Computer Security Symposium 2017
23 - 25 October 2017

実機を用いたマルウェア動的解析システムの環境復元ソフトウェアによる容易な構築方法

上野 航[†1]　石井 攻[†1]　田辺 瑠偉[†1]
吉岡 克成[†2]　松本 勉[†2]

概要：マルウェアの動的解析環境は仮想マシンやエミュレータを用いるものが多く，これらの環境を検知し動的解析を回避するマルウェアが報告されている．対策技術として実機を用いる手法が提案されているが，OS より低いレイヤで解析環境を管理する必要があるため技術的に高度であり，実装コストが高い．そこで本研究では，実機を用いたマルウェア動的解析システムを環境復元ソフトウェアにより容易に構築する方法を提案する．評価実験では，仮想環境を検知して挙動を変えることが確認されている 18 検体の実マルウェアを用いて提案手法の有効性を検証する．また，提案手法により構築した解析環境を検知するマルウェアについて考察する．

キーワード：環境復元ソフトウェア，マルウェア動的解析，解析環境検知， 実機による動的解析

Easy implementation of bare-metal sandbox with commercial system recovery software

Wataru Ueno[†1]　Kou Ishii[†1]　Rui Tanabe[†1]
Katsunari Yoshioka[†2]　Tsutomu Matsumoto[†2]

Abstract: For a long while, malware sandboxes have been mainly implemented with virtual machine or emulator for ease of system recovery and management. However, series of studies have revealed that more and more malware authors are trying to evade these virtual sandboxes. Although bare-metal systems have been proposed to overcome these evasion, their implementation is more complicated and costly than virtual ones as they require physical disk management below OS layer. In this paper, we propose an easy implementation of bare-metal malware sandbox using commercial system recovery software. At the experiment, we evaluate the proposed system with 18 malware samples known to evade virtual sandboxes. In addition, we discuss the possibility of evasion against our sandbox to prepare for future attacks.

Keywords: Recovery software, Malware sandbox analysis, Sandbox evasion, Bare-metal sandbox

1. はじめに

近年，悪意のあるソフトウェア，いわゆるマルウェアによる被害が増大している．これらのマルウェアを分析し対策を導出するため，解析対象のマルウェア検体を解析環境内で実行し，その通信挙動や内部挙動を観測するマルウェア動的解析が広く行われている．動的解析環境は解析の自動化や解析環境の復元が容易であることから仮想マシンやエミュレータを用いて実現される場合が多い．しかし，仮想環境を検知し本来の挙動を隠すことで動的解析を回避しようとするマルウェアが多く報告されている[1, 2].

このようなマルウェアの対策として，仮想マシンではなく実機を用いて動的解析システムを構築する手法が提案されている[3]．しかしながら，実機を用いた動的解析では，マルウェアを実行する OS よりも低いレイヤでシステム管理を行う必要があり，環境復元時の物理メモリの正確な読み書きが求められるなど実装コストが高い．そこで本研究では，実機を用いたマルウェア動的解析システムを，環境復元ソフトウェアを用いて容易に構築する方法を提案する．環境復元ソフトウェアとはシステムを再起動した際に，あらかじめ設定した環境に復元するためのソフトウェアである．本研究では，これらのソフトウェアを用いて解析環境を構築することで，実機を用いたマルウェア動的解析システムを容易に実装できることを示す．

評価実験では，仮想環境を検知して挙動を変えるマルウェアに対して提案手法が有効に働くことを検証するため，標的型攻撃などに用いられる RAT（Remote Administration Tool, Remote Access Trojan）検体を用いて提案手法の有効性を評価した．具体的には，まず RAT サーバのビルド機能をもつ 5 種類の RAT クライアントを用いて，仮想環境を検知して挙動を変える RAT サーバ検体を作成した．次に，提案手法によって実装された，実機による動的解析システム 2 種類と仮想マシンによる動的解析システム 3 種類，合計 5 種類の動的解析システムにより，上記の 5 つの RAT サーバ検体を実行し，その挙動を観測した．その結果，提案シス

†1 横浜国立大学
Yokohama National University
†2 横浜国立大学大学院環境情報研究院/先端科学高等研究院
Graduate School of Environment and Information Sciences, Yokohama National University / Institute of Advanced Sciences, Yokohama National University

テムでは全ての検体が RAT クライアントに通信を開始するのに対して仮想マシンを用いたシステムでは通信が観測されない場合が多数確認された．また，同様の実験を，仮想環境を検知して挙動を変えることが確認されている実マルウェア 18 検体について行った結果，13 検体について仮想マシンを用いたシステムでは観測できない，外部ホストへの通信を確認することができた．提案手法は，仮想環境を検知するマルウェアの挙動を観測するのに有効であり，実機を用いた解析環境を低コストで実現できる点で有用であるといえる．

