社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＣＳＥＣ－38 (39)
2007／7／20

マルウェアの亜種等の分類の自動化

星澤 裕二　太刀川 剛　山村 元昭

株式会社セキュアブレイン 〒102-0083 東京都千代田区麹町 2-6-7
E-mail: {yuji_hoshizawa, tsuyoshi_tachikawa, motoaki_yamamura}@securebrain.co.jp

あらまし 現在，発見されたマルウェアが新種であるか，あるいは既知のマルウェアの亜種や変種であるかは，アンチウイルスベンダのエンジニアが経験に基づき判定している．特定の分類基準に基づいた判断ではないため，エンジニアやベンダによって判定結果が異なることがある．亜種名(Variant Name)の相違ではなく，科名(Family Name)が異なるケースもある．このことから，現状においては，マルウェアの分類は曖昧であり，マルウェア名が必ずしも有益なものではないと言える．本稿では，現状のマルウェアの分類における問題点を整理し，マルウェア分類基準の確立と分類作業の自動化を検討する．

キーワード マルウェア，ボット，亜種，分類

Automatic malware classification

Yuji HOSHIZAWA　Tsuyoshi TACHIKAWA　and　Motoaki YAMAMURA

SecureBrain Corporation　2-6-7 Kojimachi, Chiyoda-ku, Tokyo, 102-0083 Japan
E-mail: {yuji_hoshizawa, tsuyoshi_tachikawa, motoaki_yamamura}@securebrain.co.jp

Abstract It is determined by engineers of Antivirus companies whether the detected malware is new or variant based on past experience. The determination may differ because there are no particular classification criteria. There are some cases that differ family names, not variant names of malware. This indicates that malware classification is vague at present, and the names of malware are not always useful. In this paper, we organize the problems of malware classification at present, and consider the establishment of classification criteria and automation of classification of malware.

Keyword Malware, Bot, Variant, Classification

1. はじめに

1990 年代初頭からアンチウイルスベンダはマルウェアの分類に多大な労力を割いてきた．世間が注目するマルウェアに対しては特にそうであるといえる．日々発見される新種のマルウェアの数が少なく，2，3ヶ月毎に新しいマルウェア検出用シグネチャが提供されていた 1990 年代初頭では，マルウェアの命名ルールを共有し維持することがはるかに容易であった．そのような状況下では，アンチウイルス研究者たちが，マルウェアの命名規則を共有，管理していく時間的余裕を持つことができた．時が経つにつれ，新種のマルウェアが発見される頻度と新しいシグネチャをリリースしなければならない頻度が変わっていった．

今日では，日に複数のシグネチャがリリースされている．そして，トロイの木馬，スパイウェア，ボットなどの新しい種類のマルウェアが存在している．このような変化の中，マルウェアを正しく分類することははるかに困難になっており，もはや人的努力だけでは管理することが難しい．初めてボットが発見された当時，複製を作らないという理由からウイルスと同等には扱われなかった．アンチウイルスソフトがこのような種類の脅威を検出すべきかどうか議論されることもあった．結局はアンチウイルスソフト批評家からの圧力と顧客の需要に後押しされる形で，アンチウイルスソフトはスパイウェア，アドウェア，トロイの木馬やボットといった“望ましくない”ソフトウェアは何でも検知するようになっていった．同時に，アンチウイルス研究者に対してはこのようなマルウェアについても名前をつけることが要求されるようになる．

多くのアンチウイルスベンダは，このような種類の脅威に対しては総称的な名前をつける手法をとった．例えば，IRC サーバを使って制御されているボットであれば，W32_IRCBot.Family.A.というようなユニークな名前ではなく IRCbot といった総称的な名前がつけられた．

2. 現状の調査

マルウェアの分類状況を調査するにあたり，世の中に出回っている実在のサンプルの中からマルウェアであるかもしれないと思われたものを取り上げる．これらのサンプルはすべて，マルウェアの分類と命名規則の研究のために実在のハニーポットから収集されたものである．

2.1. アンチウイルスソフトでのスキャン結果

シマンテック，トレンドマイクロ，マカフィー各社のアンチウイルスソフトで前述のマルウェアサンプルをスキャンし，そのスキャン結果からマルウェアの分類についての問題を考察する．

