明治大学大学院先端数理科学研究科

２０１６年度

博士学位請求論文

# 環境選択型マルウェアの挙動解析による自動対策システムの研究

# A Study on Automated Defense System with Behavior Analysis Against Environment-Sensitive Malware.

学位請求者　現象数理学専攻

仲小路 博史

# 1. 研究背景と目的

## 1.1. 研究の背景

2009 年頃より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．そのサイバー攻撃の目的は，攻撃者による自己顕示欲の誇示から金銭搾取や政治的活動（ハクティビズム），諜報活動（サイバーエスピオナージ）に変化している．これに伴い，犯行に加担する主体も単独から組織，あるいは水平分業化した連合体へと変化してきている．また，攻撃に用いられる手法もゼロデイ攻撃[1]や水飲み場攻撃[2]等に巧妙化しており，金融機関や政府機関，制御システム等の重要インフラを狙った標的型攻撃が多発している．

2010 年，イラン中部ナタンズにあるウラン濃縮核関連施設の狙った攻撃は，Stuxnet[1]と呼ばれる非常に高度なマルウェア[3]が攻撃に用いられ，同施設の遠心分離機の回転数を不正に制御して 8 千機以上を機能不全にさせた．これによってイランの核開発は二年程度退化させられたとも報じられている．これらの一連の行為はイランに敵対する国家が主体的に関わっているとも言われており，強い動機のもと，プロフェッショナルによる計画的な攻撃が現実のものとなってきている[2]．また、2015 年 6 月には日本年金機構の年金情報管理システムへの攻撃により 125 万件もの個人情報が漏えいし，社会的な問題となった[3]．本サイバー攻撃では既存のウイルス対策ソフトでは当時検知ができなかった未知のマルウェアが用いられており，このマルウェアはごく自然な日本語で記述された標的型メールに添付される形で同機構のネットワークに侵入した．また，世界的に見ても，このマルウェアの発見報告は日本国内に集中しており，日本を狙った標的型メールであったことは疑う余地もない．職員がこの標的型メールを開封し，このマルウェアに感染したことで，感染した端末は攻撃者により遠隔から操作が可能になり，氏名，基礎年金番号，生年月日等の個人情報が漏えいした．同機構は本事件の電話対応窓口を設けたが，開設初日に 10 万件以上の苦情・問い合わせがあり，サイバー攻撃による社会への影響の甚大さを改めて認識する契機となった．

前述した 2 つの事例で用いられたマルウェアは，標的とする組織でしか動作しないようにマルウェアに細工が施されていた．これが従来の手法による対策を困難とさせ，甚大な被害をもたらした 1 つの原因と言われている．

被害の甚大化を軽減させるためには，新たな脆弱性や脅威が発覚した場合の早期対策が重要となる．2014 年 9 月に bash シェルの脆弱性（通称 Shellshock）[4]が公開された．この脆弱性は Linux 等の UNIX 系 OS に標準搭載されているである bash シェルに存在するソフトウェア上の

---

[1] ソフトウェアやハードウェアに脆弱性が発見され，情報が広く世の中に公開される前に，その脆弱性を悪用して行われる攻撃

[2] 特定の個人や組織を狙った攻撃の一種で，Web サイトを改ざんする等して不正なコードを挿入し，その改ざんされた Web サイトに狙った個人や組織が訪れた時にのみマルウェア等の不正なプログラム，コードをダウンロードさせて行う攻撃

[3] コンピュータウイルス，ワーム等の不正な活動を行うプログラムの総称（Malicious Software の略）

不具合が攻略されると攻撃者により任意のコマンドを実行されてしまう重大な問題を抱えていた．この脆弱性の影響を受ける Web サーバの割合は、脆弱性公開当初、世の中の Web サーバのうちの 10%～20%もあったと推定[5]されている．この脆弱性を有するサーバを探索するスキャン行為が脆弱性公開の翌日には観測され，さらに次の日には、その脆弱性を攻撃する通信が観測されたと報告[6]されており，本事例は脆弱性の影響および早期対策の重要さを世に知らしめる契機となった．

ベライゾン社の公表しているデータ漏えい／侵害調査報告書[7]によると，最初の攻撃から最初の侵害までを数分内で完了させる事例が全体の 85%を占めており，さらに、データの漏えいまでを数分内で完了させる事例が 46%と報告されている．

## 1.2. 現状の問題

現状の標的型対策の対処には 2 つの問題がある．

**問題１：マルウェア解析の困難性**

標的型攻撃がサイバー攻撃の主流となる前の 2008 年頃は，ウイルス対策ソフトの導入により殆どのマルウェアを検知，駆除できていた．このためウイルス対策ソフトの導入およびパターンファイルを最新状態へアップデートすることにより，上述したようなマルウェアを悪用した攻撃による実害の発生を未然に防ぐことができた．ところが 2010 年以降，パターンファイルによるウイルス対策ソフトの検知率が低下しはじめ[8]，ウイルス対策ソフトの最大手である Symantec 社の上級副社長を務める Brian Dye や，NTT Group の Solutionary 社の研究グループらも，新種のマルウェアの半数以上が既存のウイルス対策ソフトでは検知できない，と報告し，ウイルス対策ソフトのみによる対策の終焉を認めている[9][10].

このような状況の中，企業の情報セキュリティに関わる事件・事故の原因のトップはクライアント PC のマルウェア感染によるもの、とも報告[11]されており，マルウェアによる脅威は依然高い状態にある．このため，侵入を防ぐことを目的とした従来のウイルス対策ソフトの導入等の入口対策に加えて，侵入後のマルウェアの活動を検知・防御する内部対策，外部への情報の流出や遠隔操作等を防止する出口対策とを多段に組み合わせた多層防御による「侵入を前提」としたシステム全体の防御の概念が普及しつつある．また，CSIRT（Computer Security Incident Response Team）のようなインシデント発生時の対応体制の整備も進められている．我々守る側は、マルウェアが組織の中に侵入してしまった場合に備え，技術や人を駆使して侵入したマルウェアの特性を解析し，早急な被害拡大防止策等の他の対策に繋げていくことが重要となる．

マルウェアの特性を解析するには，マルウェアを実際の解析環境で動作させてその挙動を解明する動的解析（後述）が有効とされている。ところが，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知して動作を停止する耐解析機能[12][13]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知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[14]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．川古谷らの論文[15]では Sysinternals[16]の利用を検知して動作を止める検体について報告されている．また，攻撃者が用意したマルウェア配布サーバから第二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]も存在し，インターネットから隔離された解析環境では解析が困難になるほか，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].

そのため，従来の動的解析技術では，それらのマルウェアの特性が解明できないという新たな問題が起きてしまっている.

**問題２：誤対策による業務悪影響懸念**

攻撃者らは最新技術を取り込んだマルウェアの開発や,メールアドレス等の攻撃先情報の収集,マルウェアの送信等のタスクを水平分業化し，連携して高度な攻撃を仕掛けてくる．対する我々守る側も単独組織による対策ではなく,組織間でインテリジェンス[4]を共有して攻撃に備える集団防御の概念が求められている．この集団防御を実現するべく，ISAC（Information Sharing and Analysis Center）[20][21]のような公益法人が業界毎にインテリジェンスの共有を進めてリスクの軽減を図っている．また公益法人だけではなく，FireEye[22]，Threat Connect[23]等の民間企業がインテリジェンス共有サービスを開始している．ところが，インテリジェンス共有の仕組みは整いつつあるもののインテリジェンスを活用した有効な対策を殆ど実施できていない．一部ではあるが，世の中で共有されているマルウェアの解析結果を利用してマルウェアによるインターネット接続を即座に遮断する技術も実用化されている．しかし，マルウェアの中には例えば検索エンジン等の正規のサイトにアクセスする種も存在するため，本技術を適用すると検索エンジンが使えなくなるなど，不確実なインテリジェンスの活用が結果的に誤った対策を講じるケースもある．これによって業務に悪影響を及ぼすことになるため，マルウェアの活動（不正なアクセス等）への対策に躊躇してしまうケースも多く，対策が後手に回る問題があった.