以降では，2 章で関連研究について説明し，3 章で提案手法である環境復元ソフトウェアを用いた動的解析システムの構成方法について説明する．そして，4 章で仮想環境を検知して挙動を変えるマルウェア検体を用いた評価実験により提案手法の有効性を検証し，5 章でまとめと今後の課題を説明する．

2. 関連研究

マルウェアによる被害の増大に伴い，マルウェア検体を解析環境内で実行し，その挙動を観測するマルウェア動的解析が広く行われるようになった．動的解析システムの実現方法は多岐に渡るが，仮想マシン[4, 5]やエミュレータ[6, 7]を用いて実現される場合が多い．しかし，これらの環境を検知して動的解析を回避するマルウェアが報告されている[1,2]．たとえば，仮想化ソフトウェアの一つである VMWare[4]を用いて作成される解析環境は，管理用バックドア I/O ポートが作成されるため，当該ポートの有無を調べることで VMWare 検知を行うマルウェアが報告されている[8].

このような動的解析環境を検知して本来の挙動を隠すマルウェアに対して，実マシンと区別がつきにくい動的解析環境を実現する研究が行われている．論文[9]では，マルウェア検体のコードやメモリのレイアウトなどから，マルウェア検体が解析環境を検知するのに使われる情報を実マシンのものに置き換える手法が提案されている．また，論文[10]では，ハードウェアに Intel VT などの仮想化支援技術を用いることで実マシンとの区別をつきにくくする手法が提案されている．これらのシステムでは，動的解析システムの OS を実マシンと区別がつかないようにしているが，タイミングベースによる検知への対策が不十分であるため，CPU サイクルの差異に着目した検知手法が提案されている [11]．一方で，論文[3]ではマルウェア検体を実ハードウェア上で実行することで仮想環境の検知を困難にする手法が提案されている．しかし，マルウェア検体を実行する OS の外部から解析環境の復元を行わなければいけないことや，物理メモリへの読み書き動作を正確にコントロールする必要があるため，実装コストが高い．

そこで本研究では，環境復元ソフトウェアを用いて解析環境を構築することで，容易に実機を用いたマルウェア動的解析システムを実現する手法を提案する．

3. 提案手法

本章では，環境復元ソフトウェアを用いた動的解析システムの構築手法を提案する．まず 3.1 節で提案手法の基本アイディアを説明し，3.2 節で提案手法の実現形態である環境復元ソフトウェアを用いた動的解析システムについて説明する．そして，3.3 節でシステムの実装について説明する．

3.1 基本アイディア

提案手法は，仮想マシンやエミュレータを検知して挙動を変えるマルウェアの対策として，実機を用いた動的解析環境を構築するものである．動的解析では，実際にマルウェア検体を解析環境内で動作させるため，マルウェア検体による攻撃が動的解析システムの外部に出ないようにする必要がある．また，マルウェア検体実行後に新たな検体を実行するために，解析環境をマルウェア感染前の状態に復元する必要がある．前者はパケットフィルタリングなどを用いることで対策を行うが，後者のシステムの復元には実機に直接インストールした OS のクリーンアップが必要となるため手間がかかる．そこで，提案手法では環境復元ソフトウェアと呼ばれるソフトウェアを用いることで，動的解析環境の復元を容易に行う．

環境復元ソフトウェアとは，システムを再起動・シャットダウンした際に，あらかじめ設定した環境に復元するためのソフトウェアである．日本国内でも多数の環境復元ソフトウェアが販売されており，学校やネットカフェといったような不特定多数のユーザが利用する PC 環境に導入されている場合が多い．環境復元ソフトウェアの具体的な実装方法は製品によって異なるが，典型的な復元の仕組みを図 1 に示す．環境復元ソフトウェアは，初めにハードウェア上に新しくパーティションを作成して保存領域とし，ユーザや OS がファイルを変更した際に，ファイル情報，環境設定変更情報等すべての操作情報をこの保存領域に書き込んでいく．当該操作により，保護対象のパーティションを操作しているように見えるが，実際には保護パーティションには一切書き込みを行わず，その代わりに保存領域へ書き込みを行う．そして，システムを再起動・シャットダウンした際に保存領域の変更方法を削除し，普段通りシステムを起動することによって環境の復元を行う．本研究では，このソフトウェアを用いて動的解析環境の復元を行うことで，実機を用いた動的解析システムを実現する．

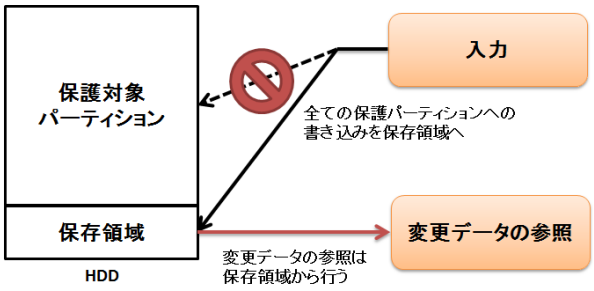

図 1　環境復元ソフトウェアの復元方法の例
Figure 1　Example of the restoration method of a system recovery software.

