マルウェア解析の効率化手法の検討

山口 和晃　堀合 啓一　田中 英彦

情報セキュリティ大学院大学

221-0835 神奈川県横浜市神奈川区鶴屋町 2-14-1

{mgs081101,dgs063101,tanaka}@iisec.ac.jp

あらまし 近年のマルウェアは、プロの手により複雑化しており、数そのものも爆発的に増えていることから、人手による解析のみでは解析者の負担が多く、解析が困難になっている。そのため、自動的なマルウェアの解析が必要であるが、最近は仮想マシンで実行されているか、ネットワーク接続の有無、日時などの動作環境要因から処理を分岐することで、振る舞いを変化させる耐解析機能を備えたマルウェアが増えている。このことから、自動的解析の精度の低下や解析効率の低下が起きている。そのため、本研究では、マルウェアの実行環境や実行方法の工夫により、自動的解析の精度の向上や解析効率の向上に対する有効性の検討を行う。

Efficiency Improvement of Malware Analysis

Kazuaki Yamaguchi　Keiichi Horiai　Hideki Tanaka

Institute of information security

2-14-1 Tsuruyacho Kanagawa-ku Yokohama-shi Kanagawa 221-0835 Japan

{mgs081101,dgs063101,tanaka}@iisec.ac.jp

Abstract In late years a malware which became complicated by malicious attackers is used. Because the number of malware increases for an explosion, a burden of work centers on a person of limited analysis, and analysis becomes difficult. Therefore it is necessary to analyze a malware automatically. However, an action of a malware changes by practice environment. So precision of automatic analysis and efficiency deteriorate. Therefore, by changing of practice environment, precision of automatic analysis and efficiency are studied.

1　はじめに

初期の攻撃者の動機は「自己表現」であり、主にファイル感染やブートセクタウイルス、マクロウイルスなどを用いての攻撃であった。しかし、近年はプロの手による複雑化したマルウェアが使われ、感染経路も複雑化するとともにソーシャルエンジニアリング手法も活用し、数そのものも倍増しているのが特徴である。また、解析者からのマルウェア解析を遅らせるために耐解析機能を備えたものがある。このことから、人手による解析のみでは解析者の負担が多く、解析が困難になっている。そのため、自動的なマルウェアの解析が必要であるが、従来のマルウェアの分析方法としての標準的なやり方は仮想マシン上でのみマルウェアの動的分析を行ってきた[9,12]。この方法は、レジストリやファイルへのアクセスを記録したり、ネットワークトラフィックをキャプチャしたりするツールが数多く存在することや、実行環境の復元の即時性から仮想マシンを使用することで比較的容易に環境を構築することができる。しかし、仮想マシンで実行されているか、ネットワーク接続の有無、日時などの動作環境要因から処理を分岐することでマルウェアの振る舞いが複雑化されているために分析が困難になりつつあり、解析精度の低下や解析効率の低下が起き

ている[1]。そのため、本研究では、CCC DATAset 2009[12]のマルウェア検体を用い、マルウェアの実行環境や実行方法の工夫により、自動的解析の精度の向上や解析効率の向上に対する有効性の検討を行う。

2　マルウェアの動的挙動自動解析システム

動的挙動解析の全体構成を図1[2]に示す。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。模擬サーバの一部は、Truman[13]の機能を利用した。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 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 の中に実装している。

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、マルウェアを実行する時期・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、文献[9,13]のように API CALL の情報を解析するのではなく、マルウェ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ェ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ェアへの対策としている。またマルウェアを実行する Windows OS の種類として Windows XP 各サービスパックの場合の比較、マルウェアを実行する感染 PC 環境を仮想マシンにするか、実マシンでするかによる比較ができる環境を構築した。

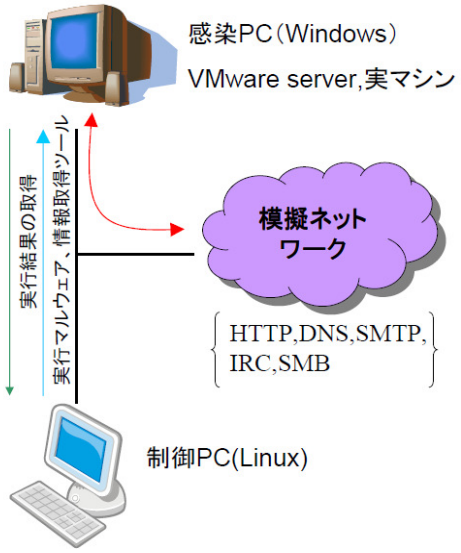

図1 動的挙動解析の全体構成[2]

