金銭搾取を目的としているものが多く観測されている．特に，トロイの木馬などのマルウェアに感染した 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，違法サイトの構築などの様々な不正行為が問題となっている．この対策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)が活動しており，そこの報告によると捕獲したマルウェアの１０～２０％程度のマルウェアは，捕獲の段階ではウイルスを検出できないと言われている．先行研究[1]が構築したマルウェア収集のための定点観測システムにおいても同様の傾向が見られ，捕獲後しばらくしてウイルス対策ソフトウェアのパターンが更新されると，ある種類の亜種として検出されることが多いと報告されている．

そのため，市販のマルウェア対策製品のパターンを常に最新の状態にしてウイルススキャンを行い，その結果マルウェアが検出されなかったというのは，必ずしも安心できる状況であるとは言えない．このため，従来のように，マルウェア対策をセキュリティ・ベンダーに全面的に依存できない状況であるため，今後は自分の組織をターゲットとしたマルウェアについては，解析の一部を自ら実施せざるを得ない可能性がある．しかし，前述のとおり近年のマルウェアには，耐解析機能が利用されており，ネットワークやシステム管理者がマルウェア解析等を行うには大変である．

このような背景から，先行研究[1]は，マルウェアを自動的に捕獲し，解析・分類するシステムを提案した．そのシステムは，ハニーポットとして機能することで，マルウェアを自動的に捕獲・動的解析を行い，その特徴を数値化した上で，ハミング距離を計算して，マルウェアの自動分類を実現している．さらに，解析した結果を可視化し，直感的にマルウェアの傾向を把握できるように工夫されている．一方で，科名と亜種名の分類精度は 72%程度である為，解析精度向上に検討の余地がある．

本研究では，先行研究[1]の自動分類システムをベースとした上で，マルウェアを解析・分類する際の問題点を整理し，分類精度を向上する仕組みを提案する．その方法としては，マルウェ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

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

情報，通信パケットの内容などの動的挙動以外の情報である，マルウェアの文字列を新たな要素として分類を行う．

2. 自動解析における自動分類手法

先行研究[1]では，定点観測によってマルウェアを捕獲し，それらのマルウェアを仮想マシン上で実行して,その挙動を自動的に解析するシステムを提案している.また，その状況を視覚化することによって，組織の管理者等が利用し易い形式で表示している．

本研究では，先行研究[1][2]の解析結果及び分類手法を利用し，収集時点では市販のマルウェア対策ソフトでマルウェアとして検出されない検体に対し，挙動の類似性を自動的に算出して，マルウェアの名称を自動的に推定する手法の改良である．そのため、先行研究[1]で構築されているマルウェアの挙動の解析環境，分類に使用する挙動データ，挙動の数値化と類似性の判定手法について述べる．

2.1 挙動の解析環境

近年のマルウェアの中には，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ものが存在している．先行研究[1]では，このようなマルウェ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築している．この模擬ネットワークには，マルウェアを実行する感染 PC（Victim PC）（OS:windows XP）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，HTTP の各サーバ群は制御 PC の内に実装している．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，マルウェアが 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．そのため，模擬環境内でも，マルウェアと IRC サーバとの通信の観測確率を高くするため，複数の TCP ポートを模擬環境の中の IRC サーバが待ち受けている 6667 へリダイレクトを行っている．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ェアについても，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っている．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，外部への影響を防ぐ対策を行ったとしても，インターネットへ接続した状態で解析を行う場合では，マルウェアを実行する時間等によって通信先の状況の影響を受ける可能性があり，マルウェ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，ボットの Herder からの指令等を観測できないという欠点もあるが，マルウェアを実行した直後の，数分間の挙動を自動的に解析し，マルウェ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，API CALL を記録し，それらの情報を解析するのではなく，マルウェアを実行する前後の状態を比較することで，それらの差分をマルウェアの挙動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知して，その挙動を変化させるマルウェアへの対策としている．

2.2 動的解析で取得する情報

前節で示した環境で取得できる情報は，レジストリの改ざん箇所と内容，マルウェアが作成・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，通信の宛先，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である．これらの情報を文字列情報として抽出し，BDB（Behavior Data Base）として蓄積する（図 1）．