3.2 環境復元ソフトウェアを用いた動的解析システム

本節では，3.1 節で説明した基本アイディアの実現形態である環境復元ソフトウェアを用いた動的解析システムについて説明する．図 2 に本システムの構成を示す．本システムは，マルウェア検体を実際に実行する**犠牲マシン**と，犠牲マシンのデフォルトゲートウェイであり，外部通信の観測とパケットフィルタリング，犠牲マシン上で実行する検体の管理などを行う**管理用マシン**の 2 つのマシンから構成される．犠牲マシンには**制御部**が存在し，犠牲マシンの制御を行う．また，管理用マシンには**アクセスコントローラ**と**解析マネージャ**が存在する．以降では，本システムの構成要素について説明する．

犠牲マシン：マルウェア検体を実行するマシンであり，事前に環境復元ソフトウェアがインストールされ，再起動を行うことによってあらかじめ設定した環境に復元を行うように設定する．制御部が犠牲マシンを再起動した際に，環境復元ソフトウェアにより設定された環境に復元される．

制御部：犠牲マシンの制御を行う．犠牲マシンの OS が起動した際に自動的に動作し，管理用マシンからマルウェア検体や設定情報のダウンロード，マルウェア検体の実行，内部挙動の観測，犠牲マシンのシャットダウンを行う．そして，解析時間が経過すると観測結果を管理用マシンに送信し，犠牲マシンの再起動を行うことで環境を復元する．

管理用マシン：解析の管理を行うマシンであり，アクセスコントローラと解析マネージャから構成される．管理用マシンを犠牲マシンのデフォルトゲートウェイとして設定することで，犠牲マシンから発生した通信を観測する．

アクセスコントローラ：アクセスコントローラは，犠牲マシンで発生した通信が動的解析システムの外部に悪影響を与えないようにアクセス制御を行う．マルウェアが行う通信のうち，危険性が十分に低いと判断された通信のみ実インターネットへ転送し，攻撃と思われる通信については遮断する．犠牲マシンから送られてきたパケットの送信元 IP アドレスを管理用マシンの IP アドレスに変換し，インターネット上のホストと通信が確立された場合には，インターネット側から送られてきたパケットの送信先 IP アドレスを犠牲ホストの IP アドレスに変換することで通信を実現する．

解析マネージャ：解析マネージャは，動的解析システムの中核として犠牲マシン上で実行するマルウェア検体の管理，犠牲マシンへのマルウェア検体・解析時間の送信，犠牲マシンから発生した外部通信の観測，アクセスコントローラの設定，解析結果の出力を行う．以下に解析の流れを示す．

解析の流れ：

① 解析者は，管理用マシン内に存在する設定ファイルに解析対象のマルウェア検体，解析時間，解析回数の情報を入力した後，犠牲マシンの電源を起動してマルウェア動的解析を開始する．

② 犠牲マシンが起動されると制御部が動作し，管理用マシン上の解析マネージャと通信を行う．管理用マシン上の解析マネージャは，解析者が入力した設定情報に従いマルウェア検体が保存されているディレクトリ，解析時間を犠牲マシンに送信する．また，アクセスコントローラにフィルタリングルールを適用する．その後，犠牲マシンの通信のキャプチャを開始する．

③ 犠牲マシンの制御部は，管理用マシンからマルウェア検体をダウンロードし，これを実行する．犠牲マシンで実行されたマルウェアの通信は全て管理用マシンを経由するため，マルウェアの通信は管理用マシンにて観測される．

④ 犠牲マシンの制御部は，設定されたマルウェア実行時間が経過すると，犠牲ホストを再起動する．また，管理用マシン上の解析マネージャも，設定されたマルウェア実行時間が経過すると犠牲マシンの通信キャプチャを止め，収集した通信挙動ログを出力する．その後，全てのマルウェア検体の解析が完了した場合，⑤に進む．一方，解析する検体が残っていた場合には，解析が終了するまで②～④を繰り返す．

