情報処理学会論文誌 Vol.56 No.9 1730–1744 (Sep. 2015)

多種環境マルウェア動的解析システムの提案および評価

仲小路 博史[1,2,a)] 重本 倫宏[2] 鬼頭 哲郎[2] 林 直樹[2] 寺田 真敏[2] 菊池 浩明[1]

受付日 2014年12月8日, 採録日 2015年6月5日

概要：近年，標的型攻撃に利用されるマルウェアが高度化し，既存の入口対策で検知できないまま組織内へ侵入を許してしまうケースが増えている．この場合，侵入したマルウェアの特性を解明し早急な被害拡大防止策を講じる必要がある．マルウェアの特性を解明する手法としてマルウェアを特殊な解析環境で実行して挙動を観測する動的解析手法が用いられている．一方，最近のマルウェアは実行環境を限定することで解析環境での解析を逃れるタイプも確認されている．本論文では，マルウェアを多種類の解析環境で実行させることで環境を選ぶマルウェアをも自動的に解析し，その挙動や動作環境を推定してレポートする多種環境マルウェア動的解析システム（M3AS）を提案する．

キーワード：マルウェア，動的解析，動作環境推定

Proposal and Evaluation of Multimodal Malware Analysis System with Multiple Types of Sandboxes

Hirofumi Nakakoji[1,2,a)] Tomohiro Shigemoto[2] Tetsuro Kito[2] Naoki Hayashi[2]
Masato Terada[2] Hiroaki Kikuchi[1]

Received: December 8, 2014, Accepted: June 5, 2015

Abstract: In recent years, a number of incursions into the organization has increased. In the situation, we should clarify the characteristics of intruding malware so that countermeasures can be taken quickly to prevent the damage from expanding. A dynamic analysis method is used in order to clarify the malware's behavior. Recently, however, some types of malware avoid being analyzed in analytical environments by detecting environment. In response, we develop the M3AS, a Multi-modal Malware Analysis System. This system lets malware executes under a variety of analytical environments so that malware that only runs under a specific environment can be automatically analyzed.

Keywords: malware, dynamic analysis, estimate targeted environment

1. はじめに

近年，標的型攻撃に代表される高度なサイバー攻撃が企業や国家にとって大きな脅威となっている．企業の情報セキュリティに関する調査報告 [1] によると，情報セキュリティに関わる事件・事故の原因のトップは依然としてクライアント PC のウイルス感染によるものとされている．さらに Symantec 社や NTT Group の研究グループによると，新種のマルウェアの半数程度が既存のウイルス対策ソフトでは検知できないと報告されている [2], [3]．このような状況下でマルウェアが組織の中に侵入してしまった場合には，侵入したマルウェアの特性を解明して被害拡大防止策を早急に講じることが重要となる．

マルウェアとしてどのような機能を有するかを解明する方法には，検体をリバースエンジニアリング等の技術によって解析する静的解析手法 [4] と，検体を特殊な解析環境で実行して，その振舞いを観測する動的解析手法 [5] の

[1] 明治大学
Meiji University, Nakano, Tokyo 164–8525, Japan
[2] 株式会社日立製作所
Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[a)] hirofumi.nakakoji.vt@hitachi.com

本論文はマルウェア対策研究人材育成ワークショップ 2014 で著者が執筆した「多種環境マルウェア動的解析システムの提案」を元としている．

2 種類がある．静的解析手法は，検体の具備する機能のすべてを詳細に解明できる利点があるが，プログラムや OS，ハードウェアの仕組み等に関する深い知識と，コードを 1 行ずつ読み解くための膨大なコストが必要となる．動的解析手法は，難読化（コード暗号化等）された検体でも容易に挙動を解析できるため，静的解析手法と比較して解析に要するコストが少ない点や，静的解析手法だけでは分からない挙動（たとえば新たなマルウェアをインターネットからダウンロードして実行した後の挙動等）を確認できる点で有利である．一方で，観測中に顕現しない機能はその挙動を把握できないという欠点もある．通常，マルウェア解析の現場では 1 つの検体に費やせる解析時間や人員等のリソースが限られていることから，検体の性質，解析の目的，解析者のスキルセットや経験則に応じて静的解析手法と動的解析手法とを補完的に組み合わせて実施する．本論文では，動的解析手法に着目して，従来の手法では顕現しにくいマルウェアを自動で解析するシステムを提案する．

マルウェアの動的解析にあたっては，マルウェア解析者が Sysinternals [6] 等のトラブルシューティングツールを使用して手作業でマルウェアを解析する方法や，解析を支援する動的解析サービス，動的解析ソフトウェアを利用する方法等がある．オンラインサービスでは Anubis [7] や，ThreatExpert [8] が，オフラインツールでは Cuckoo Sandbox [9] や Threat Analyzer [10]，WildFire [11] が提供されている．これらのサービスやツールはサンドボックスと呼ばれる環境の上で検体を安全に実行して，ネットワーク通信や API コール等の観測結果を自動的に取得する．このため解析を実務とする多くの専門家によって利用されており，MWS 2014 Datasets の 1 つである FFRI Dataset 2014 [12] にも同ツールによって取得したデータが含まれる．

これまでに述べたように守る側は，技術やツールの進化により検体の解析が容易となってきた．一方で攻撃側の進化も著しく，作成したマルウェアが検知されたり動的解析されたりすることを回避するために，マルウェアが仮想環境やデバッグ環境を検出して動作を停止する耐解析機能 [13], [14] や，OS，インストールアプリケーション等のハードウェア/ソフトウェア構成を検出して攻撃の対象であるか否かを判断して動作を変える環境検知機能 [15] を備えた環境選択型マルウェアの存在が確認されている．たとえば，川古谷らは前述した Sysinternals の実行を検知して動作しなくなる検体について報告している [16]．また，ソフトウェアの脆弱性を狙ったマルウェアは，そのソフトウェアが存在しない解析環境では動作が失敗してしまうため，結果的には環境によって挙動が変化することになる．さらに，攻撃者が用意したマルウェア配布サーバから第 2 のマルウェアをダウンロードさせることで攻撃を段階的に進めるダウンローダ型マルウェア [17] も確認されている．マルウェア配布サーバの中には，アクセス元の IP アドレスが攻撃対象の組織である場合のみマルウェアを配布し，それ以外の場合には正規のコンテンツを配布することで第三者によるマルウェア解析を回避するものまで確認されている [18], [19]．

特定の環境しか用意されていない既存の動的解析ソフトウェアやサービスによる解析では，これらのマルウェアの挙動を十分に明らかにできないという問題があった．このため，感染後の対策（インシデントレスポンス）が迅速あるいは適切にとれずに被害が甚大化するリスクがあった．

この問題に対して，Inoue らは隔離されたネットワークに接続した動的解析環境でマルウェアを解析する手法を提案している [20]．近年のネットワークに依存したマルウェアを解析する手法として有用であるが，ネットワーク以外の環境に依存するような環境選択型マルウェアに対する有効性については言及されていない．山口らは，環境選択型マルウェアの実行環境に関して，数種類の実行環境における解析前後の変化を比較する手法を用いて環境種別による挙動変化を確認している [21]．また，環境選択型マルウェアへの解析アプローチとして，Xu らは，環境調査を行う API を監視し，様々な値を返すことで，環境依存型のマルウェアを効率的，効果的に解析し，従来方式と比較して時間・メモリ空間ともに消費を低減させる手法を示している [22]．しかし，本手法は，マルウェアが環境の調査に用いる OS の API を網羅的に監視および分析する必要がある．さらに，ミドルウェアやアプリケーションの脆弱性を狙うマルウェアの振舞いを明らかにすることは困難である．

また Kirat らは，プラットフォーム（物理 PC や仮想 PC）環境で動きを変化させるマルウェアを，観測されるログの階層類似度から精度高く判別する手法を提案している [23]．しかし，環境選択型マルウェアの中には，プラットフォームを判別して意図的に動作を変える以外にも，前述したようなソフトウェアの脆弱性の有無により結果的に動作が変わってしまう種類も存在するが，この種の環境選択型マルウェアに対する効果について言及されていない．

そこで本論文では，既存の動的解析ツールを活用して複数の解析エンジン，異なる環境のサンドボックス群上でマルウェアを同時並列で実行してその挙動を観測，挙動解明，動作環境のアソシエーション分析をする多種環境マルウェア動的解析システムを提案する．複数のサンドボックスを並列に用いることにより，プラットフォームや OS，アプリケーション環境を選ぶマルウェアであっても，用意されたいずれかの環境で顕現する確率が高まる．マルウェア感染後のネットワークアクセス等の挙動の有無と，その挙動が確認されたサンドボックスの環境から，そのマルウェアが動作する環境の条件を推定する．また，すべての処理を自動化することにより，従来，多種類の環境を用いた手作業による試行錯誤と比較して大幅な効率化を図る．本論文の提案手法と既存技術・研究との比較を**表 1** にまとめる．

表 1 既存技術との比較
Table 1 Technical comparison with the existing countermeasures.

