L-003

マルウェアの類似度に基づく機能推定

Function Estimation for Malwares based on Similarity

大久保　諒 †　森井　昌克 †　伊沢　亮一 ‡　井上　大介 ‡　中尾　康二 ‡
Ryo Okubo　Masakatu Morii　Ryoichi Isawa　Daisuke Inoue　Koji Nakao

1　はじめに

マルウェアの解析において有効な手法として動的解析手法が挙げられる．動的解析手法では隔離された環境においてマルウェアを実際に動作させそのログを取ることにより挙動を出力する．実際に動作させるため，マルウェアの挙動に関する詳細な解析結果を得ることが可能である．一方で動的解析手法の欠点として解析に時間がかかる問題がある．マルウェアを実際に動作させることにより，解析自体にかかる時間に加え環境を再構築する時間が必要となるためである．

我々は本稿において短時間にマルウェアの概要を出力する手法を提案する．短時間に概要を取得することで解析すべきマルウェアを優先して解析することが可能となり，解析の効率化が図れる．提案手法は類似度を用いたマルウェアの機能推定である．我々は以前バイトコードの分布によるマルウェア間の類似度導出法を提案した [1]．本稿において以前に提案したバイトコードによる分布を用いた類似度と従来研究で用いられている Longest Common Substring(LCS)，*n*-gram を用いた類似度導出法をもとに機能推定を行い，考察する [2][3]．提案手法では，解析対象とするマルウェアと解析済の複数のマルウェア間の類似度を導出することによって機能ごとに解析対象としたマルウェアが保有するかを推定する．本稿の構成は以下の通りである．まず第 2 章において我々が以前に提案したバイトコードに基づく類似度導出法を紹介する．第 3 章では今回比較する類似度として LCS，*n*-gram について説明する．第 4 章において類似度を用いてマルウェアの機能を推定する手法を提案し，第 5 章において各類似度に基づく類似度を用いた機能推定の結果を示す．最後に第 6 章において本稿のまとめとする．

2　バイトコードの分布に基づく類似度導出

我々は以前バイトコードの分布に基づく類似度導出法を提案した．本章では exe ファイルや dll ファイルに用いられる PE ファイル構造について説明した後に類似度導出までの手順を示す．

2.1　PE ファイル構造

まず図 1 に PE ファイル構造を示す．ここで我々は section table に着目した．section table にはファイル中に含まれる各セクションの詳細な情報が含まれる．section table から得られる情報を表 1 に示す．機能推定を最終目的とするために実際に実行ファイルの動作に関わる部分での類似度を導出する必要があり，我々は characteristics

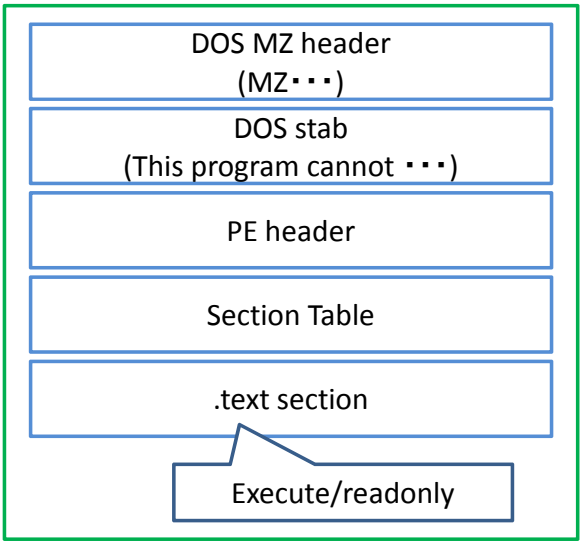

図 1　PE　File Structure

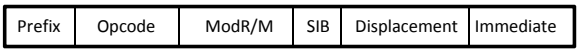

図 2　Instruction format

に着目することによって実行可能なセクションを特定し抽出した．図 2 は実行命令列の構造を示したものである．ここで動作に関わる部分は opcode 部であり，理想的にはこの部分のみを取得することによってマルウェアの機能面での類似性を測ることができると考えられる．しかし，逆アセンブルせずにバイナリの状態で解析することを目的としたため opcode 部である可能性が高い 0x0f の後のバイトコードに着目した．0x0f は 2byte コードに用いられ，そのあとに続くバイトは opcode に分類される．我々が以前提案した類似度導出法では実行可能なセクションを抽出し，そこから 0x0f の後のバイトコードの分布を取得し用いた．