⑤ 解析マネージャは，解析結果を出力し，犠牲マシンの制御部を用いて犠牲マシンの電源をシャットダウンして解析を終了する．

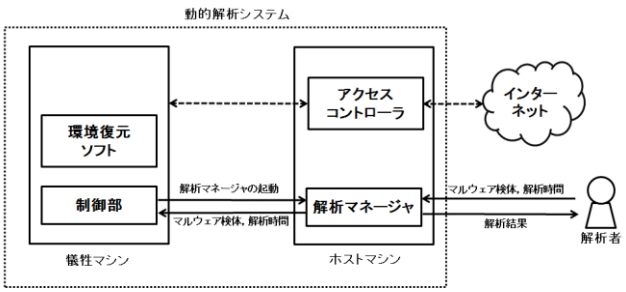

図 2　提案システムの構成
Figure 2　Structure of proposed system.

3.3 環境復元ソフトウェアを用いた動的解析システムの実装

本節では，3.2 節で説明した提案システムの実装について説明する．図 3 に本システムの全体像を示す．当該システムは犠牲マシン・管理用マシンの 2 台のマシンを用いて実装した. 以降では, 各構成要素の実装について説明する．

犠牲マシン：犠牲マシンは iiyama 社製 STYLE-15FX093-i7-RNFR 型番 IStNno-15FX093-i7_-RNFRB (CPU : インテル Core i7-7700HQ プロセッサー，メモリ：DDR4-2400 S.O.DIMM (PC4-19200) 16GB，ストレージ : Serial-ATA SSD 480GB)，OS には Windows 7 Professional SP1 を用いた．犠牲マシンのデフォルトゲートウェイをホストマシンの IP アドレスに設定した．後述の制御部のインストール後に環境復元ソフトウェアをインストールし, 再起動した場合に，制御部を含む OS イメージが復元されるように設定を行った．環境復元ソフトウェアは 2 種類用意した．使用した環境復元ソフトウェアについては 4 章にて述べる．なお，今回の実装では，犠牲マシンへの内部挙動観測機構の導入は行わず OS インストール直後のイメージを用いたが, OS イメージの差し替えや内部挙動観測機構の導入は，初期設定時に実施することで容易に可能である

制御部：制御部は，C#によって作成された EXE ファイルにより実装した. この EXE ファイルのショートカットを犠牲マシンのスタートアップフォルダに入れておくことで，犠牲マシンが起動すると制御部が自動的に動作するようにした．制御部が動作すると管理用マシンに SSH 接続し，管理用マシン上の解析マネージャと通信を行う．まず解析マネージャからマルウェア検体のパスと解析時間情報が送られると，制御部は送られてきたパス情報を用いて管理用マシンからマルウェア検体をダウンロードし，実行する．マルウェア検体実行後，設定された解析時間が経過するまで制御部は待機する．そして，設定された解析時間が経過したのち，犠牲マシンを再起動することによって解析環境の復元を行う．解析マネージャからマルウェア検体のパスと解析時間情報が送られない場合，解析終了もしくは何らかのエラーが発生したと判断し，犠牲マシンをシャットダウンする．

管理用マシン: 管理用マシンの OS には Ubuntu 14.04.4 LTS を用いた．管理用マシンは 2 つのネットワークインターフェイスを用意し，一方は犠牲ホストとの通信に，もう一方はインターネット接続することで犠牲ホストのゲートウェイとして働くようにした．

アクセスコントローラ：アクセスコントローラは，管理用マシンの OS である Ubuntu 14.04.4 LTS に標準でインストールされているパケットフィルタリングツール iptables により実装した．任意のマルウェア検体に対して安全に解析ができるフィルタリングルールを定義するのは困難であるが，今回の実験では，23/tcp，135/tcp，139/tcp，445/tcp，3389/tcp

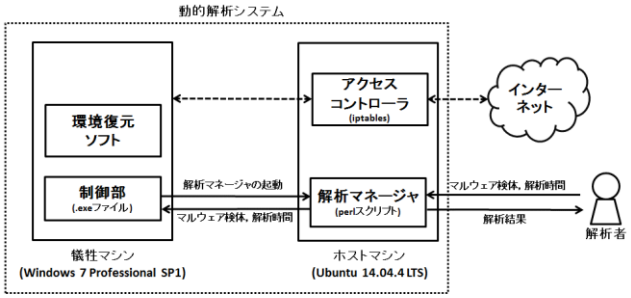

図 3　環境復元ソフトウェアを用いた動的解析システムの実装

Figure 3　Implementation of Dynamic analysis system with system recovery software.