		本論文提案手法 M3AS	BareCloud	GOLDENEYE	Palo Alto Networks WildFire
概要		多種環境を用いた環境選択マルウェアの特徴抽出と環境条件推定	階層類似度を用いた環境選択マルウェア検出	環境調査ＡＰＩ監視による環境選択型マルウェア検出	解析エンジン
サンドボックスカスタマイズ性		○	×	×	×
環境選択型マルウェア対応状況	プラットフォーム（仮想 PC 検知）	○ 物理 PC を利用	○ 物理 PC を利用	△ API の戻り値を変化させた解析結果との比較	○
	OS	○	×	△ 同上	×
	アプリケーション	○	×	×	×
環境条件推定		○	△ アプリ未対応	△ アプリ未対応	×

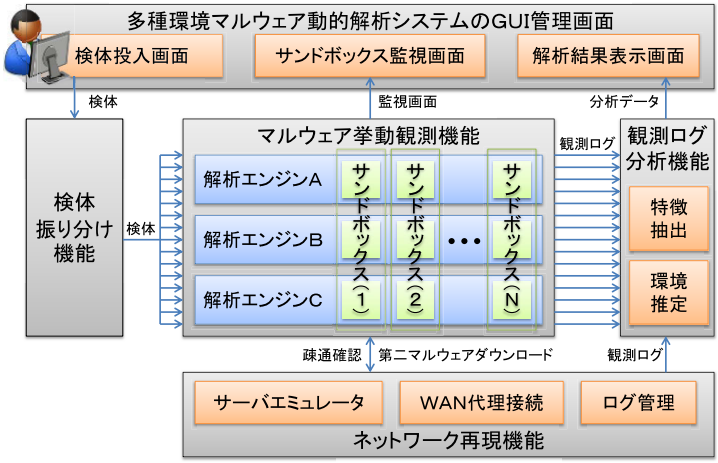


図 1 多種環境マルウェア動的解析システムの機能概要
Fig. 1 Function of the Multi-modal Malware Analysis System.

本論文の構成について述べる．2 章では多種環境マルウェア動的解析システムを提案する．3 章では 2 章で提案したシステムの実装と評価について示す．4 章は結論である．

2. システム提案

本研究では，1 章で述べた課題を解決する多種環境マルウェア動的解析システム（Multi-modal Malware Analysis System，M3AS と略記）を提案する．

M3AS は環境選択型マルウェアの解析成功率を向上させるため，複数種類の解析エンジン，複数種類の解析環境（サンドボックス）を用いて検体を解析する．本システムのアーキテクチャを図 1 に示す．

ここではシステムの機能を概説する．GUI 管理画面に構成される検体投入画面は，解析者の操作によって検体を M3AS に投入（アップロード）するインタフェースである．検体振り分け機能は，投入された検体を，各サンドボックスに投入する．マルウェア挙動観測機能は，投入された検体をサンドボックス上で自動的に実行して挙動を観測し，結果を観測ログとして出力する．出力ネットワーク再現機能はサンドボックスと通信可能で，インターネット環境を再現する．観測ログ分析機能は，マルウェア挙動観測機能やネットワーク再現機能からログを取得し，各サンドボックスにおける検体の活動状況（たとえばファイルアクセス，レジストリアクセス，ネットワークアクセス等）を統計的に分析したり，検体による生成ファイルや，ネットワーク接続先 URL を抽出したり GUI 管理画面へ分析データを提供する．マルウェアを実行中のサンドボックスの状況はサンドボックス管理画面に示される．これらの処理を同時並行かつ自動的に実施するため，解析時間の大幅な短縮や，解析作業の夜間バッチ化も期待できる．

以降では各機能について詳細に述べる．

2.1 検体振り分け機能

検体振り分け機能は，マルウェア解析者の操作する検体投入画面より投入された検体を複製して，あらかじめ登録されたマルウェア挙動観測機能の各解析エンジンに同時に振り分ける．解析エンジンごとに検体入力インタフェースが異なるため，本機能は個々のエンジンに合わせたインタフェース（WEB API 等）を実装し，非互換を吸収する．また，パスワードを指定して暗号化されたアーカイブファイルが投入された場合には，本機能によってアーカイブファ

表 2 サンドボックス環境の環境条件例
Table 2 Examples of the environmental conditions of the sandboxes.

解析エンジン	プラットフォーム	アーキテクチャ	OS	OS 言語	アプリケーション
Threat Analyzer	・物理 PC ・仮想 PC (VMware ESXi)	・32bit(x86) ・64bit(x64)	・Windows XP ・Windows Vista ・Windows 7	・英語 ・日本語	・Microsoft Office ・Adobe Reader ・Adobe Flash ・Internet Explorer ・Java ・Media Player ・一太郎
Cuckoo Sandbox	・仮想 PC (Virtual Box)				
計 2	3	2	3	2	7

表 3 ネットワーク再現機能実装サービス
Table 3 Implemented services of network emulation function.

TCP	UDP
echo(7), http(80), discard(9), pop3(110), daytime(13), ident(113), quotd(17), chargen(19), https(443), ftp(21), smtps(465), smtp(25), time(37), ftps(990), dns(53), pop3s(995), irc(6667), finger(79)	echo(7), discard(9), quotd(17), ntp(123), chargen(19), syslog(514), time(37), dns(53), tftp(69)

（括弧内は利用ポート番号）

イルを復号して振り分ける．これによって，マルウェア解析者によるマルウェアの誤実行を防止する．

2.2 マルウェア挙動観測機能

M3AS は検体を数十種類のサンドボックスで解析することにより，環境選択型マルウェアの解析効率向上を実現する．サンドボックス群は解析エンジンやプラットフォーム，ソフトウェアの種類やバージョン等の異なる組合せにより構成される．サンドボックスが多いほど環境選択型マルウェアの解析成功率向上が期待できるが，使用できる物理マシンのリソースや，ソフトウェアライセンス費用等の制約により，すべての組合せを用意することは現実的でない．そこで，サンドボックスを効率的に設計するため，構成要素を「環境条件」として**表 2** に示す解析エンジン，プラットフォーム，アーキテクチャ，OS，OS 言語，アプリケーションの 6 項目に分類する．「解析エンジン」は，前述した Threat Analyzer や Cuckoo Sandbox を指す．解析エンジンは種類によってサポートする仮想マシンが異なっている．したがって，解析エンジンの多種化は，サンドボックスのプラットフォームの多様化にもつながるため，特定のプラットフォーム（後述）を検出して挙動を変えるような耐解析機能を有するマルウェアの解析にも効果が期待できる．「プラットフォーム」は，OS を動かすハードウェア部分のことで，物理環境や VMware，Virtual Box 等の仮想化環境を指す．「OS」や「アプリケーション」は種類やバージョン等のバリエーションが多く，組合せが膨大となる．

たとえば，シンプルな構成である表 2 の場合でも，アプリケーションが未インストールの場合も考慮すると $2 \times 3 \times 2 \times 3 \times 2 \times (7+1) = 576$ 通りの環境がありうる．そのため絞り込む必要があるが，マルウェア開発者の視点に立ってマルウェアが感染および動作しやすい環境，つまり，攻撃の影響を受けやすい環境を優先的に選定する（3 章で詳述）．なお，マルウェア挙動観測機能には，各サンドボックスは検体の実行完了，あるいは事前に設定した時間を経過すると自動的に環境を感染前へ復元する機能も備える．

2.3 ネットワーク再現機能

近年のマルウェアはネットワーク接続機能を有し，マルウェア配布サーバに接続して第 2 のマルウェアをダウンロードしたり，C&C サーバと接続して遠隔操作を受けたりすることが知られている．マルウェアの中には，実行直後にネットワークの疎通性を確認することにより，解析のための閉塞ネットワーク環境で自身が実行されている否かを検出して動作を停止させる等，解析を阻害するタイプも確認されている．青木らも閉塞環境よりも開環境の方がより多くの動的解析結果を得られると報告している [5]．このため，M3AS はネットワーク再現機能を備え，サンドボックス内の検体からの各種サーバ向けリクエストに応答するサーバエミュレータ機能を持つ．これにより，ダウンローダ型マルウェアが Web サーバからファイルをダウンロードして実行するまでの挙動を再現，観測することができる．サーバエミュレータは，インターネットサービスシミュレーションソフトウェア「INetSim」[24] を用いて**表 3** に示す 21 のサービスのエミュレーションを行う．

2.4 観測ログ分析機能

観測ログ分析機能は，数十種類のサンドボックスで観測した大量の観測ログからマルウェア特有の挙動の抽出や，マルウェアが動作する環境の条件を推定する．

2.4.1 マルウェア特徴抽出機能

マルウェアの機能的な特徴を抽出する機能の設計にあたっては，サイバー攻撃観測記述形式 CybOX [25] でサポートされているアクションをマルウェアの挙動抽出対象の参考とした．マルウェアの特徴は攻撃手法の進化によって変

化することから，マルウェア特徴抽出機能（モジュール）もその進化に合わせて追加・修正する必要がある．このため，モジュールをプラグイン式にすることで，柔軟に特徴抽出機能の追加や修正，削除できる仕組みを採用する．