2.2　正規化相互相関を用いた類似度導出

我々は正規化相互相関を用いることによってバイトコードの分布からマルウェア間の類似度を導出した．正規化相互相関により類似度 R を導出する式を式 1 に示す．ここで，N は対象とするバイトコードの数であり，$N = 256$ となる．I と T が比較するバイトコードの分布である．

$$R = \frac{\sum_{i=1}^{N-1}(I(i)-\bar{I})(T(i)-\bar{T})}{\sqrt{\sum_{i=1}^{N-1}(I(i)-\bar{I})^2 \times \sum_{i=1}^{N-1}(T(i)-\bar{T})^2}} \quad (1)$$

† 神戸大学大学院工学研究科, Graduate School of Engineering, Kobe University
‡ 独立行政法人情報通信研究機構, National Institute of Information and Communications Technology

表 1　Section Header

Size(byte)	Field
8	Name
4	VirtualSize
4	VirtualAddress
4	SizeOfRawData
4	PointerToRawData
4	PointerToRelocations
4	PointerToLinenumbers
2	NumberOfRelocations
2	NumberOfLinenumbers
4	Characteristics

出力される類似度は −1〜1 であり，1 に近いほど 2 検体のマルウェア間に類似性があると判断する．

3　類似度導出法

本研究では機能推定を行うための類似度導出法として前章で紹介したバイトコードの分布を用いた手法とともに LCS，*n*-gram に基づく手法を用いて比較を行った．本章において LCS および *n*-gram を用いた類似度導出法について説明する．これらの類似度導出法はバイトコードの分布から類似度を導出する際と同様に，実行可能なセクションのバイナリを対象とする．

3.1　LCS を用いた類似度導出法

LCS は比較する文字列のうち最長の共通した部分列を取得する手法である．例えば $A = abcd$，$B = adbc$ とした時，得られる LCS は $S_{A,B} = abc$ となる．LCS の長さ $L(S_{(A,B)})$ は式 2 を用いて導出することができる．ここで A, B は比較する文字列を示す．また，表 2 は例を式 2 に適用し，表に示したものである．ここの表からも例の LCS の長さが 3 であることが明らかである．

$$L(S_{(A_i,B_j)}) = \begin{cases} 0\ (if\ A = 0 \text{ or } B = 0) \\ \max(L(S_{(A_{i-1},B_j)}), L(S_{(A_i,B_{j-1})})) \\ \qquad (if\ A_i \neq B_i) \\ L(S_{(A_{i-1},B_{j-1})}) + 1\ (if\ A_i = B_i) \end{cases} \tag{2}$$

類似度を導出する際には Jaccard 係数を用いる．$L(A_1)$, $L(B_2)$ をそれぞれ文字列 A，B の長さ，$L(S_{A,B})$ を A と B の LCS の長さとして類似度 $J(A, B)$ を式 3 のように導出する．LCS を用いることにより 2 検体のマルウェア間で共通した部分のサイズを正確に導出することができるため LCS を用いて導出した類似度は精度の面で優位性があると考えられる．一方で LCS を用いた類似度導出では計算に時間がかかる問題がある．予備実験として Windows の電卓として用いられる calc.exe の実行可能なセクション (388KB) をそれ自身と比較した．この時 LCS の長さを導出するまでに 1300 秒程度を要することがわかった．

