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

2008－CSEC－41 (11)
2008／5／23

ハミング距離によるマルウェア亜種の自動分類

堀合 啓一*† 今泉 隆文* 田中 英彦†

†情報セキュリティ大学院大学情報セキュリティ研究科〒221-0835, 神奈川県横浜市神奈川区鶴屋町 2-14-1
*防衛省技術研究本部電子装備研究所 〒154-8511, 東京都世田谷区池尻 1-2-24

あらまし マルウェアの動的な挙動を多次元のベクトルとして数値化し，ベクトル間の距離からマルウェアの亜種を判定する手法について提案する．マルウェアの実行に伴って顕在化したプロセスの起動やファイルの変化及び発生するトラフィックなどの出現頻度を基に２値のカテゴリ・データとして数値化し，距離の算出は，高速に演算が可能なハミング距離を用いる．提案手法の有用性を実験によって検証した．

キーワード マルウェア，分類，ハミング距離，正規化圧縮距離，ROC 分析

Automatic Malware Variant Classification by Hamming Distance.

Keiichi Horiai*† Takafumi Imaizumi* Hidehiko Tanaka†

†INSTITUTE of INFORMATION SECURITY, 2-14-1 Tsuruyacho, Kanagawaku, Yokohama, 221-0835 Japan
*Technical Research and Development Institute, Ministry of Defense 1-2-24 Ikejiri, Setagaya, Tokyo, 154-8511 Japan

Abstract This paper describes a method for malware variant classification using the distance between multi-dimensional vectors which represent the dynamic behavior of the malware. The dynamic behavior, involving state changes of processes, files, and network traffic, is transformed into binary category data. The measurement of the distance is based on Hamming distance, which requires less processing power than measurements proposed in previous works. The proposed method is proved its efficiency by our experiments.

Keyword Malware, Classification, Hamming Distance, Normalized Compression Distance, ROC analysis

1 はじめに

近年のマルウェアは．開発ツールの流通や難読化・暗号化などの一般化よって，非常に多くの種類が出現しパター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．マルウェアによる被害を局限するためには，できるだけ早期に検知して対策を行うことが必要である．このため，筆者らは，定点観測の手法でマルウェアを収集し，捕獲したマルウェアの動的な挙動を自動的に解析するシステムを構築しているが，捕獲したマルウェアの約 20% 程度は，市販のウィルス対策ソフトウェアではマルウェアとして検出できない[10]．その後，しばらくしてウィ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．捕獲した時点では，市販のウィルス対策ソフトウェアでマルウェアとして認識できない場合でも，マルウェアを実行して得られる，プロセスの起動，レジストリの改ざん，通信パケット等の情報を，既知のマルウェアの挙動と比較し，その類似性からマルウェアの名称を推定できれば，さらに詳細な解析を行う場合や，対策を検討する際に，既知の情報

を利用できる可能性がある.

本研究は,マルウェアの動的な挙動を解析して得られる情報を利用し,捕獲した段階では未知のマルウェアについて,既知のマルウェアとの類似性を自動的に算出し,その名称を推定する手法の提案である.以下,次章では関連研究の状況,3章で提案手法を説明し,4章では実験結果及び考察を述べて,5章でまとめる.

2 関連研究

マルウェアの自動解析手法として,[1-6,11-14]などが知られている.これらは静的解析と動的解析の2種類に大別できる.文献[1]は,マルウェアのバイナリ・ファイルを対象とした自己組織化マップ(SOM: Self Organizing Map)による視覚化である.文献[2]では,バイナリ・ファイルを対象として,ファイルのバイト列をn-gramで表現し,マルウェアと正常なプログラムの分類を行った実験についての報告である.また,文献[3]は,マルウェアのバイナリ・ファイルを対象として,正規化圧縮距離(NCD: Normalized Compression Distance)[7]で相互の距離を算出して分類する手法を提案している.