といった感染拡大活動の対象となるポートへの通信については iptables の DROP 機能を用いてデフォルトで通信を遮断する設定とした．一方，それ以外の通信については，iptables の POSTROUTING チェインにより，送信元 IP アドレスを管理用マシンの IP アドレスに変換し, インターネット上に転送した．このようなフィルタリングルールは完全ではないが, 実験で用いる検体は RAT クライアントにより自作した RAT サーバ検体，または，主に標的型攻撃や情報収集に用いられる検体であることが分かっているため，外部へ攻撃が流出する可能性は低いといえる．実際，実験中に観測された外部ホストへの通信はいずれも C&C 通信と思われるものであった．また，インターネット側からのパケットが正しく犠牲マシンに到達するよう，iptables の IP マスカレード機能により NAT 変換を行うようにした．

解析マネージャ：解析マネージャは Perl スクリプトとして実装した．解析マネージャの機能は 3.2 節で説明した通りである．犠牲マシンからの通信の観測には，パケットモニタリングツールである tcpdump を用いた．

4. 評価実験

提案手法は，実機を用いたマルウェア動的解析環境により，仮想解析環境を検知して本来の挙動を隠すマルウェアに対して有効であることが期待される．そこで，標的型攻撃などに用いられる RAT（Remote Administration Tool, Remote Access Trojan）のクライアントツールを用いて，仮想マシン検知機能を有する RAT サーバ検体を作成し, 提案手法の有効性を評価する．また，仮想環境を検知して挙動を変えることが確認されている実マルウェア 18 検体を用いて提案手法の有効性を評価する．以降では，4.1 節でそれぞれの実験内容を説明し, 4.2 節で実験結果を説明する．そして，4.3 節で考察を行う．

表 1　実験 1 に用いた RAT クライアント一覧

Table 1　List of RAT client used in experiment 1

名称	機能
Cerberus RAT 1.03.4	Anti VMWare, Anti VirtualBox
Cybergate RAT 1.07.5	Anti VMWare, Anti VirtualBox, Other
DH Rat 2.0	Anti VMWare, Anti VirtualBox
DarkTrack Alien 4.1	Anti VMWare, Anti VirtualBox
LeGeNd - of -SiR -Do0oM -RAT	Virtual System Bypass

4.1 実験概要

実験 1：自作した RAT 検体を用いた提案手法の評価

RAT は，対象ホストをリアルタイムに遠隔操作するツールであり，操作対象の侵入先ホスト上で動作する RAT サーバと，それを遠隔操作するための RAT クライアントからなる．RAT クライアントは RAT サーバのビルド機能を有しており，ビルド時に RAT サーバに仮想環境検知機能をもたせるか設定が可能なものがある．そこで実験 1 では，5 種類の RAT クライント(表 1)を用いて，仮想環境を検知する RAT サーバ（以降では，単に RAT 検体とよぶ）をそれぞれ作成し，実験を行う．RAT 検体は仮想環境を検知すると RAT クライアントへの通信を行わないため，RAT クライアントへの通信の有無を調べることで解析環境が検知されたかを確認することが可能である．

実験には，環境復元ソフトウェア”Recovery Flash[12]”と”HD 革命/WinProtector Ver.6[13]”をそれぞれ用いて実装した 2 種類の提案システムと，仮想化ソフトウェア”VMWare[4]”, “Virtual Box[5]”, “QEMU[7]”をそれぞれ用いて実装した 3 種類の動的解析システム，合計 5 種類の動的解析システムを用いた．RAT 検体の実行時間は 3 分に設定し，全ての RAT 検体を 1 回ずつ実行した．

実験 2：実マルウェア検体を用いた提案手法の評価

実験 2 では，仮想環境を検知して挙動を変えることが確認されている 18 検体の実マルウェアを実験 1 で用いた 5 種類の動的解析システム上で実行し，通信挙動を観測した．マルウェア検体の実行時間は 3 分に設定し，各マルウェア検体が名前解決を行ったドメイン名の比較を行った．ただし，マルウェアの挙動は不確定であるため，各検体を 3 回ずつ実行し，それぞれの実行時の通信を観測した．実マルウェア検体の検知名を表 2 に示す．

表 2　実マルウェア検体の検知名

Table 2　Details of malware samples used for experiment 2.