2.1　マルウェアの実行制御

次にマルウェアの実行制御について述べる。マルウェアの挙動解析は、実行制御用のホスト（以下ホスト）OS と、マルウェアを実行する感染 PC（仮想マシンの VMware Server または、実マシン）上で作動しマルウェアを実行する。仮想マシンを利用する際は、物理的には1台の PC に実装している。感染 PC の OS が起動すると続いて感染 PC 内のローダが起動し、感染 PC 内の情報取得に必要なソフトウェアをホストから受信し、マルウェア実行前のログを取得する。続いて解析対象のマルウェアをホストから受信してこのマルウェアを実行し、指定した時間経過後にマルウェア実行後のログを取得する。 また、マルウェアの実行時間は200秒とした[6]。

2.2　システム内の情報取得と利用ツール

マルウェアが感染した際の情報を取得するには、このためのソフトウェア・ツールが必要である。また、一連の解析処理を制御する仕組みが必要となるが、表1に本システムで利用したツールを示す。これらのツールをあらかじめ解析対象のマルウェアを実行する感染PC へインストールしておくことも考えられるが、OS の種類やバージョンの違いなどが、感染 PC の挙動に及ぼす影響を調査する場合には、個々の感染PC へツールをインストールする必要がある。また、挙動解析の項目を追加する

場合には、これに必要なツールを個々の感染 PC へインストールすることが必要となり非効率である。このような作業の効率化を図るため感染 PC 内へホストからダウンロードして実行する方法を使用した。

表1 使用ツール一覧

使用目的	ツール名	配布元
Registry	autorunsc	Microsoft[23]
Process	pslist	Microsoft[23]
Service	PsService	Microsoft[23]
Listen ports	Fport	Fundstone[24]
RootKit detection	RootKitRevealer	Microsoft[23]

2.3 実マシン環境と仮想マシン環境の比較

マルウェアの挙動を自動的に解析するには、マルウェアの実行環境、自動解析処理の全体制御や挙動の記録と分析、感染後の PC の復旧などの機能が必要となる。実マシン環境と仮想マシン環境の違いを整理した結果を、図2に示す。同図に示すように、ほとんどの処理は共通化が可能であり、違いのある部位は、感染後に復旧する機能を実現する部分だけとなっている。

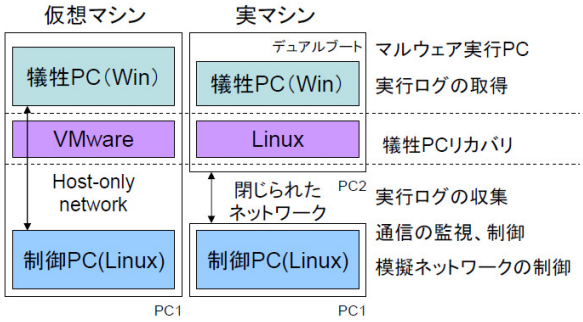

図2 実マシン環境と仮想マシン環境の違い[2]

3 マルウェアの実行で取得する情報

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、マルウェアの特徴（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。この動的挙動解析の結果から、レジストリ改竄、システムファイルの変更、プロセスリスト、Hostファイルの改竄、ルートキットの有無、通信ログ、などの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 。

4 自動解析の結果

自動解析から収集できた BDB の一例を表2にまとめる。また、この結果は仮想マシン環境の Windows XP Professional SP0 において実行した結果である。

表2 BDBの一例

[HASH] マルウェアのハッシュ値の先頭6桁
f8c19c
[REGISTRY] レジストリ改ざん
c:¥windows¥u29tzwjvzhk¥command.exe c:¥program files¥network monitor¥netmon.exe
[MD5SUM] システムファイル等の変更
Created C:¥WINDOWS¥SYSTEM32¥ATMTD.DLL.TMP
[PROCESS] Listen port プロセスリスト
628 winlogon 672 services 588 netmon 128 command 572 wuauclt
[HOSTS] HOSTS ファイル改ざん
Change Found.
[ROOT KIT] ルートキットの有無
Not Detected
[TRAFFIC] 通信ログ
FQDN(13) xxx6ry3i3x3qbrkwhxhw.xxx439.com xxxmand.xxxervs.com PORT(2),:53(4) 80(20) 35815(3) 42613(3) 44627(3)
43327(3) 45390(3) 49830(3) 58010(3) 44645(3) 48717(3) 56840(3) 54267(3)