一方,文献[4]は,動的解析の例であり,マルウェアを実行した際に記録したAPI CALLを解析の対象としてAPI呼び出しが一致する行数の割合,連続一致最大行数,関数の一覧などを比較して亜種の判定を試みている.文献[5]はマルウェアを実行して,記録したイベント・シーケンスを解析の対象とし,類似性の判定には編集距離を利用している.文献[6]では,マルウェアを実行して記録した動的な挙動を解析の対象とし,類似性の判定としてはNCDや単連結階層クラスタリングなどを利用して,分類すべきマルウェアの数とメモリーの所要量,処理時間の関係等を分析し,さらに既存のマルウェアの分類に依存しない,動的挙動に基づく新たなクラスタリング手法を提案している.

本研究は,NCDよりも計算コストが低いハミング距離(HMD: Hamming Distance)を利用したマルウェアの分類手法の提案である.また,文献[6]とは異なり,既知のマルウェアの挙動との類似性を基にして分類することから,蓄積したマルウェアのプロファイルと挙動が類似しているマルウェアの名称を推定することが可能となり,コードレベルのようなさらに詳細な分析を行う場合や,マルウェアの対策を検討する際に,既存の知見を活用できるので,未知のマルウェアへの対応を効率化することが可能となる.

3 提案手法

本研究はマルウェアを仮想マシン上のWindowsXP上で実行し,マルウェアの動的挙動を解析して得られる情報を使って自動分類を行うものである.マルウェアの動的挙動解析では,図1にその一例を示す各項目が文字列の情報としてBDB(Behavior Data Base)に蓄積される.

```
[HASH] マルウェア・ファイルのハッシュ値.
 04999957e3c78e03737cd55a61a7f3ca
[REGISTRY] レジストリファイルの変化
HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
c:¥windows¥system32¥logon.exe
[MD5SUM] Windowsシステム関連ファイルの変化
 Created C:¥WINDOWS¥SYSTEM32¥LOGON.EXE
[PROCESS] 検知したプロセスの起動・停止
 winlogon, services, logon
[HOSTS] Windows HOSTSファイルの変化.
 No change Found.
[ROOT KIT] ルートキットの検出結果
 Not Detected
[SERVCIES] 検知したサービスの起動・停止
 No change Found.
[TRAFFIC] 観測したトラフィックのポート番号
 PORT(2), domain(2), 8998(16)
[MALWARE CLASSFICATION] ウィルススキャンの結果
 C:Trojan.Lineage-80, T:Unknown, S:W32.IRCBot,
 K:Trojan-PSW.Win32.Nilage.zh
```

図1 BDB(Behavior Data Base)の一例

3.1 分類の手順

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.

本論文は,このBDBの各要素を利用して,マルウェアを自動的に分類する手法の提案である.本提案によるマルウェアの分類の手順を以下に示す.

[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ウィルス対策製品でスキャンし，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（図2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をカテゴリ・データへ変換し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PBP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固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，種類毎の固体数が多い種類を候補として出力する．提案手法の全体ブロック構成を図2に示す．

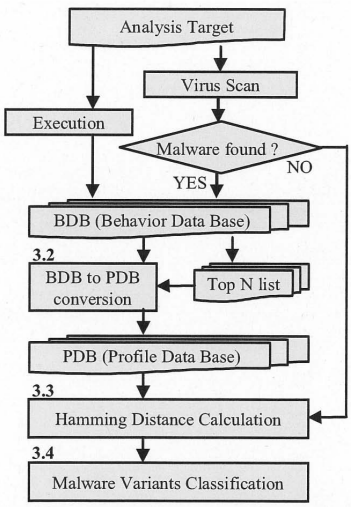

図2　マルウェア分類処理の全体構成図

以下，提案手法の中心であるBDBからPDBへの変換及び距離の算出とこれに基づくマルウェアの分類について述べる．

3.2　BDPからPDBへの変換

マルウェ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た．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．例えば，レジストリの変化については，**R**={$r_1, r_2, r_3, \cdots r_n, r_{n+1}$}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．この方法では，PDBとして利用する要素の種類，その組合せ，及びNの値（＝カテゴリの種類数）などが分類の精度に影響を与えると考えられるが，これについては次章で述べる実験で検証した．本提案におけるPDBの一例を図3に示す．

```
# hash_value_of_Malware_binary   categorized_data
049c244101170faa7c743e1ac5494ede 0001000000000600
04af7239845601e9d785a7824b6ca34e 0000001200000600
04cb88703c78a3ffa6f678a9a77c66c7 0000000008002740
04ecc1d94e27cd9f8822dc69b8e78d8a 1400000000000700
0502329d4e1c5bf4fe370e3a9e246454 0000000000002600
052b23dd1320692f6508e7c24b519d0e 8000008000000e00
05534778b7e1652237be1ca9497bd230 8000000100000e00
```