例えば，前述した日本年金機構への標的型攻撃のケースでは，攻撃手段として EMDIVI[24]と呼ばれるマルウェアが用いられていた．このマルウェアの解析情報や対処方法を他の組織に共有し，対策につなげられていれば，東京商工会議所や早稲田大学等，計 44 もの組織が短期間に同様の手法によって受けた被害[25]を未然に防げていた可能性は高い.

[4] マルウェアの解析結果等の脅威情報やそれらへの対処支援情報（Intelligence）

## 1.3. 従来手法と課題

問題１に対して，従来から不審メールなどに添付されるようなファイル（検体）がマルウェアか否か，あるいはマルウェアとしてどのような機能を有するかを解析する方法には，検体をVirusTotal等のマルウェア検査サービスで検査したり検体から文字列を抽出して不審な文字を読み解いたりする表層解析手法[26]と，検体を特殊な解析環境で実行して，その振る舞いを観測する動的解析手法[27]と，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[28]とが使われていた．通常，マルウェア解析を行う情報システム部門やSOC[5]，CSIRTのアナリストは１つの検体に費やせる解析時間や人員などのリソースが限られていることから，検体を解析する際は検体の性質，解析の目的，解析者のスキルセットや経験則に応じて、図 1.1に示すように，それぞれの手法を補完的に組み合わせて検体の挙動を解明して予防および対策に繋げるのが一般的である．

表層解析手法は，解析したい検体のファイルのハッシュ値[6]から過去の解析結果を調査したり，ウイルス対策ソフト等を利用してパターンファイルからマルウェアか否か，あるいはマルウェアの種類を判別したり，検体に含まれるテキスト列を見て具備する機能（使用する関数）を確認したりすることで危険性を判断する．本手法は低コストで解析できる反面，未知のマルウェアや難読化されたマルウェアに対しては無力であるケースが多い．このため，後述する動的解析や静的解析の要否や，これら解析の手段を決定する参考情報として表層解析の結果が用いられる．

静的解析手法は，検体をアセンブラコード等にリバースエンジニアリングして解析する．検体の具備する機能の全てを詳細に解明できる利点があるが，セキュリティのみならず，プログラムやOS，ハードウェア等の仕組みに関する深い知識と，コードを一行ずつ読み解くための膨大なコストが必要となる．一方で高度なマルウェアは，その挙動の解明を逃れるために，コードを読みにくくする「難読化」が施されているケースが多い．この場合，この難読化を解いて読みやすいコードに直す作業も必要となり，さらにコストがかかってしまうことが問題となっている．このような性質もあって，静的解析はマルウェアの挙動を解明する最終手段として用いられることが多い．

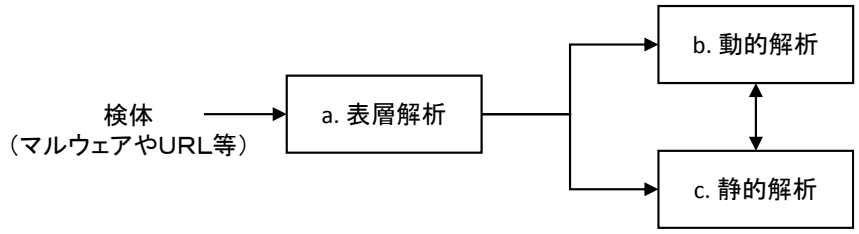



図 1.1　解析手法と関係性

[5] システムのセキュリティ運用を担当する部署（Security Operation Center）
[6] データから所定の計算手順により算出された固定長の値のことを言い，データに少しでも変化があればハッシュ値は全く異なる値となる．2つのデータの同一性を比較する際に，それぞれのハッシュ値同士を比較することで，データ全体を比較せずに処理することが可能となる．

動的解析手法は，難読化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低い点，静的解析手法だけでは分からない挙動（例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(例えば 1 月 1 日の正午にシステムファイルを削除する挙動等)はその挙動を把握できないという欠点もある．

動的解析にあたっては，マルウェア解析者が Sysinternals 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と，マルウェアの挙動観測および解析をシステム上で自動的に行う動的解析サービス，動的解析ソフトウェアを利用する方法，とがある．動的解析サービスとしては Anubis[29]や，ThreatExpert[30]が，動的解析ソフトウェアとしては Cuckoo Sandbox[31]や Threat Analyzer[32]が提供されている．本ソフトウェアは解析用に用意したサンドボックスという実際の実行環境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詳細に取得する．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 に含まれる FFRI Dataset 2014[33]も同ツールによって取得したデータが含まれる．

以上の従来手法を表 1.1 に整理する．

表 1.1　解析手法の特徴

| | 表層解析 | 動的解析 | 静的解析 |
|---|---|---|---|
| 代表例 | 中津留ら[26] | 青木ら[27] | 新井ら[28] |
| 所要時間 | 瞬時 | 数分～数時間 | 数時間～数ヶ月 |
| 必要スキル | 低い | 高い | 非常に高い |
| 長所 | ・低コストで瞬時に解析が可能<br>・オフラインで解析が可能（安全） | ・静的解析と比較して短期間で解析が可能<br>・サーバとの通信等, 検体外部と連携して初めて顕現する挙動も解析可能<br>・難読化された検体も解析可 | ・検体の備える全ての機能を把握可能<br>・オフラインで解析が可能（安全） |
| 欠点 | ・難読化等がされた高度なマルウェアには効果が低い<br>・未知マルウェアの解析が不得手 | ・仮想環境検知など動的解析を逃れるマルウェアの解析が困難<br>・インターネット隔離環境では解析中に顕現しない機能の把握が困難 | ・非常に高いスキルと時間が必要<br>・難読化された検体の解析が困難<br>・サーバとの通信等, 検体外部と連携して初めて顕現する挙動の解析が不可 |

標的型攻撃で用いられるマルウェアは，遠隔から攻撃者の指示を受けたり，新たなマルウェアをインターネットからダウンロード，実行したりして不正活動を行うことが多い．この場合，攻撃者の指示内容や，新たなマルウェアの種類によって不正活動の内容も変化し，取るべき対策も変わってくる．このようなマルウェアは，たとえ静的解析で詳細に解析を行ったとしても，攻撃者の指示内容や，（ダウンロードできていない）新たなマルウェアの機能の解明は不可能である。このため，上記指示内容や機能をある程度自動で解明できる動的解析手法の注目が高まっている．

動的解析には表 1.2 に示す従来手法がある．

表 1.2　動的解析の従来手法

| | 既存手法 | | |
|---|---|---|---|
| | 手動対処 | FireEye 社<br>製品[34] | 長谷川ら<br>提案手法[35] |
| 環境選択型マルウェアの解析 | ○<br>時間さえかければ詳細な解析が可能 | △<br>サンドボックスは決められたパターンから選択可能 | －<br>既存手法を活用 |
| リスク軽減 | △<br>人手の対策には時間が必要<br>確実な情報に基づく対策のみ実施するため，対策できないケースが発生 | ◎<br>不審な通信を遮断 | ○<br>不審な通信をしているセグメント等を局所的に遮断 |
| 副作用軽減 | ◎<br>確実な情報に基づく対策のみ実施するため，副作用の発生は稀 | ×<br>遮断することから，業務に影響が出やすい | △<br>対策による影響範囲を考慮<br>ただし誤った情報（正規通信先）が含まれる場合)、一部ユーザの業務を阻害 |
| 速度 | ×<br>全ての作業が人手のため一時間以上必要 | ○<br>数秒 | △<br>対策候補を管理者に選ばせる必要有 |