4.1 10種類の検体について特徴のまとめ

自動解析から収集できた BDB の結果をマルウェア別にそれぞれ表3～表7にまとめた。なお、使用した実行環境は、Windows XP Professional SP0（仮想マシン）である。表3から表7のフラグは過去に得られた解析データからランキングしたトップ5のファイルが変更された場合にフラグが立つところと、その他の変更があった場合にフラグが立つところに分かれている。また、表中の x フラグはデータの取得ができなかったことを示す。

表3 通信ポートログ

hash 値	通信ポート(Top 5+その他+ICMP)						
1d23f2	1	0	0	0	0	1	0
393f00	1	0	0	0	0	1	0
68ac29	0	0	0	0	0	1	0
7190e4	0	0	0	0	0	1	0
84e9c2	1	0	0	0	0	1	0
cd9125	0	0	0	0	0	1	0
d49391	0	0	1	0	0	1	0
df7585	1	0	0	0	0	1	0
f8c19c	1	0	0	0	0	1	0
fdf3bbc	1	0	0	0	0	1	0

表4 ファイル改ざん状況

hash 値	ファイル改ざん(Top 5+他+改竄+削除)							
1d23f2	0	0	0	0	0	0	0	0
393f00	0	0	0	0	0	0	0	0
68ac29	0	0	0	0	0	0	0	0
7190e4	0	0	0	0	0	0	0	0
84e9c2	0	0	0	0	0	0	0	0
cd9125	0	0	0	0	0	0	0	0
d49391	0	0	0	0	0	0	0	1
df7585	0	0	0	0	0	0	0	0
f8c19c	0	0	0	0	0	0	0	0
fdf3bbc	0	0	0	0	0	0	0	0

表5 プロセスの変更状況

hash 値	プロセス(Top 5+その他＋削除)						
1d23f2	0	0	0	0	0	0	0
393f00	0	0	0	0	0	0	0
68ac29	0	0	0	0	0	1	0
7190e4	0	1	0	1	0	0	0
84e9c2	0	0	0	0	0	0	0
cd9125	0	1	0	1	1	1	0
d49391	x	x	x	x	x	x	x
df7585	0	0	0	0	0	0	0
f8c19c	0	0	0	1	0	0	1
fdf3bbc	0	0	0	0	0	0	0

表6 レジストリの変更状況

hash 値	Registry													
1d23f2	0	0	0	0	0	0	1	0	0	0	0	0	0	1
393f00	0	0	0	0	0	0	1	0	0	0	0	0	0	1
68ac29	0	0	0	0	0	0	1	0	0	0	0	0	0	1
7190e4	0	0	0	0	0	0	1	0	0	0	0	0	0	1
84e9c2	0	0	0	0	0	0	1	0	0	0	0	0	0	1
cd9125	0	0	0	0	0	0	1	0	0	0	0	0	0	1
d49391	x	x	x	x	x	x	x	x	x	x	x	x	x	x
df7585	0	0	0	0	0	0	1	0	0	0	0	0	0	1
f8c19c	0	0	0	0	0	0	1	0	0	0	0	0	0	1
fdf3bbc	0	0	0	0	0	0	1	0	0	0	0	0	0	1

表7ホストファイル、サービス、ルートキット有無

hash 値	Service	RootKit	host
1d23f2	0	0	0
393f00	0	0	0
68ac29	0	0	0
7190e4	0	0	0
84e9c2	0	0	0
cd9125	0	0	0
d49391	x	x	x
df7585	0	0	0
f8c19c	0	0	0
fdf3bbc	0	0	0

4.2 **実行環境の変化による挙動の変化**

実行環境においてのマルウェアの挙動の違いを比較するため、Windows XP Professional SP0（仮想マシン）、SP2 の仮想マシンと実マシンの三つの実行環境を用意し、先頭ハッシュ値6桁が68ac29のマルウェアを解析したBDBの結果を表8～11にまとめた。