$$J(A, B) = \frac{L(S_{A,B})}{L(A) + L(B) - L(S_{A,B})} \tag{3}$$

表 2　LCS

	a	b	c	d
a	1	1	1	1
d	1	1	1	2
b	1	2	2	2
c	1	2	3	3

3.2　*n*-gram を用いた類似度導出法

n-gram は対象とす文字列を長さ n の要素に分割する手法である．例えば $s = abcdefg$ に 4-gram を適用することにより長さ 4 の要素の集合 $S = \{abcd, bcde, cdef, defg\}$ を得る．類似度 $J(S_1, S_2)$ を導出する際は LCS と同様に式 4 に示した Jaccard 係数を用いる．S_1，S_2 は比較する文字列の要素の集合である．比較対象とする 2 つの文字列に対して同様に *n*-gram を適用し，得られた要素のうち共通する要素がすべての要素に占める割合を導出することにより類似度とする．本稿では $n = 5$ として類似度を導出した．

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{4}$$

4　マルウェアの機能推定

本研究におけるマルウェアの機能推定においては情報通信研究機構 (NICT) が提供する NONSTOP のレポートを用いた．レポートは XML 形式で出力され，Action タグの内容を抽出することにより，対象の機能の一覧を取得することができる．ここから得た機能をマルウェアの機能として定義し，未知のマルウェアの機能を推定する手法を提案する．機能推定はマルウェア間の類似度に基づき実現した．機能推定の概要は以下のとおりである．

1. 解析済み検体のデータベースを作成する
2. 解析済み対象とするマルウェアをデータベースに格納されているマルウェア全てと比較し，類似度を導出する
3. 類似度をもとに機能ごとにポイントを付加する
4. 機能ごとのポイントを判別分析法により分類し，解析対象としたマルウェアが保有するであろう機能を推定する

本研究におけるマルウェアの機能推定では対象とする 1 検体のマルウェアにつき複数のマルウェアとの類似度を導出する必要があるため，類似度を高速に導出する必要がある．機能ごとに付加するポイントの値 $P(k)$ は式 5 により決定する．ここで，k は機能を示し，m は解析済みのマルウェアを示す．$r(m)$ は解析済み検体 m との類似度であり，$f(m, k)$ はもし解析済み検体 m が機能 k を保有する場合 1 であり，保有しない場合は 0 となる．M はデータベースに格納された検体の数を示す．表 3 は今回推定の対象とする機能の一覧である．ここで与えられる各々の sid に対してポイントを与え，機能を推定する．

$$P(k) = \frac{\sum_M |r(m)f(m,k)| r(m)f(m,k)}{\sqrt{\sum_M f(m,k)}} \tag{5}$$

表 3　Malware Functions

Action	sid
addReg	00000001
alterFile	00000006
alterProcess	04000002
alterTxtFile	00000015
attrFile	00000019
backdoor	06000001
backdoor	40000001
connect	06000013
connect	20000002
copyFile	00000004
copyFile	00200001
createFile	00000007
createFile	00600001
createMutex	01000001
createReg	00000002
createService	00300001
deleteFile	00700001
deleteReg	00000002
execute	00000005
newHash	00000017
openFile	00000019
openProcess	00000020
openwindow	02000001
readFile	00000018
search	00000009
search	00800001
showwindow	02000002

すべての機能に対して式 5 によりポイントを導出し，判別分析法により機能ごとのポイントを 2 値化することによって解析対象が持つであろう機能を推定する．判別分析法では式 6 により導出される r が最大になるようにしきい値 t を決定し，2 値化を行う．ここで用いられる $\sigma_B^2(t)$ および $\sigma_I^2(t)$ はそれぞれクラス間分散とクラス内分散の値を示し，式 7，8 により決定される．

$$r = \frac{\sigma_B^2}{\sigma_I^2} \tag{6}$$

$$\sigma_B^2(t) = \frac{\sum_{k=0}^{t-1} n_k(\overline{f_0} - \overline{f})^2 + \sum_{k=t}^{D} n_k(\overline{f_1} - \overline{f})^2}{\sum_{k=0}^{D} n_k} \tag{7}$$

$$\sigma_I^2(t) = \frac{\sum_{k=0}^{t-1} n_k(k - \overline{f_0})^2 + \sum_{k=t}^{D} n_k(k - \overline{f_1})^2}{\sum_{k=0}^{D} n_k} \tag{8}$$