検体No	Number	Symantec	TrendMicro
1	3	Trojan.Gen	TROJ_FORUCON.BMC
2	3a	Trojan.Gen.2	BKDR_ANDROM.RN
3	3b	Trojan.Gen	TROJ_FORUCON.BMC
4	3c	Trojan.Zbot	TROJ_WAUCHOS.VTI
5	3d	Backdoor.Trojan	BKDR_ANDROM.LMP
6	3e	Trojan.Klovbot	TROJ_INJECT.RZA
7	3f	Trojan.Zbot	BKDR_ANDROM.ABE
8	4	Trojan.Zbot	TROJ_FRS.BMA000LI13
9	4a	Trojan.Zbot	TSPY_ZBOT.ZZC
10	4b	Trojan.Zbot	TSPY_ZBOT.SMIG
11	4e	Downloader.Upatre!gen5	TSPY_ZBOT.YUNJH
12	4f	WS.Reputation.1	TROJ_GEN.R0CB0DCK14
13	9	ML.Relationship.HighConfidence[Downloader.Ponik]	BKDR.EMDIVI.SM
14	10	Infostealer.Dyre	TSPY_DYRE.SMNC
15	10a	Infostealer.Dyre	TSPY_DYRE.AATZ
16	11	Suspicious.Cloud.9	TROJ_BLOCKER.GA
17	13	Trojan Horse	TROJ_ROVNIX.E
18	15	Trojan Horse	BKDR_OBFUSC.SDF

表 3　RAT 検体を用いた評価実験結果（実験 1）

Table 3　Result of evaluation experiment using RAT samples (experiment 1)

名称	提案手法1 (Recovery Flash)	提案手法2 (HD革命)	仮想マシン1 (VMWare)	仮想マシン2 (Virtual Box)	仮想マシン3 (QEMU)
Cerberus RAT 1.03.4	○	○	×	×	×
Cybergate RAT 1.07.5	○	○	×	×	○
DH Rat 2.0	○	○	×	×	○
DarkTrack Alien 4.1	○	○	×	×	×
LeGeNd - of -SiR -Do0oM -RAT	○	○	×	×	○

4.2 実験結果

実験 1 の結果

仮想環境を検知して挙動を変える 5 種類の RAT 検体を 5 台の動的解析システムで実行した結果を表 3 に示す．表 3 では，RAT クライアントに対して通信を試みたものを”○”，通信を行わなかったものを”×”で示している．表 3 の通り，提案システムでは全ての RAT 検体が RAT クライアントに通信を試みた．一方，VMWare や Virtual Box を用いて構築した動的解析システムでは RAT クライアントへの通信は観測されなかった．QEMU を用いて構築した動的解析システムでは，3 検体から RAT クライアントへの通信が観測された．これは，実験に用いた RAT 検体の一部はエミュレータの検知を行っていないためと思われる．このように提案手法は仮想マシンを検知するマルウェアに対して有効であることが確認できた．

実験 2 の結果

実マルウェア検体を 5 種類の動的解析システムで実行した結果，“Recovery Flash[12]”を用いて構築した提案システムと“HD 革命/WinProtector Ver.6[13]”を用いて構築した提案システムでは，各マルウェア検体が名前解決したドメインは全て同じであった．提案システムと仮想マシンによる動的解析システムにおいて，検体が名前解決を行ったドメインの比較を表 4 に示す．表 4 では 3 回の検体実行において毎回観測されたドメインのみを記載している．各解析環境において名前解決が行われたドメインを”○”，名前解決を行わなかったものを”×”で示している．18 検体中 13 検体からは，提案システムでは観測可能であるものの 3 種類の仮想解析環境のいずれか，または全てで観測できないドメインが 19 個確認された．逆に，検体 14 と 15 については，仮想解析環境のみでドメイン名前解決が観測されているが，これらはグーグルの日本語トップページ www.google.co.jp であった．このように，提案手法を用いることで仮想解析環境に比べて C&C サーバである可能性が高いドメインが，より多く観測できることがわかった．

表 4　実マルウェア検体を用いた評価実験結果（実験 2）

Table 4　Result of evaluation experiment using real malware samples (experiment 2)

検体No	ドメイン名	提案手法1 (Recovery Flash)	提案手法2 (HD革命)	仮想マシン1 (VMWare)	仮想マシン2 (Virtual Box)	仮想マシン3 (QEMU)
1	*****res.net	○	○	○	×	×
	www.*****omains.com	○	○	○	×	×
	*****irect-LB3-890977680.us-east-1.elb.amazonaws.com	○	○	○	×	×
2	*****ec.pl	○	○	○	×	×
3	*****res.net	○	○	○	×	×
	www.*****omains.com	○	○	○	×	×
	*****irect-LB3-890977680.us-east-1.elb.amazonaws.com	○	○	○	×	×
4	*****adow.com	○	○	×	×	×
5	*****rhomeguide.com	○	○	×	×	○
6	*****rvice11.ru	○	○	×	×	○
7	*****ubdate.ml	○	○	×	×	○
	*****e.pool.ntp.org	○	○	×	×	○
8	*****.dscg.akamai.net	○	○	×	×	×
	*****z.com	○	○	×	×	○
9	*****z.com	○	○	×	×	×
10	*****z.com	○	○	×	×	×
11	*****z.com	○	○	×	×	×
12	*****.dscg.akamai.net	○	○	○	○	○
13	www.*****note.com	○	○	○	○	○
14	www.google.co.jp	×	×	○	○	○
15	www.google.co.jp	×	×	○	○	○
16	*****.dspb.akamaiedge.net	○	○	×	×	○
17	*****ank.com	○	○	○	○	○
18	awd.*****thchurch.org	○	○	×	×	×