図3　PDB(Profile Data Base)の一例

3.3　距離の算出

本提案では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する．ハミング距離(HMD:Hamming Distance)は，n次元のベクトル X=(x_1, x_2, …, x_n)，Y = (y_1, y_2, …, y_n) において要素が等しくない個数であり次式で表現される．

$$HMD(X,Y) = \sum_i^n \delta(Xi,Yi) \qquad (1)$$

ただし，$\delta(X,Y)$ は X=Y なら 0，そうでなければ 1 をとる関数である．各要素の値は２値の場合が多いが，多値の場合でもかまわない．特に X,Y が２値の場合には排他的論理和(XOR)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が得られる点に特徴がある．

提案のハミング距離を利用した分類の精度を比較する意味で，文献[3,6]が利用している正規化圧縮距離についても検証を行った．

3.4 未知のマルウェアの分類

未知のマルウェアを分類する，実践フェーズでは，分類対象のプロファイルと蓄積した PDB 内のすべてのプロファイル間の距離を算出し，距離が最短となるプロファイルを持ったマルウェアとして分類する．ただし，距離が最短のマルウェアが複数種類存在する場合には，データベース内のそれぞれの種類毎にハッシュ値が異なるマルウェアの検体の数をカウントし，その数をスコアとして，スコアが最大となる種類として分類する．

4 実験結果及び考察

本章では，実験に利用したデータ，実験の手順及び実験結果の説明とその考察を行う．分類の精度を評価する指標として，不正侵入検知システムの評価などによく使われている ROC(Receiver Operating Characteristic)分析[8]を利用した．ROC 分析における AUC(Area Under Curve)値を指標として，良好な分類結果を得るための要素毎のトップ N の値や要素の組合せを検討した．

4.1 実験データ

BDB から PDB へ変換したハッシュ値を異にするマルウェアの個体数は 3,665 個である．これらの検体は，インターネットへ接続したハニーポットで捕獲したものであるが，その大半はボット関連のマルウェアである．これらのマルウェアを市販の４種類のウィルス対策製品でスキャンして判明したマルウェアの種類を表１に示す．

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，この表から製品によって種類数が大きく異なっていることがわかる．文献[4,6]においても，あるマルウェアの固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．このような製品による名称の違いの影響を低減するために，以下の実験では 3,665 個の中から一定の手順で抽出した 1,503 個を使用した．またマルウェアの名称は，実験対象の検体について４製品の中では最も多くの種類として分類したウィルスバスター2007[9]でスキャンした結果を基準とした．

次に，マルウェアを実行した前後に何らかの挙動の変化が観測された割合を表２に示す．この表から，**P**，**R**，**M**，**T** トラッフィクの各項目については，多くのマルウェアで挙動の変化が観測されていることから，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，これらの要素を主要素と呼ぶ．また，主要素以外の **H**,**S**,**K** については，マルウェ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ェ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り，これらについてはそれぞれの要素について変化の有無を{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呼ぶ．

表１　実験対象としたマルウェアの製品別種類数

	T	K	S	C
Number of Variants	1,095	400	65	765
Number of Families	95	63	35	74

T: TrendMicro VirusBuster2007　S: Symantec Internet Security 2007
K: Kaspersky Internet Security 6.0　C: ClamAV 0.91

表２　動的挙動の変化を観測できた要素毎の割合

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

4.2 分類精度の評価方法

ハミング距離によるマルウェア分類の判定精度を評価するため，ROC 分析における AUC 値を利用した．このため，分類対象のマルウェアの PDB から相互のハミング距離を算出して距離マトリクスを生成する．一例として３種類のマルウェアをそれぞれ３個，合計９個の距離マトリクスの例を図４に示す．