5　機能推定結果

バイトコードの分布，LCS，*n*-gram のそれぞれに基づく類似度を導出し，マルウェアの機能推定を行った．本稿では CCC DATAset 2012 に含まれる検体を用いて実験を行った結果を示す．表 4 は本稿で用いた検体の一覧である．表 4 において NameByTrend は TrendMicro のスキャン結果による検体の科名を示し，Hash では各検体のハッシュ値上位 7 桁を示している．機能推定に用いる検体として TrendMicro により判定された同じ科名の検体を最大 5 検体として全体で 45 検体を抽出した．これらの検体が保有する機能は NICT の NONSTOP が導出したレポートと科名により紐付けし決定した．機能推定の実験ではこれらの検体全てについて機能を推定する．この時抽出した 45 検体全てを対象とし，各々の保有する機能をそれ以外の 44 検体から推定した．表 5 は 5815b13*の検体に対してバイトコードの分布より類似度を導出し機能ごとにポイントを導出した結果である．判別分析の結果ポイントのしきい値は 0.2204 であったため，下線を引いた機能を推定した．機能推定の結果を動的解析結果と比較すると機能 00000019 と 00300001 は False Positive であり，機能 00000015 は False Negative であった．

表 6 はすべての検体に対して機能推定を行った結果である．ここで A は推定した機能の集合を示し，B は実際にマルウェアが保有する機能を示す．表に示された $\frac{A \cap \overline{B}}{A}$ は推定した機能のうち動的解析結果からは確認されなかった機能の割合であり False Positive の割合であるといえる．$\frac{A \cap B}{B}$ は実際に検体が保持する機能のうち推定された機能の割合を示し，値が高いほど False Negative が少ないといえる．この結果から LCS とバイトコードの分布に基づく類似度を用いた場合の機能推定では全てにおいて *n*-gram に基づく類似度を用いた機能推定に比べて優位性があるといえる．LCS に基づく類似度では実際に保持しないにも関わらず保持すると推定した機能の割合がバイトコードに基づく類似度に比べて低く，この点においては優位性がある．一方でバイトコードの分布に基づく類似度を用いた場合は保持するにも関わらず保持すると推定しない割合が LCS に基づく類似度を用いた場合に比べて低く，優位性があるといえる．

6　まとめと考察

本稿では短時間でマルウェアの概要を取得する目的で類似度を用いた機能推定法を提案した．機能推定は以下のように実行する．解析対象とするマルウェアに対して複数の解析済み検体との類似度を導出し，機能ごとにポイントを導出する．ポイントに対して判別分析法を用いる事によってしきい値を決定し，機能を 2 分化する．しきい値に対してそれより大きな値のポイントを持つ機能をマルウェアが保有する機能として推定する．機能推定に用いた類似度はバイトコードの分布，LCS，*n*-gram に基づくものである．機能推定実験では CCC DATAset2012 の検体を用い，機能の定義として NONSTOP の動的解析レポートから Action タグ内の機能を用いた．45 検体の機能推定の結果 *n*-gram に基づく類似度を用いた場合は LCS，バイトコードの分布に基づく類似度に比べて優位性が認められなかった．LCS，バイトコードの分布に基づく類似度は False Positive が少ない面では LCS に基づく類似度に優位性があり，False Negative が少ない面ではバイトコードの分布に基づく類似度に優位性がある事がわかった．一方で類似度に関する計算量ではバイトコードの分布に優位性がある．LCS を用いた類似度の導出には対象