表8 環境による通信ログの違い

実行環境 OS	通信ポート(Top 5+その他+ICMP)						
XPSP0（仮想）	0	0	0	0	0	1	0
XPSP2（仮想）	0	0	0	0	0	1	0
XPSP2(実機)	0	0	0	0	0	1	0

表9 環境によるファイル改ざんの違い

実行環境 OS	ファイル改ざん(Top 5+他+改竄+削除)							
XPSP0（仮想）	0	0	0	0	0	0	0	0
XPSP2（仮想）	0	0	0	0	0	0	0	0
XPSP2(実機)	0	0	0	0	0	0	0	0

表10 環境によるプロセスの違い

実行環境 OS	プロセス(Top 5+その他＋削除)						
XPSP0（仮想）	0	0	0	0	0	1	0
XPSP2（仮想）	0	1	0	0	0	0	0
XPSP2(実機)	0	0	1	0	0	0	0

表11ホストファイル、サービス、ルートキット有無

実行環境 OS	Service	RootKit	host
XPSP0（仮想）	0	0	0
XPSP2（仮想）	0	0	0
XPSP2(実機)	0	1	0

4.3 確率的な挙動の変化

動作環境に依存せずに実行時刻などから確率的に挙動パターンを変更するマルウェアについて考察する。 マルウェアの挙動を解析することにおいて、1種類の実行パターンしか持たないマルウェアに対して何度も実行していては無駄が多い。だが、複数の実行パターンを持つマルウェアにおいてはできるだけ全てのパターンを網羅する必要がある。ここで、考えられることは、ある回数において、あるパターン数が確認されていたとして、その次の解析においてまた新たな挙動パターンを得られる確率を考え、次の解析の確率から解析を終了するかどうかを判断すればよいと考えられる。これを一般化すると図3となる。

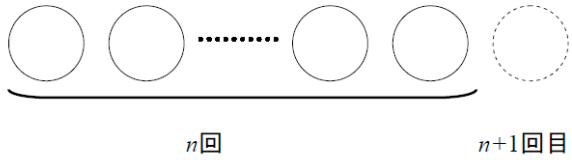

図3 試行回数における一般化

ここで真のパターン数は X パターンであったとして n 回の試行で x パターンのみ出現する確率 $f(x,n)$を考える。したがって、これは仮に X パターンあった場合に x パターンしかえられない確率であることから、n 回目の試行時に x パターン出現したときの取りこぼし確率となる。したがって、確率 $f(x,n)$が十分低ければ x パターンしか存在しないことが推定される。この確率は式(1)の大数の法則から導き出せる。式(1)から $f(x,n)$は式(2)となることがわかるので、その結果を表12にまとめた。

$$\Pr\left(\left|\bar{x}-\mu\right|\geq\varepsilon\right)\leq\frac{\sigma^2}{n\varepsilon^2} \qquad (1)$$

$$f(x,n)=\frac{\sigma^2}{n\,\mu^2} \qquad (2)$$

ここで、注意が必要なことは、この式が成り立つには確率が独立であり、一様分布している必要がある。表12の結果から、最低10回以上の解析を行わなければ、90%以上の確率で全ての挙動パターンを網羅したとは言えないことがわかる。

表12 取りこぼし確率(単位[%])

x パターン	試行回数[回]				
	1	10	20	50	100
1	100	10	5	2	1
2		20	10	4	2
3		30	15	6	3
4		40	20	8	4
5		50	25	10	5

したがって、Windows XP Professional SP0(仮想マシン)において各10回の解析を行い、えられた挙動パターン数と各挙動パターンにおいてえられた回数を表13にまとめた。

表13 解析の10回試行結果

hash 値	挙動パターン数	各出現回数
1d23f2	1	10
393f00	1	10
68ac29	2	5,5
7190e4	1	10
84e9c2	1	10
cd9125	2	9,1(1エラー含)
d49391	1	10
df7585	2	9,1(1エラー含)
f8c19c	1	10
fdf3bbc	1	10

10回行われた結果から挙動パターン数の確率的予測を行った場合の妥当性を検証するために100回試行した結果を表14にまとめた。

表14 解析の100回試行結果

hash 値	挙動パターン数	各出現回数
1d23f2	1	100
393f00	1	100
68ac29	4	69,23,6,2
7190e4	1	100
84e9c2	1	100
cd9125	2	99,1(1エラー含)
d49391	1	100
df7585	2	99,1(1エラー含)
f8c19c	1	100
fdf3bbc	1	100