手動対処は多くの情報システム部門や SOC/CSIRT 部門に属するセキュリティ専門家によって行われている手法である．マルウェアによるセキュリティ侵害の兆候をつかむと，表層解析，動的解析，静的解析の手法を組み合わせながら，不審なファイルをマルウェアであるか否か，あるいはマルウェア感染によってもたらされる影響を特定する．環境選択型マルウェア等の高度なマルウェアの解析には動的解析環境が複数必要だったり，繰り返しの解析が必要だったりと，作業工数は膨大になり，限られたリソースの中で解析するには限界がある．セキュリティ専門家は解析の結果を用いて端末の切り離しや，不正なサイトとの通信遮断等の対策をとる．この際に，業務上，誰かが必要とする可能性のあるサイトとの通信を遮断するようなことは可能な限り避ける必要があるため，確実に不正であると判明しているサイトしか対策（遮断）しない、すなわち守る対象のシステムの可用性を重要視することが現場では多い．このように人の判断を介して熟考された対策がされるため，業務への副作用は発生しにくい．その反面，可用性を重視したがために，不確実な事象への対策がなされない、あるいは後手に回るためリスクが残存することも多い．また，増加する脅威に対して，高度なセキュリティ知識を有する人材の確保も一層困難になってきている点，人によって判断基準が異なるために画一的な対策が取れない点，業務時間外の対応が困難な点も問題となっている．

FireEye 社の FireEye NX[34]は，動的解析型のネットワークセキュリティ製品である．本製品は STAP（Specialized Threat Analysis and Protection）市場においてシェア 37.9%と，圧倒的なシェアを誇っている[35]．本製品はインラインに配置することによってマルウェアの解析結果を用いて通信の遮断を即時に行える機能を有する．また本製品の動的解析実行環境（サンドボックス）は OS や Office 系のソフトウェアを変更可能であり，いくつかのパターンから選択する．さらにリアルタイム検知を謳っており，独自仮想化技術によって複数のサンドボックスで並列解析，高速検知を実現している．しかしながら，仮想化技術を用いるため，仮想環境を検知するマルウェアに対して精度が落ちる場合がある．さらに，マルウェアの解析時にノイズが紛れ込んだ場合に，そのノイズに反応して誤った対処をしてしまい，本製品を導入した組織の業務を止めてしまう可能性がある．極端な例では，意味もなく銀行のオンラインバンクにアクセスするマルウェアを作成して本製品導入組織にメールで送りつけることによって，本製品はそのマルウェアを自動解析して得たオンラインバンクサイトへの接続を遮断してしまう．これによって，本製品導入組織からはオンラインバンクの利用ができなくなり，決済などに影響が出てしまう．このような攻撃が成立してしまう点で問題があった．

長谷川らの提案する手法[36]は，管理対象のネットワーク内でマルウェア感染等のインシデントが発生した際に，迅速かつ業務への悪影響軽減を考慮した適切なインシデント対応を行うためのインシデント対応支援システムを提案している．本手法では、マルウェア解析機能自体は既存の手法の活用を想定している．マルウェアの感染を検知すると，対策の有効性と対策による影響を独自の指標により評価して対策を立案，提示する．対策手段は遮断する対象の範囲を変化させた 9 パターンが提示される．これにより遮断する範囲をインシデントの発生している箇所に限定

することで，業務への悪影響を最小限に抑えている．しかしながら，誤った解析結果等による誤った対策による業務への悪影響は発生してしまう．また，システムによって提示された対策を，人を介して選ぶことになるため，即時の対策は難しい．たとえば意思決定者が勤務していない時間帯に発生したインシデントへの対応は遅れてしまう．

問題２に対し，多くの組織で，情報システム部門，あるいは SOC/CSIRT に属するセキュリティ専門家が，発生したインシデントに手動で対応してきた．ところが高度化，頻発化，迅速化するサイバー攻撃に対しては，今いる専門家の人手によるセキュリティ対策では限界があった．この状況においてセキュリティ対策の迅速対処にむけた施策がいくつか進められている．

その一つが対策プロセスの自動化である．セキュリティ対策に膨大な時間と労力などのリソースを費やしたり，ルールの複雑化による管理負荷が増大したりするなか，対応する人が専門家であったとしても手作業の対策（設定変更など）では，設定ミスや、実施者のスキルレベル，判断基準の違い等による判断差異が発生するため、統制の取れた均一的なセキュリティ対策が困難な状況となってきている．そのため，セキュリティ対策に関わる作業を自動化することで，効率性や有効性の向上を図る取り組みが活発化している．例えばアメリカ国立標準技術研究所（NIST）では、セキュリティ対策の自動化を目指した標準規格 SCAP[37]の開発が進められている．SCAP は，システムの脆弱性対策を自動化するための脆弱性識別子の記述規則を定めた CVE[38]や，その脆弱性の危険性を定量的に表記する CVSS[39]，脆弱性の有無や設定を確認するための OVAL[40]等の標準仕様から構成されており，セキュリティ対策機器の相互運用の素地が整いつつある．こういった後押しもあって，セキュリティベンダも、本仕様に準拠したセキュリティ対策機器を開発することにより機器間で連携した脆弱性の発見，攻撃の検知から対策までの一連のセキュリティ対策の自動化を目指している．事実，これらの標準規格に準拠した製品が出荷されており連携が可能となった．しかし，SCAP に準拠した情報でも，その内容の真正性の保証はなされていない。このため，不確実な情報に基づく自動化は副作用が懸念されることから、導入，実運用が進んでいないのが実情である．特に可用性を重視する制御システム等では，セキュリティ対策技術の導入やその誤作動により生じる副作用が人命に関わる事態へとつながる可能性があるため導入には慎重にならざるをえない状況にある[36]．このようにセキュリティ対策によるリスク低減と，業務を考慮した可用性維持の両面を考慮したセキュリティ対策が必要となるが，セキュリティ対策による副作用はほとんど議論されていない状況にあり，実用的な自動対策は実現できていない問題が残っている．

## 1.4. 研究の目的

本論文では，高度なマルウェアを利用した標的型攻撃に対する早期対策の重要性に鑑み，自動対策システム「自律進化型防御システム（AED : Autonomous Evolution of Defense）」を提案し，環境選択型マルウェアへの迅速かつ自動の対策を実現する．このため 1.2 節で提起した２つの問題に対し，以下を目的とする．

**目的１**

特性の解析が困難になってきている環境選択型マルウェアの増加という問題１に対して，環境選択型マルウェアを自動解析し，攻撃者による遠隔操作や情報搾取を防止するために有用となる“マルウェアの接続先（不審サイト）”を抽出すること．

**目的２**

抽出した情報には不確実な情報も含まれているため，そのまま対策（遮断）してしまうと，ユーザの業務に悪影響を与えてしまう問題２に対して，マルウェアによる被害発生リスクの軽減と誤った情報を用いて実施した自動対策によるユーザへの業務悪影響軽減を両立．

## 1.5. 研究の新規性

本論文で提案する AED は，サーバとの通信のやりとり等，外部と連携して初めて攻撃者の意図した動作したり、特定のアプリケーションの存在に依存したりするマルウェアの挙動をも解析可能な動的解析技術を確立し，マルウェア本来の挙動を自動的に解明する（目的１）．解明して得られた結果から，マルウェアの外部接続先を抽出し，その情報を出口対策に活用する。この際に誤った情報による対策であっても、業務への悪影響を軽減する措置を取り入れることで、悪影響を懸念する管理者の意思決定を介さない、つまり迅速かつ自動的に実施可能な対策技術を確立する（目的２）．

目的１を達成するにあたって，表 1.1 下線部に示したとおり，以下の課題がある．

課題１：動的解析を逃れるマルウェアの解析
課題２：インターネット隔離環境では顕現しない機能の把握

そこで課題１を解決するために，特定の環境でしか動作しないマルウェアを，選択型マルウェアの活動しやすい 76 種のサンドボックスを構成し，動的解析を逃れるマルウェアを動作させて挙動を観測しマルウェアの接続先（不審サイト）の抽出，および依存する環境を特定するマルウェア挙動解明技術の基本方式を提案する[実績 3, 4, 14]．また，サンドボックスの数が本システムのコストに直結することから，2,000 を超えるマルウェアの解析実績から最適なサンドボックス構成を導出する手法を提案する[実績 10]．

課題２を解決するために，解析中のマルウェアを安全にインターネット上のサーバと通信させ，可能な限り解析中にマルウェアの具備する機能を顕現（本来備えている機能が作動）させるマル

ウェア通信制御技術を提案する[実績 1, 11].