表 4　Malwares used for Function Estimation

NameByTrend	Hash
Mal_Allaple	c64bb24*
Mal_Allaple	dc7de4f*
Mal_DownAd-2	2c59be0*
Mal_DownAd-2	4f3b3a2*
Mal_DownAd-2	70bb952*
Mal_DownAd-2	80ffb99*
Mal_DownAd-2	c3084d2*
PE_VIRUT.AT	66e99a0*
PE_VIRUT.AT	82c67f3*
PE_VIRUT.AV	14be48d*
PE_VIRUT.AV	3690efe*
PE_VIRUT.AV	387876f*
PE_VIRUT.AV	43790f0*
PE_VIRUT.AV	a5df56d*
TROJ_AGENT.DCG	8746eeb*
TROJ_DROPPER.CBV	e95f7bb*
WORM_ALLAPLE.AF	a6b66b1*
WORM_ALLAPLE.IK	0001316*
WORM_ALLAPLE.IK	00087db*
WORM_ALLAPLE.IK	3b66952*
WORM_ALLAPLE.IK	3b6a681*
WORM_ALLAPLE.IK	fff6e4e*
WORM_ALLAPLE.PF	0051578*
WORM_ALLAPLE.PF	0132d1d*
WORM_ALLAPLE.PF	034cb4d*
WORM_ALLAPLE.PF	0452956*
WORM_ALLAPLE.PF	29792d3*
WORM_DOWNAD.A	5815b13*
WORM_DOWNAD.A	ab163d3*
WORM_DOWNAD.A	ff1c8a1*
WORM_DOWNAD.AD	006be5c*
WORM_DOWNAD.AD	00d86cb*
WORM_DOWNAD.AD	016a8e6*
WORM_DOWNAD.AD	018ab30*
WORM_DOWNAD.AD	022f454*
WORM_DOWNAD.AZ	f68c70a*
WORM_IRCBOT.TFT	6acb0c0*
WORM_NEERIS.A	09851b2*
WORM_RAHACK.AA	beb7959*
WORM_SLENFBOT.MJ	be380f1*
WORM_SPYBOT.OQ	4ead474*
WORM_SPYBOT.OQ	9e36652*

とするデータのサイズを n とした時に $O(n^2)$ の計算量が必要であり，計算に時間かかるため大量の比較には不適切である．一方でバイトコードの分布に基づく類似度導出では固定で 256 の要素を比較することになるため計算量は O(1) であり，本研究の目的である短時間にマルウェアの概要を導出するためには今回の比較対象とした類似度のうちではバイトコードの分布に基づく類似度が最も優位性を持つと言える．今後の課題として更にデータベースを拡張することが必要であると考えられる．解析済みの検体のデータベースを拡張することにより，推定の精度

表 5　Example of Function Estimation

00000001	.3938930
00000002	.5199589
00000004	.0468069
00000005	.2014333
00000006	.4042853
00000007	.1202946
00000009	.2108010
00000015	.1248085
00000017	.3317668
00000018	.2199342
00000019	.3051737
00000020	.0740714
00200001	.2036509
00300001	.3884128
00600001	.3845414
00700001	.3979695
00800001	.3396033
01000001	.3906874
02000001	.0468069
02000002	.0104218
04000002	.0903658
06000001	.3441127
06000013	.2098088
20000002	.2726326
40000001	.0742988

表 6　Function Estimation Result

	$\frac{A \cap \overline{B}}{A}$	$\frac{A \cap B}{B}$
n-gram	0.3591	0.8137
LCS	0.2390	0.8823
Dist	0.2584	0.9215

が向上すると考えられる．また機能推定に用いる類似度に関してもその他複数の指標を用いる事により最適な類似度導出法を導くことが必要である．

参考文献

[1] 大久保諒，伊沢亮一，森井昌克，井上大介，中尾康二，“マルウェアの部分コードによる類似度判定および機能推定法，”CSS2012，1A1-2，2012

[2] 岩村誠，伊藤光恭，村岡洋一，“機械語命令列の類似性に基づく自動マルウェア分類システム，” 情報処理学会論文誌，vol.51，No.9，pp1-11，2010

[3] 東結香，中津留勇，真鍋敬士，猪俣敦夫，藤川和利，砂川秀樹，“コードに基づいたマルウェアの機能推定に関する研究，”SCIS2011，3B4-4，2011

[4] 新井悠，岩村誠，川古谷裕平，青木一史，星澤裕二，“アナライジングマルウェア，” オイラリージャパン

[5] Intel 64 and IA-32 Architectures Software Developer’s Manual，‘‘http://www.intel.com/’’

[6] Symantec Security Response，‘‘http://www.symantec.com/’’