以下に本論文で実装した 3 種のモジュールを例示する．なお，下記以外にも動的解析を逃れるために一定時間動きを停止する挙動や，マルウェアを自動起動するためにスタートアップへ追加する挙動の有無を判定に用いるモジュール等が考えられる．

(1) デバッガ検出の有無判定
耐解析機能を備える検体がデバッガによって解析されることを回避することを目的としてよく利用するデバッガモード判定 API（IsDebuggerPresent 等）の呼び出しを監視する．通常のプログラムでは，本 API を呼び出すことが少ないため，マルウェアの判定に利用できる．

(2) プロセスインジェクションの有無判定
検体が他のプロセスに不正なコードを挿入する際に利用する API（WriteProcessMemory 等）の呼び出しを監視する．マルウェアは，Internet Explorer 等の正規なプロセスにコードを挿入することで，自身の機能を iexplore.exe に隠ぺいしたり，Internet Explorer のパーソナルファイアウォールの設定ポリシを継承したりするために，プロセスインジェクション機能を悪用する．この特性を判定に利用する．

(3) 外部ネットワーク接続判定
検体が第 2 のマルウェアのダウンロードや，C&C サーバとの通信を実行する際に発生するネットワーク通信の内容を監視する．

本システムでは，上記モジュールの判定結果が所定の条件を満たしたことを，マルウェアの特徴を**検知**したと定める．所定の条件とは，たとえば「(1) の判定結果が有の場合」や「(3) の判定結果が外部のホストに接続した場合」である．

2.4.2 マルウェア動作環境推定

1 章で述べた環境選択型マルウェアは，M3AS を構成する複数のサンドボックスで実行しても一部のサンドボックスでしか動作しない．環境選択型マルウェアが動作するサンドボックスが 1 つでも存在すれば，そのマルウェアの挙動の抽出が可能である．加えて，そのマルウェアの動作条件が特定できれば，マルウェアが動作する環境条件として，さらなる詳細解析（人手による動的解析や静的解析等）に役立てることができる．また，感染可能性の有無が推定できるため，マルウェアによる影響範囲を特定する情報としても有用である．

ここでは，環境選択型マルウェアにサンドボックス環境が適合して動作したか否かを判定するにあたり，2.4.1 項に述べたマルウェア特徴抽出機能のうち，「外部ネットワーク接続判定」の結果を用いて確認した検知有無をマルウェアの**顕現状態**と定義する[*1]．また，顕現状態が有と確認できた検体を**顕現マルウェア**，同様に顕現状態が有と確認できたサンドボックスを**顕現サンドボックス**と定義する．

M3AS では顕現マルウェアの顕現サンドボックスの環境条件を絞り込むために，データマイニング手法として広く利用されているアソシエーション分析を適用し，後述するアソシエーションルールを抽出する．本論文では，アソシエーション分析の 1 種である Apriori アルゴリズムを用いることで，顕現状態と環境条件の関係性を示すアソシエーションルールと，その指標である支持度（support），確信度（confidence），リフト値（lift）を求める．

サンドボックスにおけるマルウェアの顕現状態 X を条件部，当該サンドボックスの環境条件 Y を結論部とするアソシエーションルール $X \Rightarrow Y$ の抽出を試みる．

アソシエーション分析の入力データは，M3AS によって得られるサンドボックスの観測ログに基づいて作成する．各サンドボックスの顕現状態と環境条件とを組として 1 つのトランザクションとする．サンドボックスの数 M だけトランザクションを作成する．環境条件数を N，アイテム A を含むトランザクションの数を $\sigma(A)$ とする．アソシエーションルール $X \Rightarrow Y$ に対し，支持度（support），確信度（confidence），リフト値（lift）は次の式によって求められる．

$$\mathrm{support}(X \Rightarrow Y) = \frac{\sigma(X \cap Y)}{M}$$

$$\mathrm{confidence}(X \Rightarrow Y) = \frac{\sigma(X \cap Y)}{\sigma(X)}$$

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{confidence}(X \Rightarrow Y)}{\mathrm{support}(Y)} = \frac{\mathrm{confidence}(X \Rightarrow Y) \cdot M}{\sigma(Y)}$$

ここで，英語 OS でしか顕現しないマルウェアを例に，サンドボックスの環境条件，および解析結果から得られた顕現状態の関係を**表 4** に示す．さらに，環境条件 Y_1 を Threat Analyzer，Y_2 を Windows XP，Y_3 を英語 OS として，本例から生成した $M = 3$，$N = 3$ のトランザクションを**表 5** に示す．またトランザクションからアソシエーション分析して求めたアソシエーションルールと各指標を**表 6** に示す．

このように，マルウェアが顕現した場合の環境条件として Threat Analyzer，Windows XP，英語 OS のそれぞれに関するルールが抽出できる．その中でも，確信値やリフト値の高いルールを抽出することで，マルウェアの顕現状態 X の論理条件を明らかにすることができる．通常，支

[*1] デバッガ検知やプロセスインジェクションを顕現状態の定義に用いない理由は，両者がマルウェアの不正活動の前段階にみられる挙動を検知するモジュールであり，環境の適合有無に限らず検知される可能性が高く，環境選択型マルウェアの顕現状態の判断に向かないためである．

表 4 解析結果例
Table 4 Examples of analysis results.

SB#	顕現状態	環境条件
1	有	Threat Analyzer, 英語
2	有	Windows XP, 英語
3	無	Windows XP

表 5 トランザクション例
Table 5 Examples of transactions.

SB#	顕現状態X	環境条件Y_1	環境条件Y_2	環境条件Y_3
1	1	1	0	1
2	1	0	1	1
3	0	0	1	0

表 6 アソシエーションルールと指標
Table 6 Association rules and metrics.

ルール	support	confidence	lift
$X \Rightarrow Y_1$	1/3	1/2	3/2
$X \Rightarrow Y_2$	1/3	1/2	3/4
$X \Rightarrow Y_3$	2/3	1	3/2

持度が大きいほど一般化されたルールである．また，リフト値が大きいほど環境条件を満たすサンドボックスでマルウェアが顕現する可能性が高いことを示している．

マルウェア動作環境推定機能の目的は，入力データから，顕現マルウェアにおける顕現サンドボックス群の多くに共通の環境条件（以降，顕現条件と記す）を抽出することである．このため，Apriori で用いる確信度下限は 1.0，すなわち確信度が 1.0 のルールのみを抽出する．また，支持度下限は多数サンドボックスの中の少数派の環境条件をも抽出対象とするために，2 つ以上のサンドボックス間の共通因子が抽出可能な値である $2/M$ とする．これらの設定によって抽出したルールから条件部のアイテム数が単一かつサンドボックスの顕現状態となるルールを抽出する．検体によっては複数のルールが抽出されうるが，その複数のルールの結論部に含まれる環境条件の論理積が，当該検体が顕現したサンドボックスに共通の環境条件，すなわち顕現条件といえる．再度，表 6 を用いて説明すると，顕現条件を示すルールは確信値およびリフト値ともに高い値を示したルール $X \Rightarrow Y_3$，すなわち「英語 OS」となり，表 4 で示した前提条件と一致する．

2.5 解析結果表示機能

M3AS は，数十種類のサンドボックスでの検体の動作結果のサマリと，個々のサンドボックスの解析結果を集約して一覧表示する．

解析者はこの解析結果表示機能を利用して，検体の接続先 URL や作成ファイル，生成プロセス情報，マルウェアの顕現条件を確認する．検体がマルウェアであった場合に，感染端末によるネットワークアクセスの接続先ホスト，感染端末に仕掛けられたトラップ（マルウェア関連ファイル）等を容易に把握することができる．これらの情報を用いてファイアウォールやプロキシ等で接続先ホストへの通信を禁止したり，ウイルス対策ソフトのパターンファイルに駆除情報を追加したりする．これにより企業の入口対策をすり抜けて従業員の端末に感染・発症した場合でも，感染端末におけるトラップの排除等の内部対策や，接続先ホストへのアクセス制限等の出口対策を活用した多層防御が可能となる．また，個々のサンドボックスの解析結果は本機能によって生成された画面を確認することにより，サンドボックス単位で検体の活動状況（ファイルアクセス，レジストリアクセス，プロセス操作，ネットワークアクセス）や 2.4.1 項の判定結果を確認することができる．

3. システム実装と評価

ここでは，M3AS をプロトタイプ実装し，2014 年 10 月に著者らが独自に入手したマルウェア 633 種を用いて，解析処理時間およびマルウェア動作環境推定手法を評価した結果について述べる．