また，BDBには，複数の種類の情報があるが，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示すこととする（例：Rはレジストリの改ざん情報）．BDBには，各マルウェアに対し，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ェアが検出されなかった場合には，Unknowとしている.分類にあたり,同種類のマルウェアの挙動は類似していると仮定している．

[HASH] **Hash value of a Malware binary file.**
04999957e3c78e03737cd55a61a7f3ca
[**R**EGISTRY] **Changes of registry file observed.**
HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
c:¥windows¥system32¥logon.exe
[**M**D5SUM] **Changes of Windows System related files.**
Created C:¥WINDOWS¥SYSTEM32¥LOGON.EXE
[**P**ROCESS] **Changes of process observed.**
winlogon, services, logon
[**H**OSTS] **Changes of Windows's hosts file**.
No change Found.
[ROOT **K**IT] **Result of rootKit detection.**
Not Detected
[**S**ERVICES] **Changes of services observed.**
No change Found.
[**T**RAFFIC] **Packet traffic observed.**
PORT(2), domain(2), 8998(16)
[MALWARE CLASSFICATION] **Scan result by multiple AV products.**
C:Trojan.Lineage-80, T:Unknown, S:W32.IRCBot,
K:Trojan-PSW.Win32.Nilage.zh

図 1　BDB(Behavior Data Base)の一例

2.3 分類の手順と類似性の判定

本章では，2.1 で述べたマルウェア動的解析の自動解析システムから得られる情報を利用して，マルウェアの分類を行う手法を提案する．最初に，分類の手順を述べ，マルウェ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．そして，数値化したデータを PDB（Profile Data Base）と呼び，それらの情報からマルウェアの挙動の類似性をハミング距離で判定し，市販の AV 製品ではマルウェアとして検出されない検体の名称を推定する手法の提案を行う．

2.3.1 分類手順

BDB の各要素（項目）は，不定長の文字列で表現されている．この BDB の各要素を利用して，分類を行うが，マルウェアの自動分類処理の全体図を図 2 に示す．

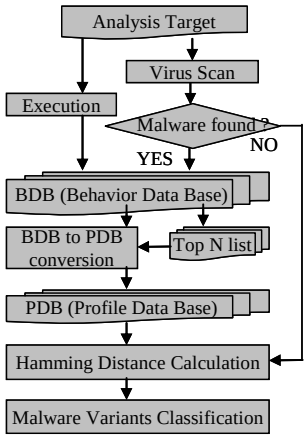

図 2　マルウェア分類処理の全体構成図

また，TOP N list とは，マルウェ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．

分類の各ステップは次の通りである．

[STEP-1] マルウェアを実行し，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，マルウェ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] マルウェアを市販の AV 製品でスキャンし，判明したマルウェアの名称を 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（Top N list）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 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 内の全てのマルウェ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ェ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ェ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．

2.3.2 類似性の判定

マルウェアの挙動の類似性を判定する手段として，ハミング距離を利用している．ハミング距離は，桁数が同じである 2 つの値を比較し，X=Y であれば 0，違うのであれば 1 をとる関数であり，1 となっている個数を結果として出力する．特に，X，Y が 2 値の場合には，排他的論理和演算の結果から，ビットが 1 となっている数をカウントすることで，距離の算出が可能であることから，少ない計算量で結果を得ることができる．

2.4 結果の視覚化

先行研究[1]では，マルウェアの挙動の結果を利用者に使いやすい形で表示することも行っている．特に，収集したログ情報から全般的な傾向の把握に利用できるだけでなく，個々のイベントや個別のマルウェアの挙動に関する詳細な情報まで掘り下げられるように工夫されている．

3. 提案手法

この章では，マルウェア分類の一致率の向上に向けて，新たな分類要素の検討を行うと共に，それを実現するシステムの改善について検討を行う．また，分類要素については，新しい要素を追加するだけでなく，現在分類で使用されている要素を再検討することで，一致率の向上を目指す．