これらの技術によって環境選択型マルウェアから抽出したマルウェアの接続先情報を対策（遮断）すべき URL として自動取得する．しかし，この URL には正規のサイトも含まれる可能性があり，その情報を用いて機械的に対策（遮断）してしまうと 1.2 節で提起した問題２が発生する．このため目的２を達成するには，以下を課題とする．

課題３：マルウェアによる不正なアクセスの遮断
課題４：誤った情報による対策であったとしても業務上必要なアクセスは許可

そこで，前述したマルウェアの自動解析によって得られた対策すべき URL が不確実な情報であっても，その URL に対してユーザが Web アクセスする際に CAPTCHA 認証を追加する．CAPTCHA は機械と人とを判別する逆チューリングテストであり，マルウェアのようなプログラム（機械）では CAPTCHA を解読することができない特性を利用する．これによって，マルウェアによる不正なアクセスを遮断する．また，対策すべき URL と，それらの URL へユーザがアクセスする際の CAPTCHA 認証の成否等の結果を機械学習することにより，前述したマルウェア解析によって得られていない未知の URL に対しても高精度で CAPTCHA 認証を出すべき URL か否かを判定してマルウェアによる通信を遮断する方式を実現する[実績 9]．さらに，CAPTCHA 認証の成否の結果を用いて，マルウェア解析結果によって得られた不確実な情報を，遮断すべき URL，あるいはそのまま許可すべき URL か、を判定して確実性の高い情報へ良質化する技術を提案する[実績 7].

これらの提案を通じて，不確実情報に基づく自動対策による副作用によって発生していたユーザの業務への悪影響を、極力軽減した自動対策手法を実現する．

## 1.6. 位置づけ

本節では，本件で提案する AED の位置づけを表 1.3 を用いて述べる．

比較対象として，現状多くの現場（SOC や CSIRT）で行われている手動対処と，マルウェア解析・対策製品の業界トップレベルのシェア[41]を有する FireEye 社の FireEye NX と，業務への悪影響を考慮したセキュリティ対策支援技術を提案した長谷川らの手法とを用いて本論文で提案する AED の特徴を述べる．

表 1.3　研究の位置づけ

| | 提案手法<br>AED | 既存手法 | | |
|---|---|---|---|---|
| | | 手動対処 | FireEye 社<br>製品 | 長谷川ら<br>提案手法 |
| 環境選択型マルウェアの解析 | ◎<br>環境選択型マルウェアの解析に対応 | ○<br>時間さえかければ詳細な解析が可能 | △<br>サンドボックスは決められたパターンから選択可能 | －<br>既存手法を活用 |
| リスク軽減 | ○<br>不審な通信を追加認証により一時遮断<br>ユーザの判断で対策キャンセル可能 | △<br>人手の対策には時間が必要<br>確実な情報に基づく対策のみ実施するため，対策できないケースが発生 | ◎<br>不審な通信を遮断 | ○<br>不審な通信をしているセグメント等を局所的に遮断 |
| 副作用軽減 | ○<br>追加認証により業務阻害防止<br>グレーリストの自動学習による認証頻度最適化 | ◎<br>確実な情報に基づく対策のみ実施するため，副作用の発生は稀 | ×<br>遮断することから，業務に影響が出やすい | △<br>対策による影響範囲を考慮<br>ただし誤った情報（正規通信先）が含まれる場合)、一部ユーザの業務を阻害 |
| 速度 | △<br>数分 | ×<br>全ての作業が人手のため一時間以上 | ○<br>数秒 | △<br>対策候補を管理者に選ばせる必要有 |

本論文の提案手法は，76 種の異なる環境を持つサンドボックスから構成される．サンドボックスには物理構成を持つ環境や，日本固有のアプリケーションへも対応でき，環境選択型マルウェアによるインターネット接続先を高精度で抽出できる点で優位である．また，環境選択型マルウェアの実行結果から，依存する環境を推定する機能も FireEye にはない．

上記，マルウェア解析から得た結果には，マルウェアのインターネット接続先 URL に関する情報も含まれるが，その情報の中には正規な URL も含まれるため，そのまま当該 URL への接続を自動的に禁止してしまうと，誤対策につながり本来業務へ悪影響を与えてしまう懸念があった．従来技術は，マルウェア等による不正サイトへと接続リスクを軽減するか，業務への影響を軽減するか，どちらかに偏っていた．これに対し，本提案手法は，不審なサイトへのアクセスに際し，ユーザの意思を確認することによりユーザにとって業務上必要なアクセスを許可し，マルウェアによる接続を遮断するとともに，それらを学習して業務への影響を最小限に抑える．またシステム管理者による意思決定プロセスを挟まないため，長谷川らの提案より即時な対策が取れる．環境選択型マルウェアによるリスク軽減と自動対策による業務への悪影響の軽減を両立する点で新規である．

## 1.7. 論文構成

論文の構成を図 1.2 に示す.

環境選択型マルウェアによる脅威に自動対策するために，本論文はマルウェアの挙動を解明する技術と，解明した挙動に基づいて自動対策する技術の 2 つによって構成される.

2 章では，環境選択型マルウェアの挙動を自動的に解明して，マルウェアの通信先を抽出する技術[実績 3, 4, 14]について述べる．3 章では解析中にマルウェアがインターネットにアクセスする際の安全性を確保する技術[実績 1, 11]について述べる. 4 章では世の中の普及している対策（認証付きプロキシ）をすり抜ける高度なマルウェアを解析可能とする技術[実績 12]について述べる. 5 章は，2 章で提案したシステムを社会実装するにあたり弊害となるコスト増大の問題を解決する手法[実績 10]について述べる．6 章では解析対象をマルウェア（ファイル）から URL（Web サイト）へ拡張し，対処可能なアタックベクタの拡大を図る技術[実績 6]について述べる.

これまで述べた技術により抽出したマルウェアの解析結果を用いて，ユーザの本来業務への悪影響を最低限に抑えた自動対策手法[実績 7, 9]について 7 章で述べる.

8 章では，本論文をまとめる.

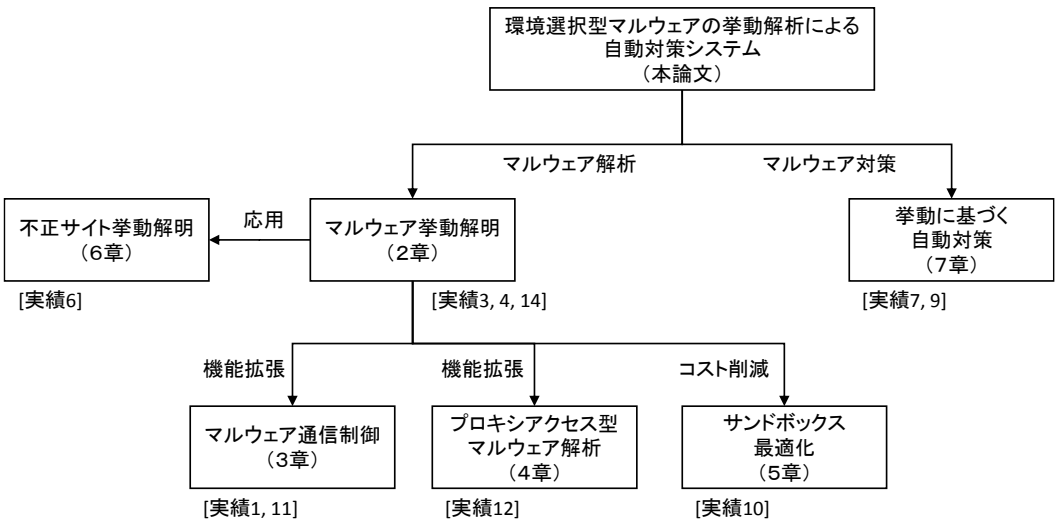



図 1.2　本論文の構成と各章の関係

## 参考文献


[1] 小熊信孝：Stuxnet ー 制御システムを狙った初のマルウエア ー，CERT/CC，入手先＜http://www.jpcert.or.jp/ics/2011/20110210-oguma.pdf＞(参照 2016-10-28)