（1）シマンテック社のアンチウイルスソフトを使用して前述の 3812 個のマルウェアをスキャンした結果，3305 個のサンプルが検知された．スキャン結果で見てみると，シマンテック社製アンチウイルスソフトはマルウェアに対してユニークな亜種名をつけていないようである．検知されたサンプルは 47 ファミリに分類された．

シマンテック社製アンチウイルスソフトはマルウェアに対してユニークな亜種名をつけていない．例えば 372 サンプルが W32.IRCbot ファミリとして分類されている．これはトレンドマイクロ社やマカフィー社ソフトと比較して非常に高い数字である．おそらく，IRC やバックドア機能が関係していると思われるサンプルであればどれも W32.IRCbot として分類されると考えられる．

（2）トレンドマイクロ社製アンチウイルスソフトでスキャンすると，3332 個のサンプルが検知された．スキャン結果で見てみると，トレンドマイクロ社製アンチウイルスソフトはマルウェアを分類している．検知されたサンプルは 85 ファミリに分類された．

トレンドマイクロ社製アンチウイルスソフトは検知した多くのマルウェアサンプルにユニークな亜種名をつけている．BKDR_IRCBOT.JM を例にとると，JM というユニークな亜種名がつけられていることが分かる．

今回のスキャン結果では，16 個のサンプルが BKDR_IRCBOT ファミリに分類され，15 個のユニークな亜種名がつけられている．その中で，異なる MD5 ハッシュ値を持つ複数のサンプルがまったく同じ亜種に分類されていることに注目した．例えば，BKDR_IRCBOT.JM に分類されている 2 つのサンプルが存在する．ファミリ名の中には，非常に似通っていておそらく同一のファミリであることを意図して作られたと思われるものもある．例えば，TROJ_MANCSYN と TROJ_MANCYSYN は同じトロイの木馬ファミリであると考えられる．

分類化アルゴリズムやデータベースを使わずに分類を行った結果として，同じファミリに対して複数のファミリ名がつけられてしまうということが起こる．一本化した分類データベースを用いらずにこのような問題を解決することは容易ではない．

（3）マカフィー社製アンチウイルスソフトでは，2705 個のファイルがウイルスとして検知された．スキャン結果で見てみると，マカフィー社製アンチウイルスソフトはマルウェアを分類してはいるが，ユニークな亜種名をつけるというよりは，総称的なファミリ名にグループ化されていることが分かる．検知されたサンプルは 45 ファミリに分類された．

マカフィー社製アンチウイルスソフトはユニークな亜種名をつけていない．例えば，9 個のサンプルが W32/IRCbot ファミリに分類された．ほとんどの亜種名に generic（総称的な）の短縮形である gen がつけられている．

2.2. 問題点

アンチウイルスベンダにとってマルウェア命名規則を標準化することは容易に解決できない問題とされてきた．アンチウイルスベンダの研究者が最善を尽くしたとしても，日々発見される新しいマルウェアすべての一つ一つについて命名規則を標準化することは現実的に無理がある．

表 1 はアンチウイルス各社ソフトの検出名を比較したものである．次に示す 3 つのサンプルを例にとってみると，3 社の内 2 社間では非常に似通ったファミリ名をつけているが，残りの 1 社はまったく異なる名前をつけていることが分かる．

表 1 アンチウイルスの検出名

サンプルファイルの MD5 ハッシュ値	シマンテック検出名	トレンドマイクロ検出名	マカフィー検出名
1ed40a526fcc4d8afc114c09667b716b7187dd4f	Backdoor.Trojan	WORM_IRCBOT.RJ	W32/IRCbot.gen
47c82190a3a53eabecee1a2d758e78fafbfb5463	W32.Spybot.Worm	WORM_IRCBOT.KO	W32/Sdbot.worm.gen.h
e888a2ab4a4a8db34adf6be8304bd0a69fab205a	W32.Esbot.A	WORM_ESBOT.A	W32/IRCbot.worm.gen