この例では，距離７が最適な閾値であるが，同種類を異種と誤判定する組合せを２件含んでいる．ROC 分析では，同種を同種と判定(TP: True Positive)した割合(TPR:TP Rate)と異種を同種と判定(FP: False Positive)した割合(FPR:FP Rate)について，閾値をパラメータとして曲線を描画し，この曲線の下側の面積を AUC 値として，この面積の大小で分類の性能を評価した．実験で得た ROC 曲線の一例を図５に示す．

	A_1	A_2	A_3	B_1	B_2	B_3	C_1	C_2	C_3
A_1	-	4	5	10	10	9	11	11	11
A_2	4	-	5	10	10	13	11	11	11
A_3	5	5	-	11	11	14	12	12	12
B_1	10	10	11	-	4	11	15	15	15
B_2	10	10	11	4	-	9	15	15	15
B_3	9	13	14	11	9	-	10	10	10
C_1	11	11	12	15	15	10	-	2	2
C_2	11	11	12	15	15	10	2	-	0
C_3	11	11	12	15	15	10	2	0	-

図４　距離マトリクスの一例

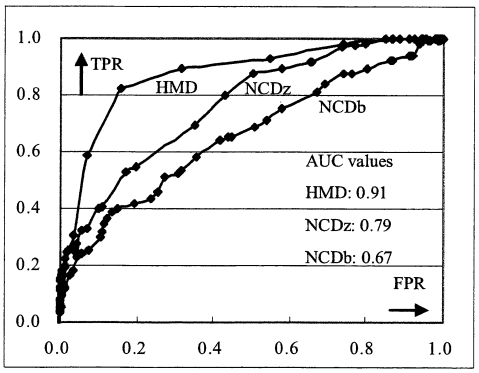

図５　実験で得た ROC 曲線の一例と AUC 値

4.3　実験の手順

最初に，4.1 節で述べた主要素について，それぞれを単独にカテゴリ・データとした場合の評価を行った．この際，出現頻度トップ N における最適な N を得るため N= {10, 20, 50, 100} について評価を行い，この中から最適な N を求めた．続いて，主要素を中心として複数の要素を組合せた場合の評価を行った．

実験の手順は，次のとおりである．

[STEP-1] 基準データの中から，10 種類のマルウェアについて，それぞれの種類からランダムに 10 個ずつ，合計 100 個の固体を抽出

[STEP-2] STEP-1 で抽出した合計 100 個について相互のマルウェアの距離マトリクスを算出し，距離の最小値(Rmin)と最大値(Rmax)を得る．距離の算出は比較のためハミング距離と NCD の２種類で行った．

[STEP-3] 同種類として判断する閾値 TH を Rmin から Rmax まで変化させて，TPR(TH)，FPR(TH)を得る

[STEP-4] STEP-3 の結果から，TH をパラメータとした ROC 曲線をプロットし，この図から AUC 値を算出する．

以上のステップを 20 回繰り返し，AUC 値の平均値を求めて，分類精度の指標とした．

4.4　要素単体による分類

前節で述べた手順により，要素単体で分類した場合の AUC 値の平均値を表３に示す(NCDz は NCD における圧縮アルゴリズム zlib，NCDb は bzlib を表す)．出現頻度トップ N における N の選び方によって，AUC 値が変化するが，**R** 及び **M** については N=50，**T** については N=100 の場合に AUC 値が 0.83 で最大となっている．**P** による分類では，他の主要素と比較して AUC 値が低い値となり，分類の主要素としては適さないことが判明した．距離算出の手法による違いとしては，HMD の方が NCD よりも高い AUC 値となった．

4.5　複数要素の組合せによる分類

AUC 値をさらに高めるために，複数の要素を組合せた場合の実験を行った．**R**, **R**, **T** について単一要素の場合に最大の AUC 値となった N を用いて，**K**, **S**, **H** の各補助要素を順次追加して，AUC 値を算出した．