5 考察

表8～11から実行環境によってマルウェアに挙動の変化があることがわかる。実マシンの挙動について注目すると、表11において、実マシンのみルートキットの実行が RootkitRevealer に検出されたことがわかる。これは、仮想マシンでの解析時にマルウェアが仮想マシンで実行されていること

から、解析されていることを検知し、挙動を変化させ、そのまま停止したか、もしくは偽のまったく違う実行動作を行い、真の実行動作をやめてしまった可能性が高い。このことから、実マシンでの解析も併用して使用する必要がることがわかる。また、仮想マシンの SP0, SP2 同士の比較においても挙動の変化があることがわかる。これは、SP2 のセキュリティ機能の強化などが関係すると考えられる。また、セキュリティが脆弱である SP0 で実行されていなかったにもかかわらず、セキュリティが強化された SP2 の実マシンで実行した場合はルートキットが RootkitRevealer に検出されたことがわかる。このことからも、実マシンでの解析の重要性がわかる。

次に、確率的な挙動についてであるが、式(1)が成立するのは確率が独立であり、一様分布している必要がある。マルウェアの挙動が実際に一様分布しているのか、表13、14から検討すると、複数の挙動が確認された“68ac29d”の出現回数に偏りがあることから、今回は一様分布であるとは言えない。だが、10回実行した状態で1パターンのみ挙動が得られたものはエラーを抜かせば100回実行を行っても1パターンのみの挙動であることから、10回の実行をして、複数のパターンが得られなければ、解析を停止することで、一定の効率化が可能と考えられる。

6 まとめと今後の課題

今回、複数の実行環境と実行方法によってマルウェアが挙動を変化させることがあることがわかった。また、マルウェアを10回実行した結果の挙動のバラツキ具合から、さらに実行を繰り返す必要があるかどうかについての推定の可能性があることが判明した。今後の課題としては、マルウェアの確率的挙動のメカニズムの調査から更なる効率化の検討、仮想マシンと実マシンとの併用にあたり、仮想マシンの解析結果から一回あたりの解析に時間がかかる実マシンの解析の回数を低減するなどの効率化の検討をしていく予定である。また、解析精度の向上においては、今回はAPIコールの監視を敢えて適用せずに行ったが、API コールの監視による挙動の変化を解析することで実行環境の最適化をするなどの研究を行っていく予定である。

参考文献

[1] 畑田充弘、他、”マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有”、MWS2009、2009/10

[2] 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 1-12、2008/4

[3] 永田 大、堀合 啓一、田中 英彦、“OLAP(多次元データ分析)を利用した攻撃元データの分析と検体の自動解析”MWS2008、2008/10

[4] 堀合 啓一、今泉 隆文、田中 英彦、“ハミング距離によるマルウェア亜種の自動分類”、第41回 CSEC、2008/5

[5] 星澤 裕二、”解析してわかる最近のマルウェアのテクニック” ISS スクウェア インターネット分科会、株式会社セキュアブレイン、2008/11/15

[6] 笠間 貴弘、吉岡 克成、井上 大介、衛藤将史、中尾 康二、松本 勉、“マルウェア動的解析におけるマルウェア実行時間に関する検討”、SCIS 2009、2009/1

[7] 市川 幸宏、神薗 雅紀、白石 善明、森井 昌克、“ウイルス解析を目的としたメモリ上の不正コード検出システムの構築”、電子情報通信学会技術研究報告、2004

[8] 星澤 裕二、岡田 晃市郎、山村 元昭、椎木 孝斉、マルウェアの動作条件の抽出、情報処理学会研究報告、2007

[9] 星澤裕二、太刀川剛、山村元昭、“マルウェアの亜種等の自動化”、情報処理学会研究報告 2007-CSEC-38 、pp271-278、2007

[10] 吉岡 克成、松本 勉、“自動マルチパス解析によるマルウェア動的解析の提案”、SCIS2009、2009

[11] IPA、”未知ウイルス検出技術に関する調査”、2004/4

[12] Tony Lee 、Jigar J.Mody 、“ Behavioral Classification”、In Proceedings of EICAR 2006、2006/4

[13] TJoe Stewar 、Truman 、“ The Reusable Unkonwn Malware Analysis Net”