3.1 一致率向上に向けての検討

先行研究[1]では，マルウェアを実行した前後で，各要素において何らかの挙動が観測された．その割合を表 1 に示す．

この表では，R,P,M,T の各要素では挙動の変化が多く観測され，この 4 つの項目が分類で重要な要素となると考えられる．逆に，S,H,K は，マルウェアを実行した前後で観測できるものが少ないことから，これらの情報だけでマルウェアの分類を行うのは難しいと考えられる．しかし，それらの挙動を行うものと行わないものなどの違いがあるため，主要素と組み合わせることで分類精度が向上する可能性がある．そのため，変化が観測できた割合によって，分類要素の比重を決める必要がある．

表 1　BDB の要素毎で挙動を観測できた割合

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

また，先行研究[2]は，マルウェアの実行に伴って顕在化した情報（マルウェ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，マルウェアの静的な情報をマルウェ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として，パッカー情報を追加した．その結果，先行研究[1]の手法より５％程度分類の精度が上がった．

本研究も同様，マルウェアの静的な情報をマルウェア分類時のパラメータとして扱うが，先行研究[2]で新たに追加されたパラメータは「何で圧縮されているか」という外部ツールの情報のみである．分類の精度が向上していることから，本研究においてもマルウェアから得られるパッカー情報は補助的に利用する．

3.2　提案手法の概要

先行研究[1]では，API CALL の記録に依存しないで，マルウェアの実行前と実行後の Windows の状態から，マルウェアの挙動を解析する方法を提案している．しかし，実行中の一時的な変化を見落とすことがある．そのため，本研究ではマルウェアから得られる文字列（API など）を用いて，分類を行う．

また，分類にはマルウェ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動的挙動と，マルウェアの実行ファイルを解析することで，動的解析で得られなかった機能などの特徴的なパターン列を新たな要素として用いる．

マルウェアの静的な情報（文字列）を入手するにはアンパックの作業が必要である．それらの流れを含めた，提案手法を以下に示す．詳細については後に説明を行う．

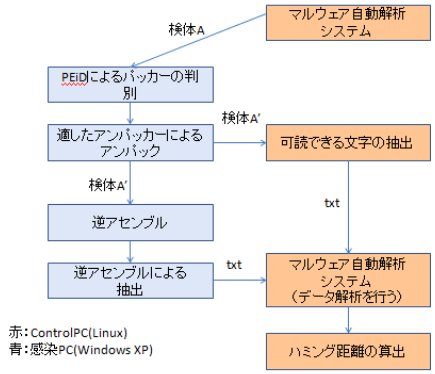

図 3　マルウェア分類処理の構成（追加部分）

3.3　マルウェアの文字列

マルウェアから静的な情報を収集できる部分としては，収集後にパッカー等で圧縮等が施された検体と，パッカー等の圧縮を外した検体から 2 つである．1 つ目のパッカー等で圧縮された状態の検体では，本来のコード部分は圧縮・難読化されていることが多いため，IAT 等には最小限の API だけしか登録されていない場合がある．そのため，この状態で得られる情報はパッカーの情報以外では LoadLibrary や GetProcAddress などの限られたものだけしか抽出できない場合があり，マルウェアを分類する上で重要な情報を抽出するのは難しいと考えられる．

そして，2 つ目はパックされたマルウェアをアンパックすることで得られるデータから情報から方法である．これは実際に解析者が静的解析を行う際に実際に行う手法であるため，有用な情報が収集できる．得に，マルウェアの機能を把握するためには，マルウェアが利用する Win32 API 等の情報が重要である．文献[3]では、ある API とその後に呼び出される API の組が現れる頻度を API 推移と定義し，マルウェア検体の特徴として分類を行っており、マルウェアを分類する上で役立つと考えられる。