3.1 システム構成

M3AS のシステム構成を**図 2** に示す．M3AS には解析者の操作によって検体を受け取り，Threat Analyzer システムや Cuckoo Sandbox システムに検体を振り分ける検体投入サーバ①がある．Threat Analyzer システムは，受け取った検体をサンドボックスに振り分ける管理サーバ②と，振り分けられた検体を実行して挙動を観測する仮想サンドボックス群③や物理サンドボックス群④によって構成され，Cuckoo Sandbox システムは受け取った検体を実行して挙動を観測する仮想サンドボックス群⑤によって構成される．また，各サンドボックスによるインターネット等へのネットワークアクセスに対して擬似的な応答を返すネットワーク再現機能を備えたネットワーク再現サーバ⑥を設置する．さらに，Threat Analyzer システムや Cuckoo Sandbox システム，ネットワーク再現サーバから検体の挙動の観測ログを収集，蓄積する解析結果統合データベース⑦と，当該データベースのデータに基づき，検体の特徴抽出や動作した環境の条件を推定するログ分析機能，および結果を表示する機能を備えた可視化サーバ⑧があり，解析者は当該サーバの出力画面を閲覧して検体の挙動や特性を把握する．

以降では，サンドボックスの構成や解析結果，環境推定について詳述する．

3.2 サンドボックス構成

本節では，評価用 M3AS のサンドボックスの構成について述べる．表 2 の環境条件を詳細化した 11 カテゴリ，44 項目の環境条件を**表 7** に示す．また，各環境条件の選定方針を**表 8** に示す．

解析エンジンは，物理 PC をサンドボックスに含めるこ

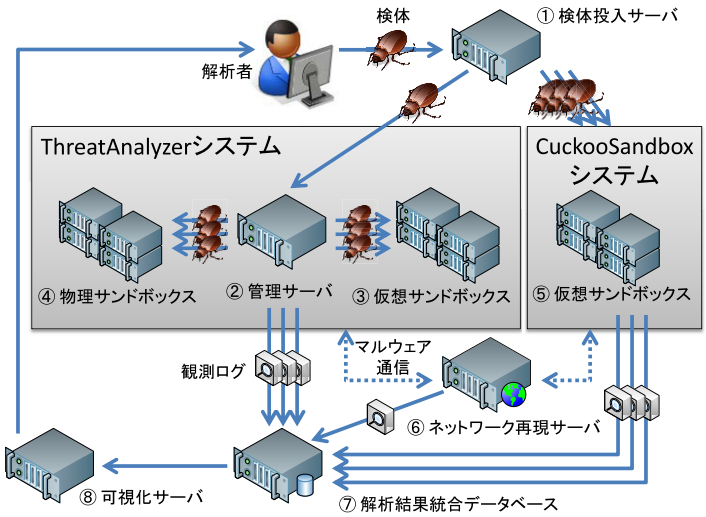


図 2 M3AS システム構成
Fig. 2 System architecture of M3AS.

表 7 評価向けサンドボックスの環境条件
Table 7 Sandbox environment used for evaluation.

環境条件カテゴリ	環境条件	数
解析エンジン	Threat Analyzer, Cuckoo Sandbox	2
プラットフォーム	VMware ESXi, Virtual Box, 物理	3
OS	Windows XP sp(2,3), Vista sp(0,1,2), 7 sp(0,1)	7
OS 言語	日本語, 英語	2
Microsoft Office	Null, 2007, 2010, 2013	4
Adobe Reader	Null, 8, 9, 10, 11	5
Adobe Flash	Null, 10, 11	3
Internet Explorer	6, 7, 8, 9, 10	5
Java	Null, 1.4, 5, 6, 7	5
Media Player	Null, 11, 12	3
一太郎	Null, 2013 玄, 2013 玄 Trial, Viewer19, Viewer23	5
合計		44

表 8 環境条件の選定方針
Table 8 Selection policy of the environmental condition.

環境条件カテゴリ	選定方針
解析エンジン	Threat Analyzer の物理 PC，仮想 PC，Cuckoo Sandbox の仮想 PC の 3 種類を均等配分するため，Threat Analyzer と Cuckoo Sandbox を 2:1 の割合で配分
プラットフォーム	概ね均等に配分
OS	
OS 言語	希少条件（英語版）を物理サンドボックスに優先割り当て
Microsoft Office	OS の新旧に合わせて配分
Adobe Reader	
Adobe Flash	
Internet Explorer	
Java	
Media Player	
一太郎	希少条件（一太郎 2013 玄）を物理サンドボックスに優先割り当て

とが可能な Threat Analyzer version 4.1（有償）と，物理 PC には非対応だがオープンソースソフトウェアとして提供されている Cuckoo Sandbox version 0.6 を選定した．これにより限られたコストで，仮想 PC を検出して動作を停止してしまうマルウェアへの対策と，サンドボックスの多種化とを両立させる．プラットフォームは前述したように物理 PC と仮想 PC とを選定した．仮想 PC は，前述した 2 種の解析エンジンの推奨仮想化ソフトウェアである VMware ESXi と VirtualBox を選定した．OS は，マルウェアの感染が多く報告されている Windows XP 以降の主要 OS を Service Pack まで区別してすべて選定した．ただし前述した解析エンジンのサポート範囲の制約により，Windows 8 は選定対象からはずしている．また同様の理由からアーキテクチャも 32 bit 版に限定している．通常のアプリケーション同様に，日本語環境対応していないマルウェアが存在することが想定されることから OS 言語には日本語と英語を選定した．

また，アプリケーション構成は脆弱性が多いアプリケーション，すなわち脆弱性情報の公開数の多いアプリケーションを優先的に選定した．脆弱性情報の公開数の調査には，JVN iPedia [26] の 2012 年 1 月 1 日から 2013 年 8 月 16 日までの情報を利用した．表 7 にある「sp0」はサービスパックが未適用のバージョンのことを指し，「Null」はソフトウェア自体がインストールされていないことを指す．なお，有償ソフトウェアや入手容易性等の理由により多くのライセンスを用意するのが困難な「OS 言語＝英語」や，「一太郎 2013 玄」は，環境選択型マルウェアの動作しやすい物理 PC のサンドボックスに優先的に割り当てている．

本論文の評価では，この $N = 44$ 項目の環境条件を組み合わせて $M = 76$ 種類のサンドボックスを用いる．**表 9** に一部を例示する．

3.3 マルウェアの解析

前節で示したサンドボックス構成を備える M3AS を用いて，41,108（76 サンドボックス観測ログ/検体 × 633 検体）のサンドボックスの観測ログを取得した．また，Windows に標準でインストールされているメモ帳（notepad.exe）や電卓（calc.exe）等のほか，空のドキュメント（DOC, XLS,

表 9 環境条件 $Y_1, \cdots, Y_N$ の例（一部）
Table 9 Examples of the environmental conditions $Y_1, \cdots, Y_N$.

SB #	Threat Analyzer	Cuckoo Sandbox	VMware ESXi	Virtual Box	物理	Windows XP sp2	Windows XP sp3	…	一太郎 Viewer19	一太郎 Viewer23
1	1	0	1	0	0	1	0	…	0	0
2	1	0	0	0	1	0	1	…	0	1
3	0	1	0	1	0	0	1	…	1	0
:	:	:	:	:	:	:	:	:	:	:
N(=76)	0	1	0	1	0	0	0	…	0	1

表 10 モジュールの検知結果
Table 10 Experimental results of detected modules.

項目	検知率	誤検知率
(1) デバッガ検出の有無判定	58.1% (368)	40.0% (4)
(2) プロセスインジェクションの有無判定	40.1% (257)	0.0% (0)
(3) 外部ネットワーク接続判定	82.8% (524)	0.0% (0)

（括弧内は検体数）

表 11 マルウェアの解析結果
Table 11 Malware analysis results.

項目	数	割合
全マルウェア	633	100.0%
顕現マルウェア	524	82.8%
全サンドボックス	41,108	100.0%
顕現サンドボックス	9,895	24.1%
解析エラー	2,848	7.0%
非顕現サンドボックス	28,365	68.9%

表 12 ネットワーク再現機能へのアクセス状況
Table 12 Traffic log of the network emulator.