4.3 考察

以下では，評価実験の結果と提案手法の限界について考察する．

評価実験の妥当性について

実験 1，実験 2 共に仮想環境を検知することが確認されているマルウェア検体により，提案手法の有効性を評価したが, 両実験を合わせて評価用検体を 23 検体しか用意できなかった．このため，さらに大規模なデータセットを用いた厳密な評価が必要である．しかしながら，解析環境検知機能を有することが確認されているマルウェア検体の大規模データセットは我々が知る限り存在していない．ビルダ等により自作した検体を除いて，実マルウェア検体の解析環境検知機能は詳細な静的解析を行う以外に正確に把握することが難しいため，そのようなデータセットの生成自体が研究課題となると思われる．また，今回はマルウェア検体の通信挙動のみに着目して比較を行ったが，解析環境検知の実態を正確に把握するためには，内部挙動の観測に基づく評価が必要となる．一方，内部挙動を観測するための機構自体もマルウェアによる検知の対象となるため，評価がより複雑になることが予想される．

マルウェアによる仮想環境検知の実態

実験 1 で用いた RAT 検体 5 体は，いずれも VMWare や Virtual Box で構築した動的解析システムを検知した一方で, そのうち 3 検体は, エミュレータである QEMU で構築した動的解析システムを検知しなかった．実験 2 では，検体 1〜3 は，VMWare で構築した解析システムを検知せず，検体 5〜8 は QEMU を用いて構築した動的解析システムを検知していないと思われる．このように，今回実験に使用したマルウェア検体による仮想環境検知は完全ではなく，一部の仮想環境のみを検知する場合も多いことがわかった．しかし上述の通り，今回使用した検体数が少ないため，傾向分析にはさらに多くの検体の調査が必要といえる．

提案システムの検知の可能性

提案システムは，市販の環境復元ソフトウェアを利用しているため，これらに特有のファイル，プロセス，レジストリの存在から環境復元ソフトウェアの存在が攻撃側に露見し，解析が妨害される恐れがある．一方，環境復元機能をもつソフトウェアは多数販売されており，国内だけでも少なくとも 10 種類以上の製品が存在する. それらの全てを検知する機能を実装するのは攻撃者にとってもコストがかかるため，技術的には可能であっても，現実的には検知が行われない可能性がある．実際，前述の通り，今回の評価実験でも主要な仮想化ソフトウェアである VMWare, Virtual Box, QEMU に対して網羅的な検知を行っていない検体が多数確認されている．また，学校やネットカフェといった施設では，環境復元ソフトウェアが正規のユーザ環境にインストールされているため，このような環境に侵入を試みる攻撃者は正規ユーザ環境と解析環境を別の指標により判別する必要がある．加えて，仮に特定の環境復元ソフトウェアを検知するマルウェアが登場した場合には，それらのソフトウェアの存在を確認する動作自体が正規のプログラムの挙動とはかけ離れていることから，逆にそのようなマルウェアを検知できる可能性がある．なお，解析環境の検知を試みる挙動を捉えてマルウェアを検知する方法としては既に論文[15]などがある．

提案システムは仮想環境を検知するマルウェアに耐性があるという点を除いては従来の動的解析システムと同様であるため，仮想環境検知以外の動的解析検知手法については別途対策が必要である．例えば，動的解析環境にはユーザ操作履歴が欠如しているという特徴を用いた検知 [16, 17]などが懸念されるが，この問題を解決するためには，論文[18]で提案されているユーザの操作履歴を付加する方法と提案手法を組み合わせることが望ましい．

OS より低いレイヤを変更するマルウェア

環境復元ソフトウェアは，3.1 節で説明したように保護対象パーティションへの書き込み動作を保存領域に書き込み，復元する際には保存領域の内容を削除する，といった復元方法を用いているものが多い．したがって，MBR に感染するマルウェア[19]のように OS より低いレイヤを変更するマルウェアに感染してしまった場合，正しく解析環境を復元することができない可能性がある. 解決方法として，単純に MBR などマルウェアが感染することが確認されている領域へのアクセスを禁止することなどが挙げられるが, この検証は今後の課題とする．