ベンダ間での標準化が難しいとしても，多くのアンチウイルスベンダが一つ一つのマルウェアに対してユニークな識別子をつけていないことは明らかである．次に考えられる原因を挙げる．

（1）ほとんどすべてのアンチウイルスベンダは，自社ソフトが検知するマルウェアすべてについて詳細な説明を提供している．もし一つ一つのマルウェアにユニークな名前をつければ，作成しなければならないマルウェア情報の数は必然的に増加する．情報を公開するまでに要する時間にも影響を与えるだろう．

（2）アンチウイルスソフトによっては割り当てられるウイルス名の数に制限があることも考えられる．また，ソフトウェアの仕様によっては，スキャンする際にすべてのウイルス名をメモリにロードするために

更なるメモリ領域割り当てが必要になるだろう．これは余計な負荷となり得るので，パフォーマンスを考慮すればアンチウイルスベンダは採用しないだろう．

(3) 多くのアンチウイルスベンダは，新しいシグネチャを開発するエンジニアを複数抱えている．つまり，マルウェアを命名するエンジニアが複数存在することになる．重複した変種・亜種名をつけないためには命名工程を管理するためのインフラやプロセスが必要となる．これはアンチウイルスベンダにとってさらなる負荷となる．

(4) アンチウイルスソフトでは，オリジナルのプログラムコードをもとにサンプルの一意性を判断している．今日では，大抵のマルウェアが EXE パッカーを使って変異されている．同一のマルウェアが異なるパッカーで圧縮されている可能性があるため，マルウェアが新しい亜種かそうでないかを判断するためには復元されたコードを比較する必要がある．これは非常に複雑な工程となる．なぜなら，マルウェアをオリジナルのプログラムの状態に完全に復元することが不可能である場合が多いからである．このような状況下では，マルウェアの一意性を判断して新しい亜種であるかを判断するためには長い時間を費やさなければならない可能性がある．

3. 動的解析結果による分類

動的解析解析環境で得られた API 呼び出しログを用い，マルウェアを分類することを検討した．

動的解析環境とは，マルウェアを実行する感染ホスト 1 台と擬似インターネット環境を提供するダミーサーバ 1 台の計 2 台で構成される．感染ホストでは，マルウェアが使用する一連のマシン操作を監視するためのモジュールにより，レジストリキーの変更やファイル操作，メモリ操作やストリングの改ざん，ネットワーク活動などをログに記録する．ログには，呼び出された Windows API 名と各パラメータ，戻り値などが記録される．本調査では，当該ログのみを使用した．ダミーサーバ上では，擬似インターネット環境を実現するために必要な，カスタマイズされた DNS，HTTP，SMTP，IRC などのサーバモジュールを動作させた．前述の擬似的被害者となる感染ホストにとっては，擬似インターネット環境を提供するマシンが唯一のゲートウェイであり DNS となる．環境設定により，トラフィックがルーティングされるべき宛て先 IP アドレスやホスト名に関わらず，すべてのネットワーク・トラフィックが擬似インターネット環境を提供するダミーサーバにリルートされるようにした．

後述する亜種判定では，次の加工を施した API 呼び出しログを使用した．

(1) API 呼び出しログ中に出現するマルウェアサンプルファイル名を XXXXXX に変更した．

(2) ログ作成日時が異なるため，各イベントのタイムスタンプを削除した．

(3) RegCloseKey(), FindClose(), fclose(), FileClose() といったクローズ系の関数は，亜種判定上不要とみなし削除した．

(4) FindFirstFile(), WriteFile(), DsRole(), NetShare(), NetUser(), NetApiBufferFree(), FindNext(), FindClose(), DsRole(), DsOpenFile(), LZOpenFile(), LZCopy()から不要な引数を削除した．