その結果を表４に示すが，**R**,**M**,**T**ともに，補助要素をPDBへ追加することによって，逐次AUC値が増加し，分類の精度が向上することがわかる．また，これらの中では**R**と３種類の補助要素を組合せた場合にAUC値が0.89で最大となった．さらに距離の算出法についてHMDとNCDを比較すると， HMDの方が常に高いAUC値となった．

表３　要素単体によるAUC値

R	N10	N20	N50	N100
HMD	0.68	0.75	0.82	0.81
NCDz	0.67	0.72	0.81	0.82
NCDb	0.70	0.72	0.78	0.72

M	N10	N20	N50	N100
HMD	0.72	0.80	0.86	0.85
NCDz	0.64	0.75	0.76	0.77
NCDb	0.64	0.68	0.78	0.75

T	N10	N20	N50	N100
HMD	0.80	0.82	0.82	0.83
NCDz	0.53	0.57	0.59	0.61
NCDb	0.70	0.70	0.74	0.70

P	N10	N20	N50	N100
HMD	0.66	0.65	0.68	0.68
NCDz	0.58	0.36	0.69	0.65
NCDb	0.60	0.63	0.66	0.68

表４　要素単体と補助要素によるAUC値

		C1	C2	C3	C4
M	HMD	0.86	0.85	0.85	0.88
	NCDz	0.76	0.73	0.72	0.76
	NCDb	0.78	0.74	0.69	0.70
T	HMD	0.83	0.84	0.85	0.86
	NCDz	0.61	0.58	0.60	0.67
	NCDb	0.70	0.69	0.69	0.72
R	HMD	0.82	0.86	0.86	0.89
	NCDz	0.81	0.84	0.81	0.85
	NCDb	0.78	0.81	0.79	0.82

C1: Single major element　　C3: C2 & root**K**it
C2:C1 & **S**ervice　　C4: C3 & **H**osts

5　おわりに

マルウェアを仮想マシン上のWindows環境で実行し，取得したマルウェアの動的挙動に関する情報を含むログからカテゴリ・データへ変換して相互のハミング距離を算出することによって，マルウェアの種類を推定する手法についての提案を行い，その有効性を実験によって検証した．本提案の手法は，NCDよりも計算量が少なく，かつ分類の精度も高いことから，マルウェア分類の自動化や対策立案のための解析の初期フェーズに有効と考えられる．

参考文献

[1] InSeon Yoo, Visualizing Windows Executable Viruses Using Self-Organizing Maps, VizSEC/DMSEC'04, Oct 29, 2004

[2] J. Zico Kolter, Marcus A. Maloof, Learning to Detect and Classify Malicious Executables in the Wild., Journal of Machine Learning Research 7(2006)

[3] Stephanie Wehner, Analyzing Worms and Network Traffic using Compression. May 17, 2006

[4] 星沢　裕二, 太刀川　剛, 村山　元昭, マルウェア亜種等の分類の自動化2007-CSEC

[5] Tony Lee & Jigar J. Mody, Microsoft Corp., Behavioral Classification, EICAR Conference, May, 2006

[6] Michael Bailey, Jon Andersen, Z. Morley Mao, Farnam Jahanian, Jose Nazario, Automated Classification and Analysis of Internet Malware. Apr 26, 2007

[7] Paul Vitányi・(翻訳) 渡辺治, 「圧縮度にもとづいた汎用な類似度測定法」, 数理科学NO. 521, NOVEMBER 2006

[8] Peter Mell, Richard Lippmann, Josh Harines, Marc Zissman, An Overview of Issues in Testing Intrusion Detection Systems

[9] ウィルスバスター2007 http://jp.trendmicro.com/jp/home/index.html

[10] 堀合　啓一, 今泉　隆文, 田中　英彦, 定点観測によるボットネットの観測とMalwareの動的挙動解析システムの提案, 情報処理学会論文誌 Vol.49 No.4 Apr.2008

[11] 市川幸宏, 伊沢　亮一, 白石　善明, 森井昌克, メモリ上に展開されたコードを使うウイルス解析支援システム, 情報処理学会論文誌 Vol.47 No.8 Aug.2006

[12] http://www.cwsandbox.org/

[13] http://www.norman.com/microsites/nsic/

[14] http://www.cyber-ta.org/