今回、実際にマルウェアから抽出した文字列としては，図 4 のようなものがある．図 4 の左側の数字は 865 検体中に右側の文字列が含まれていた検体数を示し，右側の文字列は strings コマンドで抽出した文字列である．文字列は 47923 個抽出され，分類のパラメータにしたいと考えていた文字列も数多く抽出されていた．

```
794 GetProcAddress
768 LoadLibraryA
764 kernel32.dll
699 GetModuleHandleA
637 ExitProcess
606 GetModuleFileNameA
592 RegOpenKeyExA
590 RegCloseKey
582 ShellExecuteA
574 user32.dll
569 %d.%d.%d.%d
567 RegQueryValueExA
564 Windows for Workgroups 3.1a
564 PC NETWORK PROGRAM 1.0
564 NT LM 0.12
564 LM1.2X002
564 LANMAN2.1
564 LANMAN1.0
563 ws2_32.dll
562 WNetAddConnection2A
560 wininet.dll
```

図 4　マルウェアから抽出した文字列（一例）

文字列の抽出は，アンパックできた全てのマルウェアに対して行う．マルウェアから抽出した文字列は 1 つのマルウェアに同じ文字列が何度も出現することがあるため，各検体から抽出した文字列をユニークにする．そして，それらのマルウェアから可読出来る文字列を取り出し，頻出順に並べる．でそうすることで，全マルウェアに同じ文字列がどのくらい含まれていたかがわかり，分類の候補となる文字列が選びやすくなる．

図 5 は，865 検体か抽出した全ての文字列をグラフ化したものである．このグラフより，頻出順に並べられた文字列の TOP10 は約 570 検体（約 66%）に含まれていたことがわかった．また，その他にも TOP100 では約 46%，TOP1000 では約 21%含まれていることが分かる．そして，重複しない文字列が 44000 程度あったが，これは対象となるマルウェアが 1 つしかないため，分類のパラメータとしては非効率，または無意味なパラメータであることがわかる．

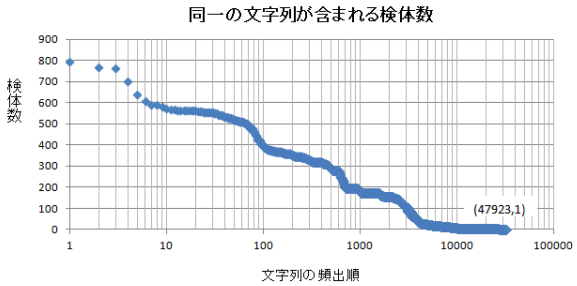

図 5　同一の文字列が含まれる検体数

これらの文字列は動的解析以外の情報を新たに取得できることから，これらを新たな分類要素として利用するために，先行研究の PDB に追加することで，分類精度の向上が期待できる．