[2] Ellen Nakashima，Joby Warrick：Stuxnet was work of U.S. and Israeli experts, officials say，The Washington Post，入手先＜https://www.washingtonpost.com/world/national-security/stuxnet-was-work-of-us-and-israeli-experts-officials-say/2012/06/01/gJQAlnEy6U_story.html＞(参照 2016-10-28)

[3] サイバーセキュリティ戦略本部：日本年金機構における個人情報流出事案に関する原因究明調査結果，入手先<http://www.nisc.go.jp/active/kihon/pdf/incident_report.pdf>（参照 2015-12-7）

[4] Common Vulnerabilities and Exposures：CVE-2014-6271，MITRE，入手先＜https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271＞(参照 2016-10-28)

[5] THE ZERO ONE：Shellshock は脆弱性対応の好教材，入手先＜https://the01.jp/p000172/＞(参照 2016-10-28)

[6] @police：Bash の脆弱性を標的としたアクセスの観測について（第 2 報），警察庁，入手先＜https://www.npa.go.jp/cyberpolice/detect/pdf/20141007.pdf＞

[7] Verizon Enterprise Solutions：2012 年度データ漏洩/侵害調査報告書，入手先＜http://www.verizonenterprise.com/resources/reports/rp_2012_Data_Breach_Investigations_Report_ja_xg.pdf＞(参照 2016-10-28)

[8] NTT セキュリティ・ジャパン株式会社：サイバー攻撃と脅威分析に関する現場からの報告，入手先＜http://www.ntt.com/content/dam/nttcom/hq/jp/business/go-event/pdf/s/1-7.pdf＞(参照 2016-10-28)

[9] Guardian News and Media Limited or its affiliated companies：Antivirus software is dead, says security expert at Symantec，入手先< http://www.theguardian.com/technology/2014/may/06/antivirus-software-fails-catch-attacks-security-expert-symantec>(参照 2014-11-24)

[10] Solutionary： 2014 NTT Group Global Threat Intelligence Report, Annual Threat Report，入手先<http://www.solutionary.com/research/threat-reports/annual-threat-report/ntt-solutionary-global-threat-intelligence-report-2014/>(参照 2014-11-24)

[11] 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)

[12] Rodrigo Rubira Branc：Scientific but Not Academical Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies, Black Hat USA Conference 2012，入手先<http://research.dissect.pe/docs/blackhat2012-presentation.pdf>(参照 2014-11-24)

[13] 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

[14] 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）

[15] 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，pp115-120，2008

[16] Microsoft： Windows Sysinternals，
入手先<http://technet.microsoft.com/ja-jp/sysinternals/bb545021.aspx>(参照 2014-11-24)

[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] Telecom-ISAC JAPAN，入手先< https://www.telecom-isac.jp/>（参照 2015-12-7）

[21] 一般社団法人金融 ISAC，入手先< http://www.f-isac.jp/>（参照 2015-12-7）

[22] FireEye：FireEye Threat Intelligence，入手先< https://www.fireeye.jp/content/dam/fireeye-www/regional/ja_JP/products/pdfs/ds-threat-intelligence.pdf >（参照 2015-12-7）

[23] THREATCONNECT,INC.：Enterprise Threat Intelligence Platform，
入手先<https://www.threatconnect.com/>（参照 2015-12-7）

[24] Symantec Corporation：Backdoor.Emdivi，入手先< https://www.symantec.com/security_response/writeup.jsp?docid=2014-101715-1341-99>（参照 2015-12-7）

[25] 久保 啓司，標的型攻撃への対応，一般社団法人 JPCERT コーディネーションセンターインシデントレスポンスグループ，入手先< https://www.jpcert.or.jp/present/2015/JNSAWG20150630-apt.pdf>（参照 2015-12-7）

[26] You Nakatsuru：Understanding Malware. Security Camp 2015, JPCERT/CC Analysis Center, 入手先 https://www.jpcert.or.jp/present/2015/seccamp2015_13D_14D_understanding_mmalwar.pdf(参照 2016-10-24)

[27] 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).

[28] 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，オライリー・ジャパン，pp.42-48(2010)

[29] International Secure Systems Lab：Anubis - Malware Analysis for Unknown Binaries,

入手先<http://anubis.iseclab.org/>(参照 2014-11-24)
[30] ThreatExpert Ltd. : ThreatExpert, 入手先<http://www.threatexpert.com/>(参照 2014-11-24)
[31] Claudio “nex” Guarnieri & Cuckoo Sandbox Developers : Automated Malware Analysis - Cuckoo Sandbox, 入手先<http://www.cuckoosandbox.org/>(参照 2014-11-24)
[32] ThreatTrack Security Inc. : Threat Analyzer, Threat Analyzer Overview, 入手先<http://www.threattracksecurity.com/enterprise-security/malware-analysis-sandbox-software.aspx>(参照 2014-11-24)
[33] 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).
[34] FireEye : Network Security Essentials，入手先＜https://www.fireeye.jp/products/nx-network-security-products.html＞(参照 2016-11-24)
[35] 長谷川皓一，山口由紀子，嶋田創，高倉弘喜：標的型攻撃に対するインシデント対応支援システム，情報処理学会論文誌，Vol. 57，No. 3，pp. 836-848，2016
[36] 細川 嵩：制御システムセキュリティの対策技術紹介，制御システムセキュリティセンター，入手先＜https://www.jpcert.or.jp/present/2016/20160217_CSC-CSSC.pdf＞(参照 2016-11-24)
[37] National Institute of Standards and Technology : The Security Content Automation Protocol (SCAP)，入手先＜https://scap.nist.gov/＞(参照 2016-11-24)
[38] MITRE : Common Vulnerabilities and Exposures，入手先＜http://www.cve.mitre.org/news/archives/2016/news.html＞(参照 2016-11-24)
[39] National Vulnerability Database : Common Vulnerability Scoring System，入手先＜https://nvd.nist.gov/cvss.cfm＞(参照 2016-11-24)
[40] MITRE : Open Vulnerability and Assessment Language，入手先＜https://oval.mitre.org/＞(参照 2016-11-24)
[41] IDC Research, Inc. : Worldwide Specialized Threat Analysis and Protection Market Shares, 2014: Rapidly Evolving Security Defenses，入手先＜http://www.idc.com/getdoc.jsp?containerId=259667＞(参照 2016-11-24)

# 2. マルウェア挙動解明

本章では，既存の動的解析ツールを活用して複数の解析エンジン，異なる種類の解析環境（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先等の挙動の有無と，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，全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．

## 2.1. 多種環境マルウェア動的解析システム

本章では，1 章で述べた課題 1 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 2.1 に示す．

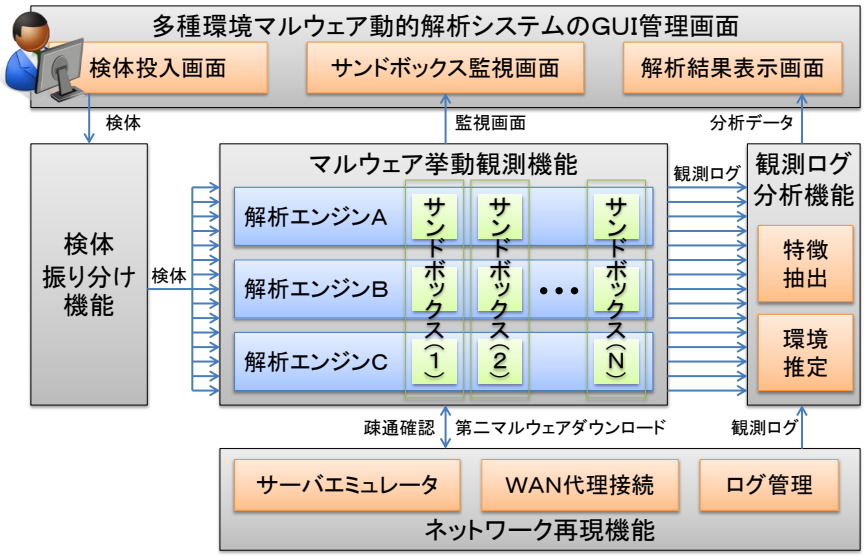



図 2.1　アーキテクチャと検体解析の流れ

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．観測ログ分析機能は，マルウェア挙動観測機能から出力された大量の観測ログを収集し，各サンドボックスにおける検体の活動状況（例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）を分析し，検体による生成ファイルや，ネットワーク接続先 URL を抽出する．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取得したログを分析し，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．