通信ポート	通信検体数	割合
80/tcp	518	81.8%
139/tcp	290	45.9%
8080/tcp	257	40.7%
443/tcp	122	19.3%
2869/tcp	55	8.6%
その他	82	9.7%

PPT，PDF，RTF，JTD ファイル）等，明らかにマルウェアではない検体も 10 種類用意して 760（76 サンドボックス観測ログ/検体 × 10 検体）のサンドボックスの観測ログを取得した．これらの観測ログからマルウェア特徴抽出機能モジュールによって得られた検知および誤検知の結果を**表 10** に示す．

633 検体のうち，すべてのモジュールで検知されたマルウェアは全体の 17.5%，すべてのモジュールで検知されなかったマルウェアは 7.0%，1 つ以上のモジュールで検知されたマルウェアは 93.0%であった．後者のマルウェアのハッシュ値を用いて VirusTotal [27] で調査した結果，マルウェアとして登録されていないものや，ネットワーク活動をともなわない古いタイプのワームが多く含まれていた．また，デバッガ検出の有無判定で誤検知した検体は，DOC，RTF，PDF，XLS であった．これらのファイルの解析時には WORD や Acrobat，EXCEL 等，関連付けられたアプリケーションが起動する．これらのアプリケーションのインポートアドレステーブルには「IsDebuggerPresent」が含まれていたため，関連付けられたアプリケーションにデバッガ検出の機能が実装されていることから誤検知したと考える．

本評価では 2.4.2 項で述べたように，「外部ネットワーク接続判定」モジュールを顕現の有無判定と定義する．なお，外部ネットワーク接続先として NTP サーバやソフトウェア更新サーバ，ループバックアドレス等，明らかに無害なホストへのアクセスは判定から除外する．その結果，633 検体のうち顕現マルウェアは 524 検体であった．**表 11** に解析結果を示す．解析エラーは，マルウェアの実行あるいは観測が所期のとおり完了せずに異常終了したことを示し，その原因はサンドボックスのハングアップや，サンドボックスの起動不具合である．また，2.3 節で述べたネットワーク再現機能へのアクセス状況を**表 12** に示す．結果，8 割以上のマルウェアが外部ホストへ 80/tcp を使った通信をしていることが分かった．2869/tcp は UPnP（ユニバーサルプラグアンドプレイサービス）で利用されるポートで，マルウェアが攻撃者と通信チャネルを確立するためのポートフォワーディングを設定する際に利用されることが多い．

3.4 マルウェア解析処理性能

M3AS のマルウェア解析プロセスは大きく分けて，以下の 3 つに分類される．

(1) マルウェア観測処理
(2) ログ分析処理
(3) 環境復元処理

上記の各プロセスにおける各サンドボックスの処理時間の計測結果を以降で示す．評価にあたっては，633 検体のうち無作為に抽出した検体 10 種の上記 3 つの処理時間の平均値をそれぞれ求める．また，本評価に用いる M3AS は，前述した 76 種類のサンドボックスのうち Threat Analyzer（仮想），Threat Analyzer（物理），Cuckoo Sandbox（仮想），それぞれを 15 種ずつ合計 45 種類のサンドボックスを抽出した．結果を**表 13** と**図 3** に示す．

各処理の時間は解析エンジンの違いで異なる傾向が確認できる．全体的に Threat Analyzer の仮想環境の処理時間が長くなっている．これは**表 14** に示すように，Threat Analyzer の仮想環境が 1 台の物理サーバに 13 台動作して

いるため，処理性能上のボトルネックが発生しているものと考えられる．

実際の M3AS におけるマルウェアの解析処理は上記（1）の処理完了後に処理（3）が実行され，そのバックグランドで処理（2）が実行される．このため，マルウェアあたりの平均処理時間は（1）＋（3）の 97.9 秒となる．M3AS はすべてのサンドボックスの解析が終了してから次の検体の解析が開始されることから，システムとしての処理速度は，解析に最も時間のかかったサンドボックスに律速する．このため本システムにおける平均解析時間は処理（1）と処理（3）の最大値の和である 162.6 秒である．

これはスループットに換算すると 22 検体/時間，531 検体/日となる．

3.5 環境選択型マルウェアの顕現条件推定

次に，環境選択型マルウェアが顕現化するためのサンドボックスの顕現条件の絞り込みを行う．

表 13 マルウェア解析処理時間
Table 13 Processing time of malware analysis.

処理内容	平均時間（s）	最大時間(s)
(1) マルウェア観測処理	64.0	116.8
(2) ログ分析処理	83.9	150.2
(3) 環境復元処理	33.9	45.8
合計時間	181.7	312.8

3.5.1 Apriori アルゴリズムの設定

ここでは 2.4.2 項に述べた Apriori アルゴリズムを利用してアソシエーションルール（顕現条件）を抽出するための設定について述べる．

入力データとしては検体ごとに，各サンドボックスの顕現状態（1 アイテム）と，表 7 に示す各環境条件（44 アイテム）を 1 つのトランザクションとし，サンドボックスの数（76 個）だけトランザクションを作成する．

確信度下限は 1.0 とし，サンドボックス数は 76 種類あることから $M = 76$，支持度下限は $2/M = 2/76 \fallingdotseq 0.026$ とする．

3.5.2 アソシエーションルールの抽出

3.3 節で述べた顕現マルウェア 524 検体に対して，アソシエーションルールの抽出を試みた．その結果，ルールが抽出できた顕現マルウェアは 357 検体，総ルール数は 2,106 ルールであった．図 4 に検体ごとに抽出されたアソシエーションルール数と，顕現サンドボックス数の関係を散布図に示す．

両者には負の相関（相関係数にして −0.70）を確認できるが，これは顕現したサンドボックスの数が少なくなると，偶発的に共通する環境条件が増加してしまうことに起因している．具体的には，顕現サンドボックス数が 55 以上のマルウェアからはアソシエーションルールが 1 つも抽出さ

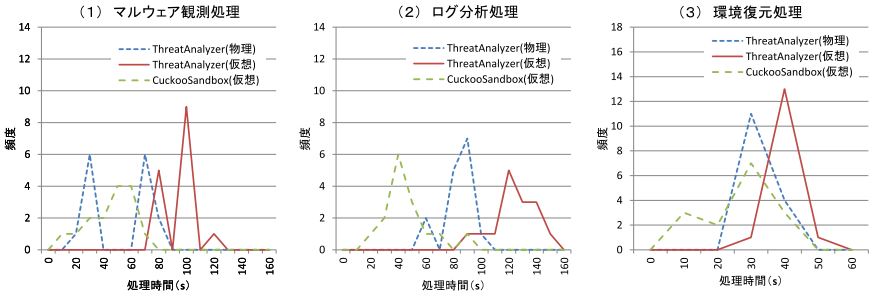


図 3 解析処理時間（サンドボックス単位）
Fig. 3 Analysis time for each sandbox.

表 14 M3AS 構成ハードウェアスペック
Table 14 Hardware specifications of M3AS.

装置名称		ハードウェアスペック	
① 検体投入サーバ		CPU：intel Xeon E5-2690(2.90GHz, 8cores) × 2 Memory：48GB Storage：HDD900GB × 5 (RAID5)	※1
② ThreatAnalyzer 管理サーバ		CPU：intel Xeon E5-2680(2.70GHz, 8cores) × 2 Memory：64GB Storage：SSD 250GB ×16 (RAID5)	
サンドボックス	③ Threat Analyzer 仮想 PC（13 仮想 PC/台 × 2 台）	※1 と同じ	
	④ Threat Analyzer 物理 PC（29 台）	CPU：intel Core i3-2120T(2.60GHz, 2cores) Memory：8GB Storage：HDD250GB	※2
	⑤ Cuckoo Sandbox 仮想 PC（3 仮想 PC/台 × 7 台）	※2 と同じ	
⑥ ネットワーク再現サーバ		※2 と同じ	
⑦ 解析結果統合 DB （2 台）		※1 と同じ	
⑧ 可視化サーバ		※2 と同じ	

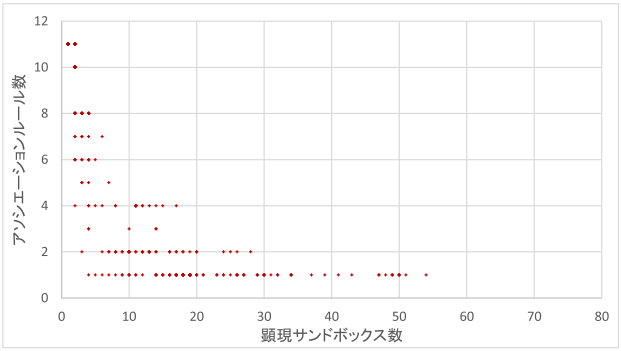


図 4 アソシエーションルール数
Fig. 4 Number of association rules.

表 15 アソシエーションルールの一部
Table 15 Examples of association rules.

検体	条件部X	結論部Y	支持度	リフト値
1	顕現 = 有	解析エンジン = Threat Analyzer	0.64	1.36
2		解析エンジン = Threat Analyzer	0.05	1.36
		プラットフォーム = VMware ESXi	0.05	2.92
		OS 言語 = 日本語	0.05	1.07
		一太郎 = Null	0.05	1.23

れなかった．これらのマルウェアは大半のサンドボックス（72%以上）で顕現したため，環境選択型のマルウェアでなかったことに起因する．また，顕現サンドボックス数が 5 以下（約 38%）のマルウェアからは必ず 1 つ以上のアソシエーションルールが抽出できた．

ここで，抽出されたアソシエーションルールの一部を**表 15** に示す．すべてのアソシエーションルールの確信度は 1.0 であることから，この項目は省略する．検体 1 は顕現したサンドボックスのすべての解析エンジンが Threat Analyzer であったことを示している．また，検体 2 は検体 1 の条件のほかに，プラットフォームが VMware ESXi，OS 言語が日本語，一太郎が未インストールで顕現したことを示している．

このように，M3AS の観測結果をアソシエーション分析することにより，顕現サンドボックス数が少ない顕現マルウェアの動作条件をアソシエーションルールとして正しく抽出することができた．

3.5.3 環境選択型マルウェアのルール抽出精度

本項では前項の環境選択型マルウェアの顕現条件の推定の結果抽出されるアソシエーションルールについて問題と理論的な精度，評価の目的，評価の結果をそれぞれ述べる．

(1) ルール抽出精度に関する問題

アソシエーションルールが期待どおり抽出できるか否かは，各サンドボックスでのマルウェアの実行成否によって決まる．マルウェアは実装上の不具合やヒープ・スプレー等の脆弱性攻撃の不安定性が原因で，動くはずの環境でも実行時エラーが発生することもある．このため，すべてのケースでマルウェアの実行結果から期待どおりのアソシエーションルールが抽出できるとは限らない．さらに，環境条件のすべての組合せについてサンドボックスを用意することは現実的ではないため，顕現条件を識別するのに効果的に考えられる限られた数のサンドボックスを用意して，その結果から正しい顕現状態をマイニングしなくてはならない．

表 16 環境選択型マルウェアサンプル
Table 16 Samples of environment targeted malware.