5. まとめと今後の課題

実機を用いたマルウェア動的解析システムを，環境復元ソフトウェアを用いて容易に構築する方法を提案し，仮想環境を検知する RAT 検体と実マルウェア検体を用いて, 提案手法の有効性を示した．

今後の課題は，大規模なマルウェア検体データセットによる評価実験と MBR に感染するマルウェアのように OS より低いレイヤを変更するマルウェアに対する評価である．また，今回使用した 2 種類の環境復元ソフトウェア以外での実現可能性についても検証する．

謝辞

本研究の一部は文部科学省国立大学改革強化推進事業の支援を受けて行われた．本研究成果の一部は、国立研究開発法人情報通信研究機構（NICT）の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」の支援により得られた．

参考文献

[1] T. Raffetseder, C. Kruegel, and E. Kirda, "Detecting system emulators."

[2] P. Ferrie, "Attacks on virtual machine emulators," Symantec Corporation, Tech. Rep., 2007.

[3] D. Kirat, G. Vigna, C. Kruegel, "Barebox: Efficient malware analysis on bare-metal", Annual Computer Security Application Conference(ACSAC), 2011, 403-412.

[4] "VMWare", https://www.vmware.com.

[5] “Oracle vm virtualbox,” https://www.virtualbox.org

[6] “bochs: The open source ia-32 emulation project,” http://bochs.sourceforge.net.

[7] F. Bellard, “Qemu, a fast and portable dynamic translator,” in Pro-ceedings of the Annual Conference on USENIX Annual Technical Conference, ser. ATEC ’05, 2005, pp. 41–41.

[8] G. N. Barbosa and R. R. Branco, “Prevalent characteristics in modern malware,” https://www.blackhat.com/docs/us-14/materials/us-14-Branco-Prevalent-Characteristics-In-Modern-Malware.pdf, 2014.

[9] VASUDEVAN, A, AND YERRABALLI, "Cobra: Fine-grained Malware Analysis using Stealth Localized-executions", IEEE Symposium on Security and Privacy, 2006.

[10] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether: malware analysis via hardware virtualization extensions," in Proceedings of the 15th ACM conference on Computer and communications security, ser. CCS '08. New York, NY, USA: ACM, 2008, pp.51-62.

[11] G. Pék, B. Bencsáth, and L. Buttyán, "nether: in-guest detection of out-of-the-guest malware analyzers," in Proceedings of the Fourth European Workshop on System Security, ser. EUROSEC '11. New York, NY, USA: ACM, 2011, pp.3:1-3:6.

[12] “株式会社グリーンフラッシュジャパン”, “Recovery Flash Ver6”, http://www.gfj.co.jp/rf-ver6.htm.

[13] “イーディーコントライブ株式会社”, “法人向け環境復元ソフトウェア Windows 10 対応｜HD 革命/WinProtector ver.6 with Network Controller Corp. Edition ”, http://www.safety-disclosure.jp/lp/winp.

[14] "Virus Total", https://www.virustotal.com.

[15] D. Kirat, G. Vigna, and C. Kruegel, “Barecloud: Bare-metal analysisbased evasive malware detection,” in Proceedings of the 23rd USENIX Security Symposium, 2014, pp. 287–301.

[16] A. Yokoyama, K. Ishii, R. Tanabe, Y. Papa, K. Yoshioka, T. Matsumoto, T. Kasama, D. Inoue, M. Brengel, M. Backes, and C. Rossow, “SandPrint: Fingerprinting malware sandboxes to provide intelligence for sandbox evasion,” in Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2016, pp. 165–187.

[17] D. Desai, “Malicious Documents leveraging new Anti-VM & Anti-Sandbox techniques,” https://www.zscaler.com/blogs/research/malicious-documents-leveraging-new-anti-vm-anti-sandbox-techniques, 2016.

[18] 田辺瑠偉，八幡篤司，石井攻，横山日明，吉岡克成，松本勉，"サンドボックス解析回避への耐性を高めるツール SandVeil の提案"， 情報セキュリティ研究会 2017(ISEC2017)，2017，pp. 43-49

[19] “PETYA Crypto-ransomware Overwrites MBR to Lock Users Out of Their Computers”, http://blog.trendmicro.com/trendlabs-security-intelligence/petya-crypto-ransomware-overwrites-mbr-lock-users-computers/