### 2.1.1. 検体振り分け機能

検体振り分け機能は，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，予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．解析エンジン毎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，マルウェア解析者がマルウェア解析業務中に誤ってマルウェアを実行してしまう事故を予防する．

### 2.1.2. 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（マルウェア本来の挙動を解明できる確率）の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，全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 2.1 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，1 章で述べた Threat Analyzer や Cuckoo Sandbox 等の既存の動的解析ツールを指す．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．従って，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繋がるため，特定のプラットフォー

ム（後述）を検出して挙動を変化させ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．

例えば，シンプルな構成である表 2.1 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 2×3×2×3×2×(7+1)=576 通りの環境があり得る．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（後述）．なお，マルウェア挙動観測機能は，各サンドボックスが検体を実行完了した時，あるいは事前に設定した時間を経過した時に自動的に環境を解析前（感染前）へ復元する機能も備える．

表 2.1　サンドボックス環境の環境条件例

| 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<br>・仮想 PC (VMware ESXi) | ・32bit (x86)<br>・64bit (x64) | ・Windows XP<br>・Windows Vista<br>・Windows 7 | ・英語<br>・日本語 | ・Microsoft Office<br>・Adobe Reader<br>・Adobe Flash<br>・Internet Explorer<br>・Java<br>・Media Player<br>・一太郎 |
| Cuckoo Sandbox | ・仮想 PC (Virtual Box) | | | | |
| 計 2 | 3 | 2 | 3 | 2 | 7 |

表 2.2　ネットワーク再現機能実装サービス

| 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) |

### 2.1.3. ネットワーク再現機能

標的型攻撃で用いられる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二のマルウェアをダウンロードしたり，C&C サーバ[7]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[27]．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」[42]を用いて表 2.2 に示す 21 のサービスのエミュレーションを行う．

### 2.1.4. 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条件を推定する．

(ア) 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[43]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．

以下に本章で実装した 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．

i. 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．

ii. 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

[7] 攻撃者がマルウェア感染端末を遠隔操作（指令）するために利用されるサーバ（Command and Control server）。

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，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．

iii. 外部ネットワーク接続判定

検体が第二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．

(イ) マルウェア動作環境推定

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．マルウェア動作環境推定機能の目的は，入力データから，2.1.6 項で定義する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．

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，環境条件と顕現状態の関係をアソシエーションルールとして抽出する．本章では，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，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）は次式で求める．

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

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

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

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 2.3 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 2.4 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 2.5 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, 顕現サンドボックス群の多くに共通の環境条件（顕現条件）を抽出することにあることから，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，支持度下限は多数の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 2.5 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 2.3 で示した前提条件と一致する．

表 2.3　解析結果例

| 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 2.4　トランザクション例

| 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 2.5　アソシエーションルールと指標

| 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 |

### 2.1.5. 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，個々のサンドボックス上の検体の活動状況（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 2.1.4 節(ア)の判定結果を確認することができる．

### 2.1.6. 検知および顕現の定義

本システムでは，2.1.4 項に述べた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，例えば 2.1.4 項(ア)に述べた「i の判定結果が有の場合」や「iii の判定結果が外部のホストに接続した場合」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，(ア)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 ．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[8].

[8] 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．

## 2.2. M3AS の実装

### 2.2.1. システム構成

M3AS のシステム構成を図 2.2 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．

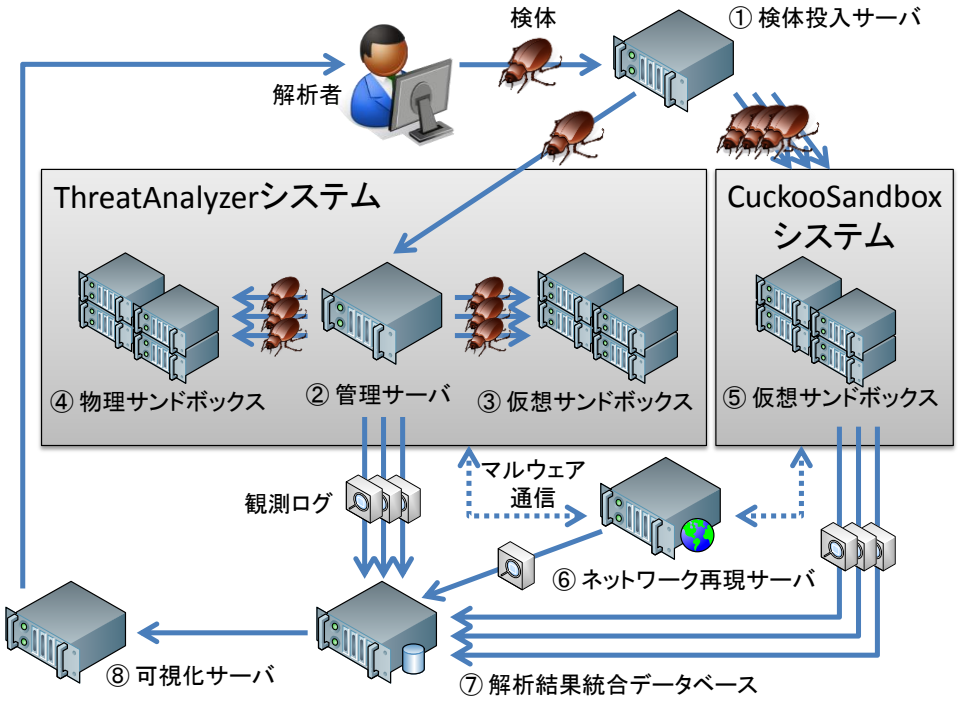



図 2.2　M3AS システム構成

### 2.2.2. サンドボックス構成

M3AS のサンドボックスの構成について述べる．表 2.1 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 2.6 に示す．また，各環境条件の選定方針を表 2.7 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する．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 まで区別して全て選定した．ただし 2014 年 12 月時点で，前述した 2 種の解析エンジンはサンドボックスを最新の OS で構成することに対応していなかったため，Windows 8 以降の OS は選定対象から除外した．また同様の理由からアーキテクチャも 32bit 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．

表 2.6　評価向けサンドボックスの環境条件

| 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),<br>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,<br>Viewer19, Viewer23 | 5 |
| 合計 | | 44 |

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[44]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 2.6 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示す．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た．

本章の評価では，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 2.8 に一部を例示する．

表 2.7　環境条件の選定方針

| 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 に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当 |

表 2.8　環境条件$Y_1, \cdots, Y_N$の例（一部）

| SB＃ | Threat Analyzer | Cuckoo Sandbox | VMware ESXi | Virtual Box | 物理 |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 0 |
| : | : | : | : | : | : |
| N(=76) | 0 | 1 | 0 | 1 | 0 |

| SB＃ | Windows XP sp2 | Windows XP sp3 | … | 一太郎 Viewer19 | 一太郎 Viewer23 |
|---|---|---|---|---|---|
| 1 | 1 | 0 | … | 0 | 0 |
| 2 | 0 | 1 | … | 0 | 1 |
| 3 | 0 | 1 | … | 1 | 0 |
| : | : | : | : | : | : |
| N(=76) | 0 | 0 | … | 0 | 1 |