検体	環境条件	備考
A	一太郎 2013 玄	日本を狙った標的型攻撃で利用されたマルウェア（PlugX[28]の新種[29]）で，一太郎の脆弱性（CVE-2013-5990[30]）を悪用
B	WindowsXPsp2 物理環境	疑似マルウェア．GetVersionEx API を利用して XP sp2 のみで動作．また，搭載 CPU 数を用いた仮想マシン検知方式[31]を利用して物理でのみ動作するよう製作
C	英語版 Windows	検知名 TROJ_FAKEAV.BME[32]で知られている．著者らの静的解析による調査で，Dropper 型の本マルウェアは実行ファイル生成時に文字列変換 API を利用していることが原因で，英語版 OS 以外では不完全な実行ファイルが生成されることが判明

(2) 理論的な精度

上記の不安定性の問題が存在せず，すべての可能な組合せのサンドボックス上で不具合なく解析ができた場合，理論上，アソシエーションルールの支持度，サポート条件を満たすルールは 100%の確率で抽出される．サンドボックスが k 個だけ与えられているとき，識別できる顕現状態は 2^k 通りである．サンドボックス数が有限であるとき，それらを避けるマルウェアが理論上は存在するが，現実的には脆弱性の制約等があるため，ほとんどすべてのマルウェアのルールを抽出できると考えられる．

(3) 評価の目的

そこで，この仮説を検証するため，本論文では，2 つの観点で評価を行う．

1 つ目は，既知の環境選択型マルウェアに対し，解析結果からアソシエーションルールが正しく抽出できるか否かを評価する．具体的には，実在する環境選択型マルウェアで，かつ挙動が解明されている 2 つの検体（A，C）と，著者らが作製した検体（B）を用意し，これらの検体の仕様を正解データ（**表 16**）として用いる．環境選択型マルウェアの解析結果からアソシエーション分析を行い，得られたアソシエーションルールに上記正解データが含まれているか否かを検証する．

2 つ目は，サンドボックス数が限られたときに正しく条件を抽出できるか否かを評価する．前述した問題（各サンドボックスでのマルウェアの実行時エラー）がアソシエーションルールの抽出可否（前述した理論的な精度）に与える影響について検証することで，提案したマルウェア動作環境推定手法の抽出精度を評価する．利用するサンドボッ

表 17 環境選択型マルウェアのアソシエーションルール
Table 17 Association rules of environmental conditions.

検体	結論部Y	支持度	リフト値
A	解析エンジン = Threat Analyzer	0.03	1.36
	プラットフォーム = 物理	0.03	1.07
	OS 言語 = 日本語	0.03	1.07
	Microsoft Office = 2007	0.03	2.62
	Adobe Flash = 10	0.03	2.71
	Java = 1.4	0.03	3.17
	Media Player = 11	0.03	4.47
	一太郎 = 2013 玄	0.03	**25.33**
B	解析エンジン = Threat Analyzer	0.08	1.36
	プラットフォーム = 物理	0.08	2.53
	OS = WindowsXP sp2	0.08	**8.44**
	一太郎 = Null	0.08	1.23
C	解析エンジン = Threat Analyzer	0.07	1.36
	プラットフォーム = 物理	0.07	2.53
	OS 言語 = 英語	0.07	**15.20**
	Java = 1.4	0.07	3.17
	一太郎 = 2013 玄	0.07	1.23
	Adobe Flash = 10	0.07	2.71

クス数を意図的に増減させて不安定性を再現し，3 種の検体のアソシエーションルールが正しく抽出される精度を明らかにする．抽出成否の定義は，表 16 に示した環境選択型マルウェアサンプルの特徴を反映したルールが抽出できたか否かで判断する．サンドボックス数は全 76 種の中から無作為に 5 から 76 まで段階的に数を増やして解析する．また，各段階ではサンドボックスを選びなおしてアソシエーションルールの抽出を 100 回ずつ試み，その成功数からアソシエーションルール抽出成功率（以下，成功率）を算出する．

(4) 評価の結果

1 つ目の評価に関し，環境選択型マルウェアの解析結果からアソシエーションルールを抽出した結果を**表 17** に示す．すべてのルールの条件部 X は「顕現＝有」であることから省略する．検体 A は 8 つのルールが抽出されたが，リフト値に着目すると「一太郎＝2013 玄」が際立っていることから，本条件が検体 A の顕現条件に大きく影響を与えていると判断できる．これは表 16 の条件と一致する．検体 B は 4 つのルールが抽出された．なかでも OS およびプラットフォームのルールでリフト値が高くなっており，顕現条件に強い影響を与えていることが分かる．これも表 16 の条件と一致する．物理環境のサンドボックスを扱える解析エンジンは必ず Threat Analyzer であることから，「解析エンジン＝Threat Analyzer」も抽出されている．また，Windows XP sp2 がインストールされているサンドボックスには一太郎がインストールされていなかったことから，「一太郎＝Null」も抽出されている．さらに，検体 C は「OS 言語＝英語」を含む 4 つのルールが抽出された．検体 C も検体 A と同様にリフト値に着目すると OS の言語が英語であることが顕現条件に大きく影響しており，これも表 16 の条件と一致する．

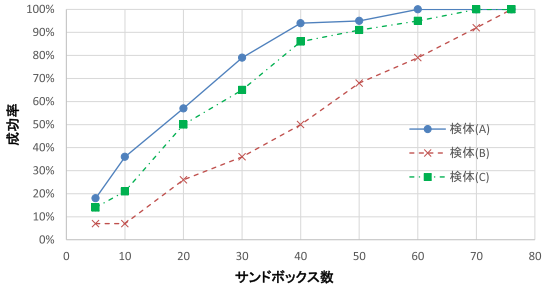


図 5 アソシエーションルール抽出成功率
Fig. 5 Successful extraction of association rules.

次に 2 つ目の評価について述べる．**図 5** に 3 種の検体のそれぞれについて，サンドボックス数と成功率の関係を示す．この結果から，検体 A，B，C はサンドボックス数の増加とともに成功率が向上し，サンドボックス数が 60，70，76 種で 100%となることを確認した．また，リフト値の高い検体 A や検体 C は，対数的な成功率増加傾向がみられ，リフト値の低い検体 B は線形的な増加傾向がみられた．

以上の検証により，環境選択型マルウェアの挙動解明および環境条件の推定が正しく行われていることと，サンドボックスの数がアソシエーションルールの抽出成否の決定に重要であることを確認した．

3.6 サンドボックス構成の課題

前節で実在するマルウェアや評価用の検体の解析結果から，M3AS によって特定のソフトウェアがインストールされた環境でしか動作しない環境選択型マルウェアの解析および環境の絞り込みにアソシエーション分析が有効であることを示した．しかしながら，アソシエーションルールが必要以上に多く抽出されることも分かった．この原因としては，**図 6** の環境条件の分布が示すように，一太郎の各バージョンや英語版 OS を環境条件とするサンドボックスの数が非常に少ないこと，サンドボックス間の環境条件の独立性欠如にあること等が考えられる．

実際に，3.5.1 項で設定した環境条件のみをトランザクションのアイテムとして Apriori アルゴリズムを適用したところ，サンドボックスの環境条件間で 338,777 のアソシエーションルールが抽出された．3.5.2 項で抽出した顕現条件ルールは 1 検体あたり 3.33 ルール（633 検体で 2,106 ルール）であったことから，単純に 3.5.1 項で設定したトランザクションからアソシエーションルールを抽出すると 99.999%が顕現状態とは無関係な環境条件間のルールといえる．理想的にはそれぞれの環境条件は直交性を持つべきであるが，現実的にはサンドボックス数のリソース上の制約や，ソフトウェア間の同居可否による制約が発生する．現状の構成でも，リフト値をもとにそれらのルールから最大因子を推定することは可能であったが，より精緻なルールの絞り込みが必要である．

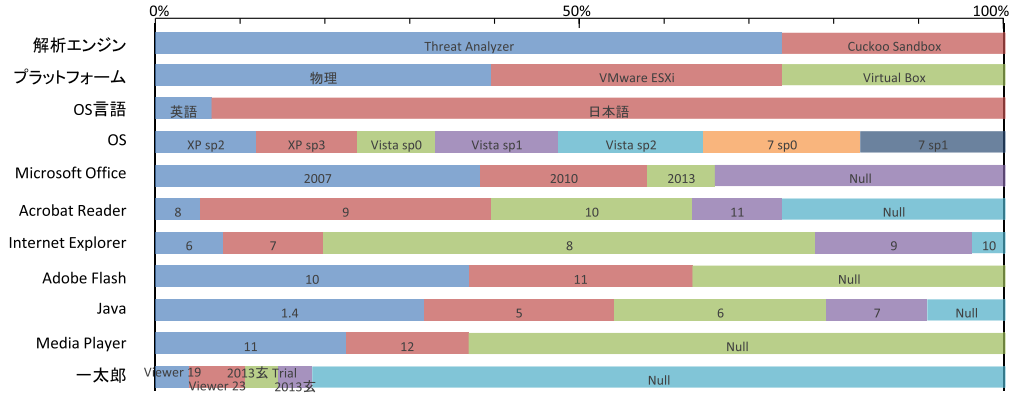


図 6 環境条件の分布

Fig. 6 Ratios of environmental conditions.