(5) 戻り値が負数の場合は FAIL，100 以上の場合は SUCCESS に変更した．

3.1. API 呼び出し比較による亜種判定

指定したマルウェアの API 呼び出しログと他のマルウェアの API 呼び出しログを比較した結果により，指定したマルウェアの亜種を探す．

ログの比較は行単位に行うので，前述のログの加工処理により，メモリアドレス，実行ファイル名，時刻といった実行環境に依存する値を除いた加工後のログを比較に使用する．

比較結果として，一致する行数の割合，連続一致最大行数，関数一覧の情報を得る．

2 つの API 呼び出しログを行単位に比較し，一致する行数を求める．比較元の総行数のうち，一致する行数の割合を得る．行の比較は順序を考慮する．

一致行を比較する場合，一致した行数だけでなく，連続して一致した行数の最大数も取得する．一致する行数の割合が低くても連続した一致行の数が多い場合，マルウェアのうち一部を流用したことが考えられる．

比較する 2 つログの使用関数一覧を作成し，関数ごとの呼び出し回数と一致した回数のリストを得る．一致する行数の割合と連続一致最大行数は行単位の比較であり，引数や戻り値も含めてすべて一致する行が対象であるため，引数の一部が異なるために一致する行が少なくなってしまうことがある．こういったケースにおいても，使用している関数の傾向が同じ場合に亜種と判定できる場合もあるため関数一覧を作成する．

(1) 表 2 は BKDR_IRCBOT.RB とその他のログを比較した結果の一部である．一致する行数の割合が 80% を超えるものは同一のマルウェアと判断できる．

表 2 BKDR_IRCBOT.RB とその他のマルウェアの比較結果

マルウェア名	ログ行数	一致行数割合(%)	連続一致最大行数
WORM_SDBOT.BAF	616	93	200
WORM_RBOT.BOM	621	92	200
WORM_SDBOT.EAB	607	91	200
WORM_RBOT.CJQ	613	89	200
WORM_SDBOT.CTR	617	89	193

一致する行数の割合が 93% である WARM_SDBOT.BAF と BKDR_IRCBOT.RB のログをさらに詳しく見てみると，一致しない行でも引数の一部が異なっているだけのものが多いことがわかった．例えば，algs.exe が explorer.exe，xygketr.bat が itsqr.bat に置き換わっている（表 3）．これは，ファイル名ををランダムに生成する処理によるものと考えられる．

表 3 BKDR_IRCBOT.RB と WARM_SDBOT.BAF のログの diff コマンドの出力結果の抜粋

```
43,46c43,47
< XXXXXX.exe, File, DeleteFileA, [C:¥WINDOWS¥system32¥algs.exe], 0
< XXXXXX.exe, File, CopyFileA, [c:¥vemu¥virus¥XXXXXX.exe][C:¥WINDOWS¥system32¥algs.exe][0], 1
< XXXXXX.exe, File, SetFileAttributesA, [C:¥WINDOWS¥system32¥algs.exe][4], 1
< XXXXXX.exe, File, SetFileAttributesA, [C:¥WINDOWS¥system32¥algs.exe][2], 1
---
> XXXXXX.exe, File, DeleteFileA, [C:¥WINDOWS¥system32¥explorer.exe], 0
> XXXXXX.exe, File, CopyFileA, [c:¥vemu¥virus¥XXXXXX.exe][C:¥WINDOWS¥system32¥explorer.exe][0], 1
> XXXXXX.exe, API, LoadLibraryExA, [C:¥WINDOWS¥system32¥explorer.exe], SUCCESS
> XXXXXX.exe, File, SetFileAttributesA, [C:¥WINDOWS¥system32¥explorer.exe][4], 1
> XXXXXX.exe, File, SetFileAttributesA, [C:¥WINDOWS¥system32¥explorer.exe][2], 1
48,53c49,54
< XXXXXX.exe, File, CreateFileA(exists), [C:¥WINDOWS¥system32¥algs.exe], 12
< XXXXXX.exe, File, SetFileTime, [C:¥WINDOWS¥system32¥algs.exe], 1
< XXXXXX.exe, File, DeleteFileA, [xygketr.bat], 0
< XXXXXX.exe, File, CreateFileA(new), [xygketr.bat], 12
< XXXXXX.exe, File, WriteFile, [xygketr.bat], 1
< XXXXXX.exe, File, Write_Data, [xygketr.bat][@echo off :deleteagain del /A:H /F XXXXXX.exe del /F XXXXXX.exe if exist XXXXXX.exe goto deleteagain del xygketr.bat ][e1][12e940][0], 1
---
> XXXXXX.exe, File, CreateFileA(exists), [C:¥WINDOWS¥system32¥explorer.exe], 12
> XXXXXX.exe, File, SetFileTime, [C:¥WINDOWS¥system32¥explorer.exe], 1
> XXXXXX.exe, File, DeleteFileA, [itsqr.bat], 0
> XXXXXX.exe, File, CreateFileA(new), [itsqr.bat], 12
> XXXXXX.exe, File, WriteFile, [itsqr.bat], 1
> XXXXXX.exe, File, Write_Data, [itsqr.bat][@echo off :deleteagain del /A:H /F XXXXXX.exe del /F XXXXXX.exe if exist XXXXXX.exe goto deleteagain del itsqr.bat ][df][12e940][0], 1
99c100
:
```