## 2.3. 評価実験

### 2.3.1. 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 2014 年 10 月に著者が所属する情報システム部門から入手したマルウェア 633 種を解析し，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，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 2.9 に示す．

633 検体のうち，全てのモジュールで検知されたマルウェアは全体の 17.5%，全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[45]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伴わない古いタイプのワームが多く含まれていた．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 2.1.4 項(イ)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．

解析によって顕現マルウェアが感染時にインターネットに接続する URL を 6,542 種類得た．表 2.10 に顕現状態の結果を示す．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．

2.1.3 項で述べたネットワーク再現機能へのアクセス状況を表 2.11 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．

表 2.9　モジュールの検知結果

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

（括弧内は検体数）

表 2.10　マルウェアの解析結果

| 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 2.11　ネットワーク再現機能へのアクセス状況

| 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% |

### 2.3.2. 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．

(ア) マルウェア観測処理

(イ) ログ分析処理

(ウ) 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 2.12 と図 2.3 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 2.13 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．

表 2.12　マルウェア解析処理時間

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

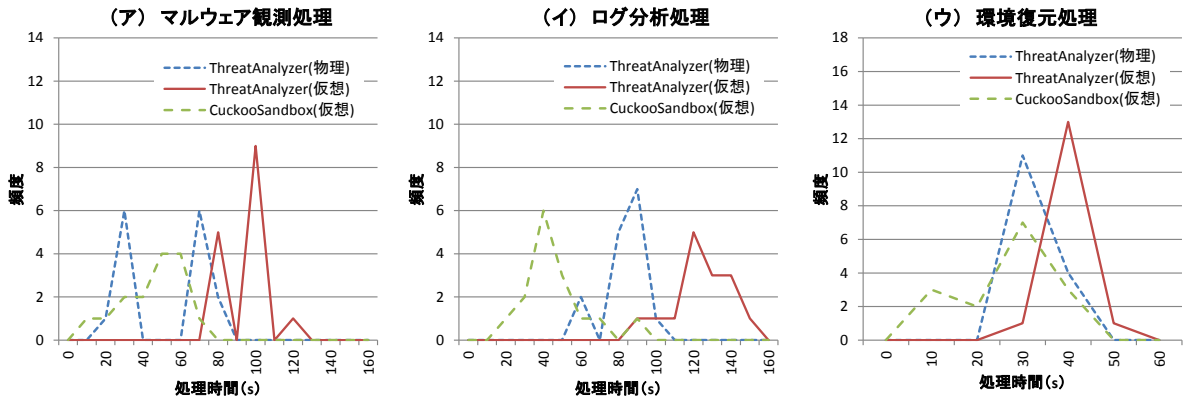



図 2.3　解析処理時間（サンドボックス単位）

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 は全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．

表 2.13　M3AS 構成ハードウェアスペック

| 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<br>Memory：48GB<br>Storage：HDD900GB × 5 (RAID5) | ※1 |
| ② ThreatAnalyzer 管理サーバ | | CPU：intel Xeon E5-2680(2.70GHz, 8cores) × 2<br>Memory：64GB<br>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)<br>Memory：8GB<br>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 | |

### 2.3.3. 環境選択型マルウェアの顕現条件推定精度

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．

(ア) Apriori アルゴリズムの設定

ここでは 2.1.4 項(イ)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毎に，各サンドボックスの顕現状態（1 アイテム）と，表 2.6 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．

(イ) アソシエーションルールの抽出

2.3.1 項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 2.4 に検体毎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．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．例えば，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 2.14 に示す．全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，確信度は省略する．検体 1 は顕現したサンドボックスの全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 の条件の他に，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．

表 2.14　アソシエーションルールの一部

| 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 |

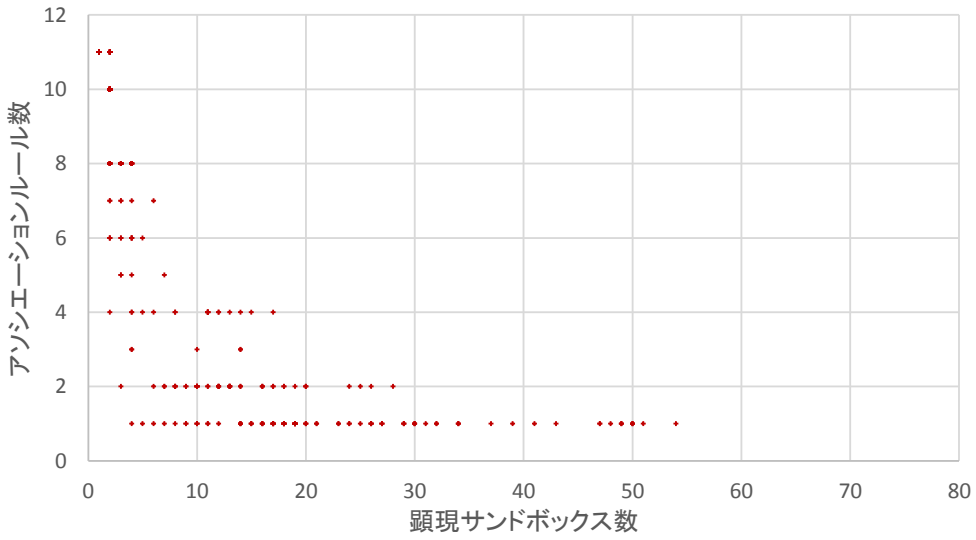



図 2.4　アソシエーションルール数

(ウ) 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の抽出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．

アソシエーションルールが期待通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，全てのケースでマルウェアの実行結果から期待通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．更に，環境条件の全ての組み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．

上記の不安定性の問題が存在せず，すべての可能な組み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．サンドボックス数Mがk個とした場合，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などがあるため，ほとんど全てのマルウェアのルールを抽出できると考える．

そこで，環境選択型マルウェアのアソシエーションルールの抽出可否と，サンドボックスの数がアソシエーションルールの抽出可否に与える影響を検証するため，本章では，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 2.15）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．

表 2.15　環境選択型マルウェアサンプル

| 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[46]の新種[47]）で，一太郎の脆弱性（CVE-2013-5990[48]）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[49]を利用して物理でのみ動作するよう作製 |
| C | 英語版 Windows | 検知名 TROJ_FAKEAV.BME[50]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ため日本語版 OS では顕現しないことが判明 |

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 2.15 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．

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 2.16 に示す．全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 2.15 の条件と一致する．検体 B は 4 つのルールが抽出された．中でも 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わかる．これも表 2.15 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 2.15 の条件と一致する．

次に 2 つ目の評価について述べる．図 2.5 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 種でそれぞれ 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．

表 2.16　環境選択型マルウェアのアソシエーションルール

| 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 |

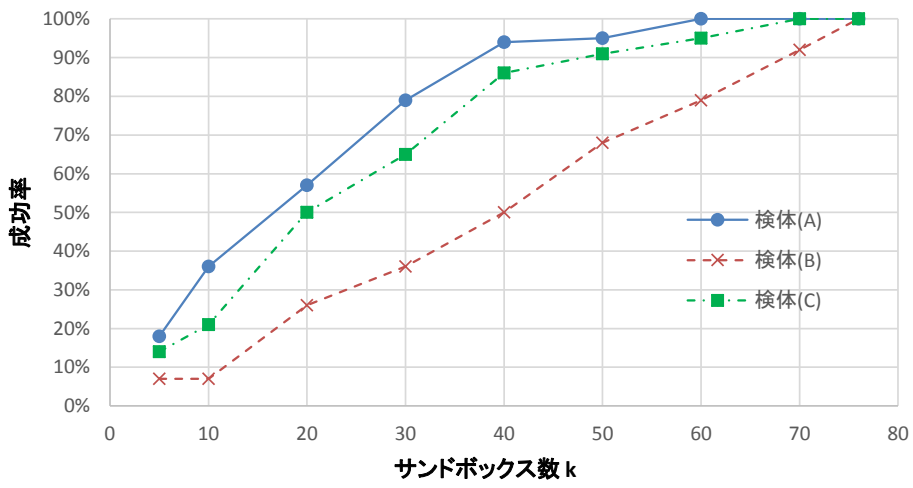



図 2.5　アソシエーションルール抽出成功率

### 2.3.4. サンドボックス構成の課題

前項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 2.6 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，2.1.4 項(ア)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．2.1.4 項(イ)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 2.1.4 項(ア)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もつべきであるが，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．