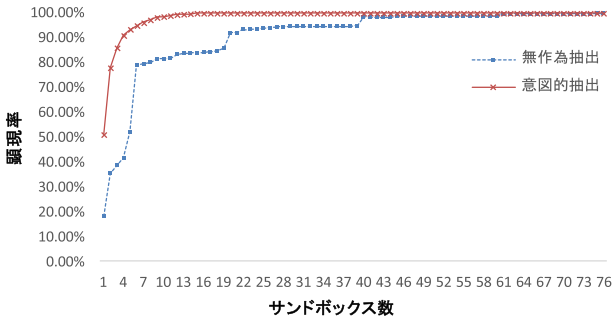


図 7 サンドボックス数と顕現率の関係

Fig. 7 Manifestation rates with respect to number of sandboxes.

上記アソシエーションルールの抽出を成功させるには，第 1 段階として，マルウェアを顕現させる必要がある．このため，限られたリソースで効率的にアソシエーションルールを抽出するには，マルウェアの顕現のしやすいサンドボックスの構成が重要となる．そこでサンドボックスの選定方法と顕現マルウェア数の関係の有無を検証するために，76 種類のサンドボックスから無作為にサンドボックスを抽出して顕現するマルウェア（1 つ以上のサンドボックスで顕現するマルウェア）の数を，抽出するサンドボックスの数を変化させながら検証した．その結果，**図 7** に示すように，無作為抽出では 524 種の顕現マルウェアのすべてを顕現させるのに 75 種類のサンドボックスが必要であることが分かった．一方，顕現マルウェア数の多いサンドボックスから順に選ぶ意図的な選定を行った場合，15 種類のサンドボックスですべての顕現マルウェアを顕現させることができた．これによって，サンドボックスの選定方法が顕現率（全 524 検体に占める顕現するマルウェアの割合）に大きく影響を与えることが確認できた．

3.7 考察

本章では提案する多種環境マルウェア動的解析システムの提案，およびマルウェア解析処理性能ならびに環境選択型マルウェアの顕現条件推定精度の評価を行った．

マルウェア解析処理性能評価では，1 検体あたり 162.6 秒で 76 種類すべてのサンドボックスでの解析処理が完了する結果となった．同じ検体であってもサンドボックスごとの解析時間は一定ではない．このため並列で解析処理を実行してもサンドボックス数が増えることによって本システムの解析処理時間は長くなることが予測される．解析処理のうちマルウェア観測処理は，マルウェアを実行している時間である．この時間を短くすることは，一定時間停止してから不正を行うような時限的な処理を実装したマルウェアの顕現率の低下を招く可能性があるため，安易に短くすることはできない．マルウェア観測処理にかける時間と顕現マルウェア数との関係を検証することでこれらの最適値を求めることが可能だと考える．

環境選択型マルウェアの顕現条件推定精度の評価では，サンドボックスの数，すなわちマルウェアの実行できた数がアソシエーションルールの抽出成功率に影響があることを示した．また，検体の種類（アソシエーションルールのリフト値）によって，この成功率の増加傾向に変化が現れることも確認した．これは「一太郎」や「OS 言語が英語」「Windows XP sp2」等のアソシエーションルールに含まれる環境条件を満たしたサンドボックス数に起因していると考える．つまり，顕現するはずのサンドボックス数の多い検体ほど，アソシエーションルールの抽出成功率が低くなる傾向があるといえる．

またサンドボックスの選定方法により顕現するマルウェア数に影響があることを確認した．本システムにおいてマルウェアが顕現する確率を維持向上させるという観点では，新たな攻撃手法や脆弱性の出現に合わせてサンドボックスの構成を変化させたり，バリエーションを増やしたりするべきである．一方，コスト削減の観点では，サンドボックスの追加がハードウェアやソフトウェアライセンス費用等のコスト増に直結するため，適度なバランスが重要である．よって，攻撃手法や脆弱性に関わる情報に注視して影響を受けやすい環境を追加するとともに，継続的にマルウェアを解析して得られたサンドボックスの顕現状態の

傾向に基づいて冗長なサンドボックスを排除することが必要である．

4. 結論

本論文では，近年巧妙化がますます進むマルウェアに対抗するために，多種環境でマルウェアを同時並列的に解析する M3AS の提案を行い，実在するマルウェア 633 種を解析した．マルウェア特徴抽出機能モジュールにより，全検体のうち 93%からマルウェアの何らかの特徴を検知できることが分かった．また，顕現条件推定機能により，特定の環境でのみ動作（外部ホストへ接続）する環境選択型マルウェアについて，全検体のうち 64%（357 検体）から顕現条件を抽出できることを示した．

M3AS は，構成する全機能をハーフラックに構築している．このため，標的とされている組織や顧客先での解析が可能である．これにより機密情報を含む可能性のある検体を外部に提供することなく，攻撃対象のネットワーク環境で解析可能である．

M3AS によって得られたマルウェアの挙動情報や顕現条件は，人手による解析のための解析環境構築の手がかりや，Firewall やプロキシサーバ等の出口対策の設定情報として与えることでマルウェア感染時の被害発生予防や拡大防止につなげたりすることができる．

また，マルウェアの動作する環境を絞り込めるため，従来は挙動の把握が困難であった環境選択型マルウェアによる脅威に対抗するための将来研究の促進に寄与することを期待する．

一方で，解析あたりのログの量が膨大になるという新たな問題がある．3 章で実施した評価では，1 検体・1 サンドボックスあたりに出力されるログのサイズは Threat Analyzer で平均 14.9 MBytes，Cuckoo Sandbox で平均 10.9 MBytes，全体平均で 13.9 MBytes にもなる（**図 8**）．このため 76 種のサンドボックス数で換算した場合，1 回の解析につき平均で 1 GBytes 以上ものログが生成されることになる．これらのログにはマルウェアの特性とは関係のない API コールログ等が大量に含まれていることから，今後は，これらのログの縮約が課題となる．加えて，インターネットに代理接続してマルウェア配布サーバや C&C サーバと安全に通信する WAN 代理接続機能を開発する．

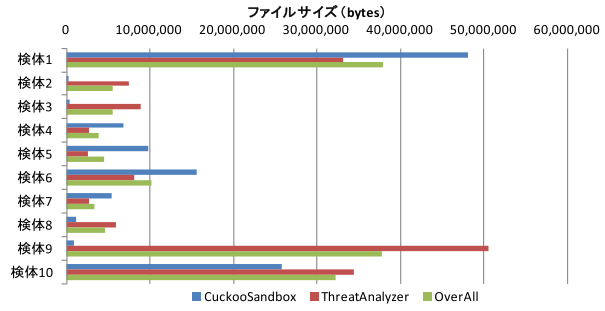


図 8 解析結果のログサイズ
Fig. 8 File sizes of the analysis results.