(2) 表 4 は WORM_SDBOT.AUV とその他のログを比較した結果の一部である．この結果のうち，一致する行数の割合，連続一致最大行数，関数一覧の 3 点から総合的に判定を行う．

表 4 WORM_SDBOT.AUV とその他のマルウェアの比較結果

マルウェア名	ログ行数	一致行数割合(%)	連続一致最大行数
WORM_RBOT.CNY	846	87	266
WORM_RBOT.BCE	846	85	298
WORM_SDBOT.BCB	862	57	216
BKDR_Generic	877	57	205
WORM_SDBOT.BKL	877	57	205
WORM_SDBOT.CTM	862	57	203
WORM_RBOT.CBJ	847	23	197
WORM_RBOT.ANN	2777	23	190
WORM_RBOT.CE	2949	23	190

表 4 をもとに下表を作成した．一致する行数の割合が 80%以上を高，50%以上を中，50%未満を低，連続一致最大行数が 200 以上を高，100 以上を中，100 未満を低とした．また，関数一覧は目視により，高，中，低と判定した．

表 5 WORM_SDBOT.AUV の亜種判定結果

マルウェア名	一致行数	連続一致最大行数	関数一覧	判定結果
WORM_RBOT.CNY	高	高	高	同一
WORM_RBOT.BCE	高	高	高	同一
WORM_SDBOT.BCB	中	高	高	同一
BKDR_Generic	中	高	高	同一
WORM_SDBOT.BKL	中	高	高	同一
WORM_SDBOT.CTM	中	高	高	同一
WORM_RBOT.CBJ	低	中	高	同一
WORM_RBOT.ANN	低	中	中	不明
WORM_RBOT.CE	低	中	高	亜種

(3) シマンテック社，トレンドマイクロ社，マカフィー社のアンチウイルスソフトで検出されなかったマルウェアサンプルのログと既知のマルウェアのログを比較してみた（表 6）．この結果から，一致する行数の割合が 98%の WORM_IRCBOT.RI と一致する行数の

割合が 96%の WORM_RBOT.ATU とは同一のマルウェアであると判断できる.

また, WORM_RANDEX.BG は, 連続一致最大行数が多く, 関数一覧の傾向が同じことから, 亜種と判定できる.

表 6 未検出ファイルとその他のログの比較結果

マルウェア名	ログ行数	一致行数割合(%)	連続一致最大行数
WORM_IRCBOT.RI	738	98	227
WORM_RBOT.ATU	738	96	222
WORM_RANDEX.BG	801	28	196
WORM_SDBOT.APS	437	27	188
WORM_SDBOT.ARK	449	27	188

本実験で一致する行数の割合の基準となっているのは, ログファイルを行単位で比較して一致した行数の数である. ログファイルの比較は, 1 行が完全に一致することが前提となっているため, 実行ファイル名や引数の一部が異なると全く一致しない場合と同じ結果になってしまう. 行の一致の判断にファジー比較のアルゴリズムを追加することにより, 精度の高い結果が得られるようになるだろう.

また, 関数一覧を目視で判定したが, 自動的に判定するアルゴリズムが必要である.