3.4　分類に使用するPDBの生成

マルウェアに TOP20 番目以内の文字列があれば，0 を出力し，２０番以内に入らなかった場合 1 を出力する．図 6 の左側はマルウェア名（ハッシュ），右側が出力した値である．

```
02c9786f7d306fb16bcc8549e13206b2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
02d87eb606fc102564901362e2d9675d 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
034680ae512fcd183a5a1e75e5b34986 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1
036cf2d195b0811fe16e061f59cade50 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
03c9e19670e98e0a2fa073ff0035114a 0 0 0 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1
03d686f5a8a6165d1cd031ded799b1b4 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1
0472e982506396b32a3fd87e5bd90dbf 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0496de129dff0e9e46ac8123f4db2b35 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
04999957e3c78e03737cd55a61a7f3ca 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0
050232d9d4e1c5bf4fe370e3a9e246454 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

図 6　提案手法で追加した部分の PDB

また，ハミング距離の計算に，XOR（排他的論理和）の計算をしているが，0｜1 の値を 1 ビットずつ計算するよりも，16 進に変換して，まとめて計算することによって，計算速度を向上させている．その他にもデータサイズが小さくなり，見やすくなっている．

4.　実験と考察

本章では，3.2 の概要で説明した方法を既存のシステムに追加し，先行研究から得られる動的解析の結果と，新たなに追加した要素を用いて，一致率の算出を行う．また，本方式の有効性を示すために，先行研究[2]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[2]で行われた実験データは，先行研究[1]の定点観測システム捕獲し，PDB へ変換したハッシュ値がユニークとするマルウェア 6828 検体を対象にしているが，本研究ではその検体の中でもパッカーが判断でき，アンパックに成功した 865 検体について行う．

表 2　先行研究との対象範囲の違い

	合計	PE ファイルでない	パッカーがわからない検体	パッカーが判別できた検体 (アンパック出来た検体)
先行研究[2]の対象検体数	6828	109	4997	1722(865)

マルウェアの名称は，科名(Family name)と亜種名(Variant name)の組み合わせとなっているが，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ことなることがわかっている．文献[4,5]においても，あるマルウェ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]では，マルウェア 6828 検体を市販のウイルス対策製品（Trend Micro:VirusBuster，kaspersky:Internet Security，Symantec:Internet Security）でスキャンした結果，最も多く分類した，VirusBuster のスキャン結果をマルウェア名として利用している．

4.2　実験の手順

最初に，先行研究[2]で用いられている要素で VirusBuster との一致率を算出し，次に本研究で提案する文字列を PDB に加えて，再度一致率の算出を行い，先行研究[2]で提案されている手法との比較を行う．手順は以下の通りである。

[STEP1-1]　先行研究[2]で提案されている，port100，md50，hostN，rootN，svcN，file，peid のカテゴリ・データを元に一致率の算出を行う．

[STEP1-2]　本研究で提案するマルウェアの文字列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，str20 を用意し，一致率の算出を行う．

[STEP1-3]　STEP1-1 と STEP1-2 の結果を科名の一致率と亜種名の一致率の 2 つに分けて，比較を行う．

上記の処理を同種個体数 1～10 個間の 10 回繰り返す．

4.3　実験結果

● 一致率