謝辞 本論文で試作したシステムの評価にあたっては，総務省実証事業「サイバー攻撃解析・防御モデル実践演習の実証実験の請負」および北陸 StarBED 技術センターの協力を得て実施しています．関係者の方々に感謝いたします．

本論文中で使われているシステム・製品・サービス名は，一般に各社の商標または登録商標です．

参考文献

[1] Recruit Marketing Partners Co., Ltd.：企業における情報セキュリティ対策状況，キーマンズネット，入手先 ⟨http://www.keyman.or.jp/at/30004867/⟩（参照 2014-11-24）．
[2] Guardian News and Media Limited or its affiliated companies: Antivirus software is dead, says security expert at Symantec, available from ⟨http://www.theguardian.com/technology/2014/may/06/antivirus-software-fails-catch-attacks-security-expert-symantec⟩ (accessed 2014-11-24).
[3] Solutionary: 2014 NTT Group Global Threat Intelligence Report, Annual Threat Report, available from ⟨http://www.solutionary.com/research/threat-reports/annual-threat-report/ntt-solutionary-global-threat-intelligence-report-2014/⟩ (accessed 2014-11-24).
[4] 新井 悠，岩村 誠，川古谷裕平ほか：アナライジング・マルウェア，pp.42–48, オライリー・ジャパン (2010).
[5] 青木一史，川古谷裕平，岩村誠ほか：半透性仮想インターネットによるマルウェアの動的解析，コンピュータセキュリティシンポジウム 2009 論文集，Vol.2009, pp.1–6 (2009).
[6] Microsoft: Windows Sysinternals, available from ⟨http://technet.microsoft.com/ja-jp/sysinternals/bb545021.aspx⟩ (accessed 2014-11-24).
[7] International Secure Systems Lab.: Anubis – Malware Analysis for Unknown Binaries, available from ⟨http://anubis.iseclab.org/⟩ (accessed 2014-11-24).
[8] ThreatExpert Ltd.: ThreatExpert, available from ⟨http://www.threatexpert.com/⟩ (accessed 2014-11-24).
[9] Claudio “nex” Guarnieri & Cuckoo Sandbox Developers: Automated Malware Analysis – Cuckoo Sandbox, available from ⟨http://www.cuckoosandbox.org/⟩ (accessed 2014-11-24).
[10] ThreatTrack Security Inc., Threat Analyzer, Threat Analyzer Overview, available from ⟨http://www.threattracksecurity.com/enterprise-security/malware-analysis-sandbox-software.aspx⟩ (accessed 2014-11-24).
[11] Palo Alto Networks, Inc., WildFire, PALO ALTO NETWORKS: WildFire Datasheet, available from ⟨https://www.paloaltonetworks.jp/content/dam/paloaltonetworks-com/en_US/assets/pdf/datasheets/wildfire/wildfire-ja.pdf⟩ (accessed 2015-03-24).
[12] 秋山満昭，神薗雅紀，松木隆宏ほか：マルウェア対策のための研究用データセット〜MWS Datasets 2014〜，情報処理学会研究報告コンピュータセキュリティ（CSEC），Vol.2014-CSEC-66, No.19, pp.1–7 (2014).
[13] Rodrigo Rubira Branc: Scientific but Not Academical Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies, Black Hat USA Conference 2012, available from ⟨http://research.dissect.pe/docs/

blackhat2012-presentation.pdf⟩ (accessed 2014-11-24).
[14] Chen, X., Andersen, J., Mao, Z.M., et al.: Towards an Understanding of Anti-Virtualization and Anti-Debugging Behavior in Modern Malware, *The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp.177–186 (2008).
[15] 独立行政法人情報処理推進機構セキュリティセンター：『新しいタイプの攻撃』に関するレポート，IPA テクニカルウォッチ，入手先 ⟨https://www.ipa.go.jp/files/000009366.pdf⟩（参照 2014-11-24）．
[16] 川古谷裕平，岩村 誠，伊藤光恭：ステルスデバッガを利用したマルウェア解析手法の提案，マルウェア対策研究人材育成ワークショップ 2008，Vol.2008, No.8, pp.115–120 (2008).
[17] 柏井祐樹，森井昌克，井上大介ほか：NONSTOP データを用いたマルウェアの時系列分析，コンピュータセキュリティシンポジウム 2013 論文集，Vol.2013, No.4, pp.848–853 (2013).
[18] Emurasoft：今回のハッカーによる攻撃の詳細について，EmEditor ブログ，入手先 ⟨https://jp.emeditor.com/general/今回のハッカーによる攻撃の詳細について/⟩（参照 2014-11-24）．
[19] 株式会社ラック：日本における水飲み場型攻撃に関する注意喚起，入手先 ⟨http://www.lac.co.jp/security/alert/2013/10/09_alert_01.html⟩（参照 2014-11-24）．
[20] Inoue, D., Yoshioka, K., Eto, M., et al.: Malware Behavior Analysis in Isolated Miniature Network for Revealing Malware's Network Activity, *IEE ICC 2008*, pp.1715–1721 (2008).
[21] 山口和晃，堀合啓一，田中英彦：マルウェア解析の効率化手法の検討，情報処理学会，コンピュータセキュリティシンポジウム 2009，pp.925–930 (2009).
[22] Xu, Z., Zhang, J., Gu, G., et al.: GOLDENEYE: Efficiently and Effectively Unveiling Malware's Targeted Environment, *RAID*, Vol.8688, pp.22–45 (2014).
[23] Kirat, D., Vigna, G. and Kruegel, C.: BareCloud: Bare-metal Analysis-based Evasive Malware Detection, *USENIX Security 2014*, pp.287–301 (2014).
[24] Thomas Hungenberg & Matthias Eckert: INetSim Internet Services Simulation Suite, available from ⟨http://www.inetsim.org/index.html⟩ (accessed 2014-11-24).
[25] 独立行政法人情報処理推進機構セキュリティセンター：サイバー攻撃観測記述形式 CybOX 概説，情報セキュリティ，入手先 ⟨http://www.ipa.go.jp/security/vuln/CybOX.html⟩（参照 2014-11-24）．
[26] IPA：脆弱性対策情報データベース，入手先 ⟨http://jvndb.jvn.jp/⟩（参照 2014-11-24）．
[27] VirusTotal – Free Online Virus, Malware and URL Scanner, available from ⟨https://www.virustotal.com/ja/⟩ (accessed 2015-03-24).
[28] TREND MICRO：標的型攻撃に利用される「PlugX」を徹底解析，トレンドマイクロセキュリティブログ，入手先 ⟨http://blog.trendmicro.co.jp/archives/6026⟩（参照 2014-11-24）．
[29] naked security: From the Labs: New PlugX malware variant takes aim at Japan, available from ⟨http://nakedsecurity.sophos.com/2013/12/04/new-plugx-malware-variant-takes-aim-at-japan/⟩ (accessed 2014-11-24).
[30] Common Vulnerabilities and Exposures: CVE-2013-5990, available from ⟨http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5990⟩ (accessed 2014-11-24).
[31] Sudeep Singh: Breaking the Sandbox, available from ⟨http://www.exploit-db.com/wp-content/themes/exploit/docs/34591.pdf⟩ (accessed 2014-11-24).
[32] TRENDMICRO：セキュリティ情報，TROJ_FAKEAV.BME．入手先 ⟨http://about-threats.trendmicro.com/Malware.aspx?language=jp&name=TROJ_FAKEAV.BME⟩．

仲小路 博史 （正会員）

2001 年東京理科大学大学院理工学研究科情報科学専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，研究開発グループシステムイノベーションセンタ）入所．以来，サイバー攻撃対策技術の研究開発に従事．現在，同センタセキュリティ研究部主任研究員．明治大学大学院先端数理科学研究科現象数理学専攻博士後期課程在籍．

重本 倫宏 （正会員）

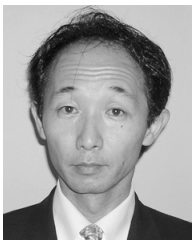
2006 年大阪大学大学院基礎工学研究科システム創成専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，研究開発グループシステムイノベーションセンタ）入所．現在はネットワークセキュリティ技術に関する研究開発に従事．

鬼頭 哲郎 （正会員）

2005 年東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，研究開発グループシステムイノベーションセンタ）に入所．以来，ネットワークセキュリティ技術に関する研究開発に従事．

林 直樹 （正会員）

2007年京都大学大学院情報学研究科数理工学専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，研究開発グループシステムイノベーションセンタ）入所．次世代ネットワーク向け認証連携技術の研究開発に従事．現在はネットワークセキュリティ技術に関する研究開発に従事．



寺田 真敏 （正会員）

1986年千葉大学大学院工学研究科写真工学専攻修士課程修了．同年（株）日立製作所入社．博士（工学）．現在，研究開発グループシステムイノベーションセンタにてネットワークセキュリティの研究に従事．2004年からHitachi Incident Response Team チーフコーディネーションデザイナー．2004年4月からJPCERTコーディネーションセンター専門委員．2004年4月から2007年まで中央大学研究開発機構客員研究員．2004年8月から情報処理推進機構セキュリティセンター研究員．2008年から中央大学大学院客員講師を兼務．

菊池 浩明 （フェロー）

1988年明治大学工学部電子通信工学科卒業．1990年同大学院博士前期課程修了．1994年同博士（工学）．1990年（株）富士通研究所入社．1994年東海大学工学部電気工学科助手．1995年同専任講師．1999年同助教授，2000年同電子情報学部情報メディア学科助教授，2006年同情報理工学部情報メディア学科教授．2008年同情報通信学部通信ネットワーク工学科教授．1997年カーネギーメロン大学計算機科学学部客員研究員．2013年明治大学総合数理学部先端メディアサイエンス学科教授．WIDEプロジェクト暗号メールシステムFJPEMの開発，認証実用化実験協議会（ICAT），IPA独創情報技術育成事業等に従事．1990年日本ファジィ学会奨励賞，1993年情報処理学会奨励賞，1996年SCIS論文賞，2010年情報処理学会JIP Outstanding Paper Award．2013年IEEE AINA Best Paper Award．2014年情報セキュリティ文化賞．電子情報通信学会，日本知能情報ファジィ学会，IEEE，ACM各会員．