3.2. 処理のブロック化による亜種判定

API 呼び出しログに含まれる関数や引数からマルウェア特有の部分を加工し, 可能な限り処理のブロック化を行う. ブロック化とは, ファイルの操作やレジストリの操作など, ある程度同一の処理を行っている箇所を一行にまとめることである. 加工後のログを比較することによって, API 呼び出しとその引数の単純な比較ではなくより確率の高い判定ができるかを考察する.

ログの加工は, 次の点を考慮し, 亜種判定に関連の無い情報をログファイルから削除する.

- API 種別が変わる箇所で処理が追加される部分が多い. API 種別とは, API をファイル関連, レジストリ関連, ネットワーク関連, その他に分類したものである.
- 亜種の場合, ファイル操作で共通する部分がある.
- レジストリの操作はある程度まとまって行われる.
- 自分自身のコピーや作成するバッチファイルのファイル名は亜種ごとに異なるが, それを作成するフォルダはある程度共通する.

ログファイルの加工を行った後, ファイル単位で比較を行った. ログファイルの比較のために, 専用のプログラムを開発した. このプログラムは, 比較元のログファイルを一行ずつ読み込み, 比較対象のログファイルに一致する行があるかを調べる. プログラムは比較元のログファイルで何パーセントの行が一致したか表示する. また, 検証用の比較ツールとして, Araxis 社製の Araxis Merge を利用した.

(1) BKDR_IRCBOT.RB のログと他のマルウェアのログを比較し, 一致している割合が高いものを下表にまとめた.

表 7 一致した割合が高かったマルウェア

マルウェア名	一致割合 (%)
WORM_SDBOT.CTR	99
WORM_SDBOT.BAF	99
WORM_SPYBOT.AE	99
WORM_RBOT.BOM	99
WORM_IRCBOT.QD	99
WORM_POEBOT.HF	98
WORM_LINKBOT.H	98
WORM_SDBOT.EAB	98
WORM_RBOT.GZ	98
WORM_POEBOT.DC	98

一致している割合が 99%と極めて高い WORM_SDBOT.CTR, WORM_SDBOT.BAF などは同種のマルウェアと判断できる.

比較ツールを使用して, BKDR_IRCBOT.RB と WORM_SDBOT.CTR のログファイルを比較した結果, 異なる行は 3 行のみであった. 同じく, BKDR_IRCBOT.RB と WORM_SDBOT.BAF との比較では異なる行は 4 行のみであった.

(2) ログのファイルサイズが大きい WORM_RBOT.GEN と他のログファイルを比較し, 一致している割合が高いものを下表にまとめた.

表 8 WORM_IRCBOT.RB と他のマルウェアの比較結果

マルウェア名	一致割合 (%)
WORM_RBOT.AWO	98
WORM_RBOT.EHI	97
WORM_RBOT.DVL	97
WORM_RBOT.APW	97
WORM_RBOT.BAV	96
WORM_RBOT.EIC	96
WORM_RBOT.DJR	96
WORM_RBOT.CWT	96
WORM_RBOT.BCF	95
WORM_RBOT.BED	83

比較の結果，WORM_RBOT.GEN と一致する割合が高い WORM_RBOT.AWO, WORM_RBOT.EHI は亜種と判断できる．

(3) どのベンダでも検出できなかったマルウェアのログと検出されたマルウェアのログを比較した．

表 9 未知のマルウェアと検出済みマルウェアの比較結果

マルウェア名	一致割合 (%)
WORM_RBOT.ATU	99
WORM_IRCBOT.RI	99
WORM_RANDEX.BG	96
WORM_RBOT.BAX	90
WORM_RBOT.ASQ	90
WORM_SDBOT.AXL	89
WORM_RBOT.CLP	89
WORM_RBOT.BKC	89
WORM_SDBOT.CRT	88
WORM_SDBOT.ALB	88

この結果，WORM_RBOT.ATU と WORM_IRCBOT.R で一致する行数の割合が非常に高く，この場合，同種と判断できる．また，一致する行数の割合が 90%以上の WORM_RANDEX.BG ， WORM_RBOT.BAX ， WORM_RBOT.ASQ も同種と考えられる．