ウイルス対策ソフトとの一致率の算出には，科名が一致する場合と，亜種名までが完全に一致する場合の 2 種類について行った．

先行研究が提案する手法では，距離が最短となるマルウェ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，下限を横軸として科名一致率をプロットした結果を下記に示す. 図 7 では, マルウェアの科名の一致率を示し，先行研究では 55%～80%であり，提案手法では 60%～83%となった．平均で 4.6%の精度向上が確認できた．図 8 では，マルウェアの亜種名の一致率を示し，先行研究では 46%～75%であり，提案手法では 50%～80%となった．平均 4.4%%の精度向上が確認できた．

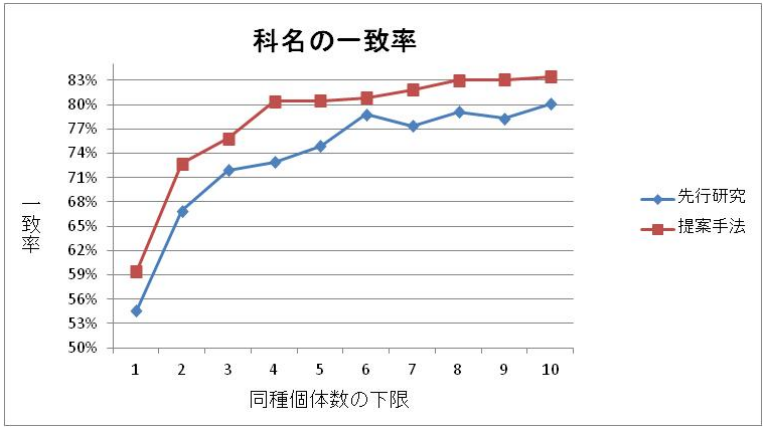

図 7　科名の一致率

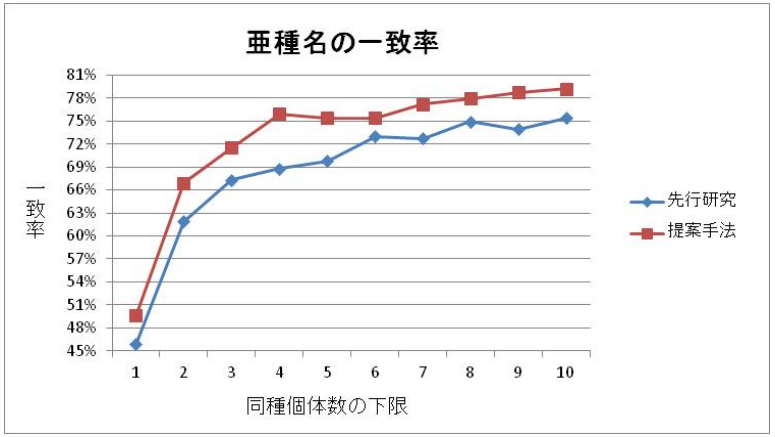

図 8　亜種名の一致率

● 分類時間

同種個体数の下限毎に所要した分類時間は以下のようになった．

表 3　同種個体数毎の分類時間

分類所要時間(秒)										
同種個体数の下限	1	2	3	4	5	6	7	8	9	10
先行研究	9.371	5.884	4.789	4.233	3.96	3.777	3.399	3.295	3.119	3
提案手法	14.462	8.669	7.187	6.101	5.764	5.384	4.859	4.771	4.507	4.307
増加した時間(提案手法-先行研究)	5.091	2.785	2.398	1.868	1.804	1.607	1.46	1.476	1.388	1.307

分類処理の時間に関しては，865 検体という少ない数だったせいか，あまり時間に差が見られなかった．そのため，今後はアンパックの精度を上げると共に，分類できる検体を増やしていくことで,提案手法の処理時間を評価して行きたいと考えている．

5. まとめと今後の課題

5.1 まとめ

本研究ではマルウェアの自動分類手法について，マルウェアを実行した際に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挙動情報の他に，マルウェアに含まれる静的な情報をマルウェア解析のパラメータとして扱うことで，分類精度の向上を試みた．先行研究[2]では，パッカー情報を用いて分類精度を向上させたが，提案手法ではその情報からマルウェアをアンパックし，そこから得られる文字列を用いて，自動分類を行った．従来の手法に比べて，科名・亜種名共に 4.5%程度精度の向上が見られた．これによって，ウイルス対策ソフトで検知ができないマルウェアに対しても，マルウェアの挙動や文字列と，既存のデータを比較することで，67%以上の精度でその科名と亜種名を判別することができる．本提案は，先行研究の特徴である短時間で分類を終え，また分類精度も向上させることに成功した．

5.2 今後の課題

本研究でのアプローチで見つかった今後の課題は３つある．

1 つ目はマルウェアを収集する仕組みである．先行研究[1]によれば，マルウェアの挙動が格納されている PDB が増えることで精度が向上すると報告されている.そのため，より多くのマルウェアを収集する仕組みを検討し，実装する必要があると考えられる．

2 つ目は，アンパックの成功率を上げる方式の導入である．本研究では先行研究が対象としている検体数が少なくなっているため、先行研究の科名の一致率が低下している。そのため，独自のアンパッカーを用意するなど，アンパックの成功率を上げ，多くのマルウェアから情報を収集する必要がある．

3 つ目は，マルウェア分類に最適な要素の選定である．今回，分類に使った TOP20 の文字列は,「とある種類のマルウェアにはこの文字列が含まれているため，分類精度が向上する」と確証を持ってやっていないものもある．そのため，逆に精度を下げている文字列がある可能性があるため，分類に使う文字列についても検討を行う必要があると考えられる．

また，上記以外にも逆アセンブルから得られる情報についても解析を進めることで新たな要素を発掘し，精度の向上が考えられる．また，処理時間を考慮した，分類方法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)
2)　畑上英毅，橋本正樹，堀合啓一，田中英彦:マルウエア動的解析に於ける自動分類手法の研究，Vol.2011-CSEC-52，No.51，pp.1-7(2011)
3) 岩本一樹，和崎克己:静的解析によるマルウェアの自動分類と結果の検討，情報処理学会シンポジウム，vol.2010，No.1，pp477-491(2010)
4)　星澤裕二，太刀川剛，山村元昭: マルウェアの亜種等の分類の自動化，情報処理学会研究報告，Vol.2007-CSEC-38，No.，pp.271-278(2007)
5) Michael Bailey, Jon Andersen, Z. Morley Mao, Farnam Jahanian, Jose Nazario, Automated Classification and Analysis of Internet Malware. In　Proceeding of the 10th Symposium on Recent Advances in Intrusion Detection　(RAID’07), pp.178-197,2007.