本実験では，API 種別と引数から API 呼び出しログのブロック化を行ったが，さらに検出効率が向上するようなアルゴリズムを検討する必要がある．具体的には次の処理を認識し，ブロック化する．

- ループの検出
- 特徴のある処理の検出
- プログラムの起動
- ダイナミックリンクライブラリの読み込みに関連する処理
- レジストリを操作する処理

また，ログの比較に行単位で判断するプログラムを作成したが，プログラムで比較した結果では高い確率で一致したログでも，目視で比較すると，かなりの差異があり，亜種とは判定できないものが存在した．ログの比較には単純な比較ではなく，高精度な比較アルゴリズムを使う必要がある．

3.3. 視覚化による亜種判定

API 呼び出しログの API 種別と関数名に着目し，API 種別と関数の種類毎にそれぞれ着色してログの視覚化を行う．視覚化されたログを目視によって亜種の判定が可能かどうかを考察する．

視覚化したログを判別しやすいように加工した後，3 種類のイメージを並べて比較した．

ログの加工には マイクロソフト社の Excel のマクロ機能を利用した．また，マルウェアの名称はトレンドマイクロ社のアンチウイルスソフトの検出名である．

(1) WORM_SDBOT ファミリに属するマルウェアのログを比較した結果，明らかにイメージが異なるものが存在した（図 1）．

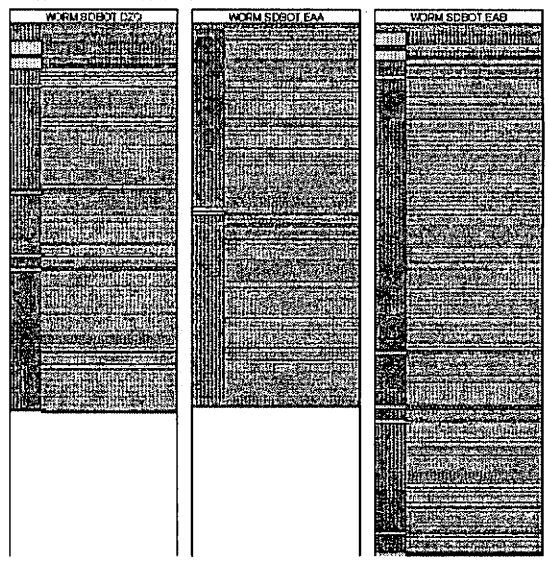

図 1 WORM_SDBOT ファミリに属するマルウェアの比較結果

(2) 異なる検出名のマルウェアを比較した結果，明らかに似たイメージが存在した（図 2）．

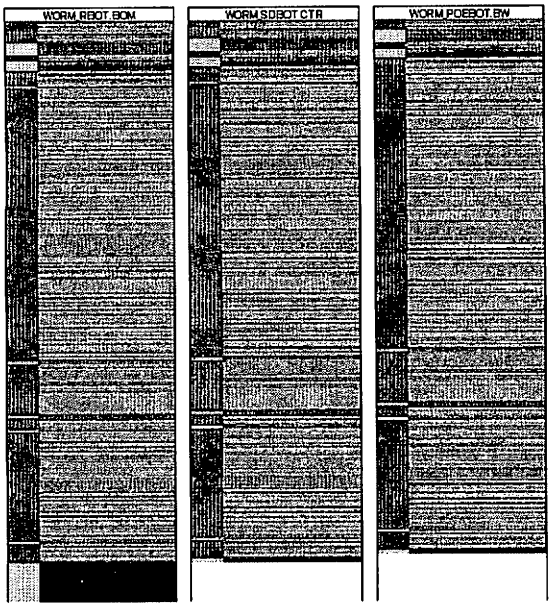

図 2 異なるファミリに属するマルウェアの比較結果

(3) 亜種判定を目視で行っているため，イメージが酷似していなければ識別できない．例えば，図 3 の場合，いくつか類似する箇所はあるが，同種と判断するのは困難である．

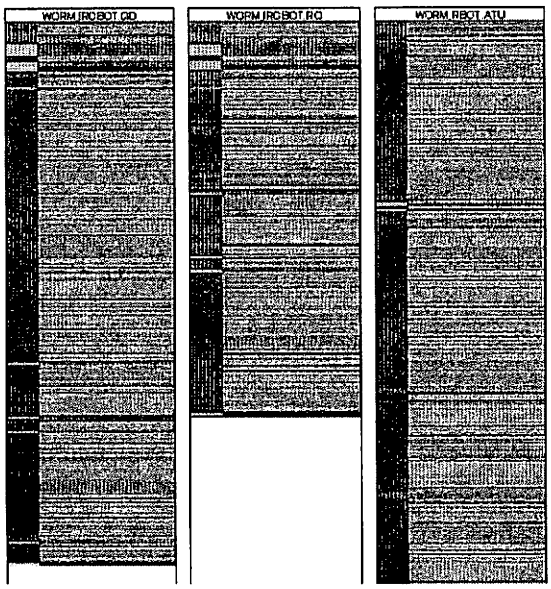

図 3 任意に選択した 3 つのマルウェアの比較結果

(4) 既知のマルウェアのイメージから未知のマルウェアのイメージと類似しているもの探した（図 4）．WORM_IRCBOT.RI と酷似している．WORK_SPYBOT.GEN も前半は酷似しているが，全体的に見ると異なるものであることがわかる．

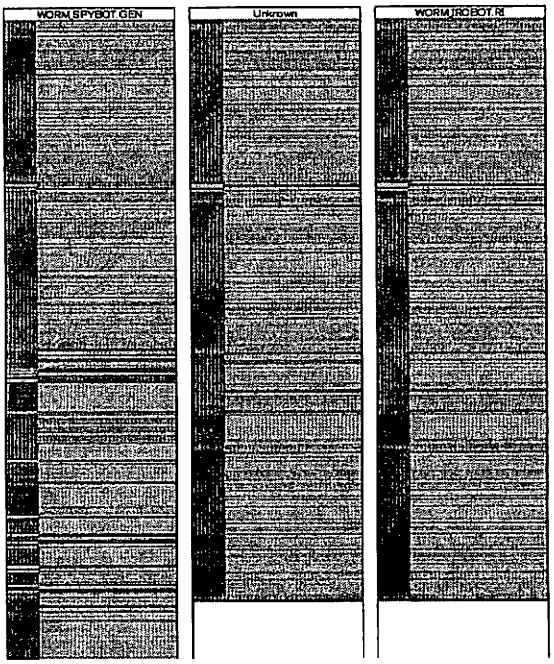

図 4 未知のマルウェアのイメージと類似する既知のマルウェア

同種に分類されているマルウェアでも，今回のような比較することによって分類が間違っていることを簡単に認識することができた．ただし，異種判別を視覚に頼っているため，特徴のあるログではないと判定が難しい．また，未知のマルウェアを調べる作業はまるで神経衰弱のように困難な作業であった．

今回は，API 種別と関数の種類毎にそれぞれ着色してログの視覚化を行ったが，別の方法で視覚化することも検討したい．また，API 呼び出しログの視覚化は，マルウェアの分類だけでなく，マルウェアの機能を容易に知るという目的でも利用できそうである．

4. まとめ

マルウェアが実行時に呼び出すWindows APIの種類やタイミングなどに着目し，マルウェアを分類することができるかどうかを検討した．提案方式を適用した結果，BKDR_VANBOT.S と BKDR_SPYBOT.WQ はほぼ同一であるが，トレンドマイクロ社では両者を異なるファミリに分類していることが判明した．

マルウェアの分類を行う上で，動的解析のみによる変種・亜種分類は完全な解決策ではないが，アンチウイルスベンダのエキスパートが人的努力によって行った分類と比較しても，より効果的な手法であることが証明された．

今後は，API 呼び出しログだけでなく，ファイル属性やプログラムコードから得られる情報を利用するなどして新たなアルゴリズムを検討し，より効果的な方法を提案していきたい．

5. 謝辞

本研究は平成 18 年度に総務省から委託を受け実施した「スパムメールやフィッシング等サイバー攻撃の停止に向けた試行」の成果の一部である．

本研究を進めるにあたって有益な助言と協力を頂いた関係者各位に深く感謝致します．