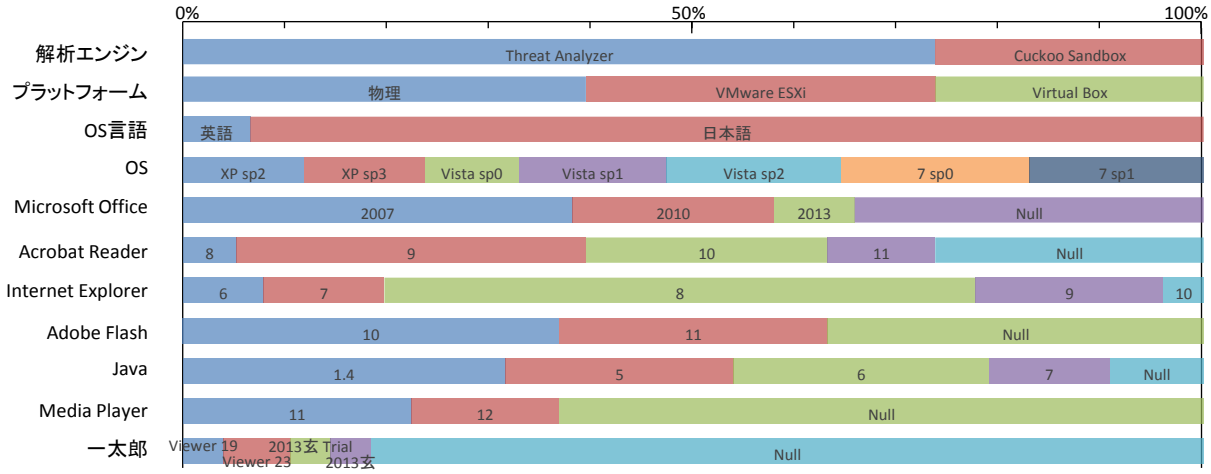



図 2.6　環境条件の分布

## 2.4. 考察

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言える．

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する観点では，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費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．本課題については 5 章で述べる．

## 2.5. 結論

本章では巧妙化が益々進むマルウェアに対抗するために，環境選択型マルウェアが顕現しやすい 76 種の環境を再現したサンドボックス上でマルウェアを同時並列的に解析してマルウェアによる外部ホストへの通信先等の特徴を自動抽出する機能と，環境選択型マルウェアが顕現した環境条件を Apriori アルゴリズムにより導出する機能を備えた M3AS を提案した．

M3AS を用いて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 83%のマルウェアが顕現（外部ホストへ接続）し特徴を検知できることを確認した．顕現条件推定機能により，特定の環境でのみ顕現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．また，解析によって 6,542 種類のマルウェア接続先（マルウェアによる外部ホストへの通信先）を抽出した．

本章で実装した M3AS は，構成する全機能を可搬ハーフラックに構築している．このため，マルウェアの検体を外部に持ち出すことが難しい組織に届いたマルウェアや，クローキング技術等により標的とされている組織でしか顕現しないマルウェアの解析も可能である．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掛かりや，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繋げたりすることができる．

## 参考文献


[42] Thomas Hungenberg & Matthias Eckert： INetSim Internet Services Simulation Suite, 入手先<http://www.inetsim.org/index.html>(参照 2014-11-24)

[43] 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)

[44] 情報通信処理機構 ： 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)

[45] VirusTotal - Free Online Virus : Malware and URL Scanner, 入手先< https://www.virustotal.com/ja/>(参照 2015-3-24)

[46] 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)

[47] naked security : From the Labs: New PlugX malware variant takes aim at Japan, 入手先<http://nakedsecurity.sophos.com/2013/12/04/new-plugx-malware-variant-takes-aim-at-japan/>(参照 2014-11-24)

[48] Common Vulnerabilities and Exposures： CVE-2013-5990, 入手先<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5990>(参照 2014-11-24)

[49] Sudeep Singh : Breaking the Sandbox, 入手先< http://www.exploit-db.com/wp-content/themes/exploit/docs/34591.pdf>(参照 2014-11-24)

[50] 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＞(参照 2014-11-24)

# 3. マルウェア通信制御

## 3.1. 背景と目的

マルウェアの中には，攻撃者が用意したマルウェア配布サーバから第二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[51]が存在する．また，マルウェア配布サーバの中には，攻撃の正体を隠ぺいするため，マルウェアがダウンロードできるのは，最初のアクセス 1 回のみで，同じ IP アドレスからアクセスすると，2 回目以降は正規サイトに誘導されるものも存在する[52]．さらに，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（以下，クローキング）も確認されている[53][54].

このため，マルウェアによる攻撃の全貌を明らかにするためには，インターネットに接続させた状態で動的解析するのが望ましい．しかし，マルウェアによるインターネットへの通信をすべて許可すると，解析中に解析環境が外部（インターネット上のサーバ等）に攻撃を行って他組織に危害を与える等，意図せず攻撃者に加担してしまうことになりかねない.

そこで本章では，ダウンローダ型マルウェアの通信（以下，MW ダウンロード通信）を検出し，当該 MW ダウンロード通信のみをインターネットに代理で接続することにより，マルウェアによる外部への攻撃を抑制しつつマルウェア解析を行うマルウェア通信制御システムを提案する.

## 3.2. 関連研究

動的解析ツールを用いてマルウェアの解析を行う場合，インターネットから隔離した閉塞環境で解析を行う方法と，インターネットに接続した環境で解析を行う方法とが存在する.

閉塞環境を用いた解析は，解析中に外部へ攻撃してしまう不測の事態を抑制することができ，ネットワークアクセスを伴わないマルウェアの解析には有効な方法である．しかし，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し，隔離された環境では本来の挙動を現さない種も存在する．青木らも閉塞環境よりもインターネットに接続した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[55]．そこで，ネットワーク環境をエミュレートすることで，マルウェアに対して疑似的なインターネット環境を提供するシステム「M3AS」を 2 章で提案した．しかし，疑似インターネット環境を利用した解析手法では，ダウンローダ型マルウェアのようなインターネットを介してマルウェア配布サーバから新たなマルウェアをダウンロードするマルウェアは，新たなマルウェアをダウンロードするまでの挙動は解析できるが，新たなマルウェアのダウンロード後の挙動（新たなマルウェアの実行や実行後の挙動）を解析することができない．以降では，ダウンローダ型マルウェアによってダウンロードされた新たなマルウェアのことを「第二のマルウェア」と呼ぶ.

一方，インターネットに接続してマルウェア解析を行う解析システムとして，Botnet

Watcher[56]が挙げられる．Botnet Watcher では，GateKeeper と呼ばれるモジュールが，解析環境とインターネットとの間の通信を仲介しており，C&C サーバとの通信や，HTTP によるファイルダウンロードの通信であると判断された場合にインターネットと接続する．しかし，上記手法は，C&C サーバとの通信や，HTTP によるファイルダウンロードの通信を装った外部への攻撃を抑制することができない．また，Kreibich ら[57]は，通信フローごとに，ポリシに従って通信可否の制御を行う GQ honyform というシステムを提案している．しかし，Kreibich らの手法ではポリシの生成方法を言及しておらず，ポリシの決定は解析を行う管理者にゆだねられている．Yoshioka ら[58]は，解析環境で観測されたマルウェアの通信の中から危険性が低いと判断された通信に関して，順次インターネット接続を許可して解析を行う，マルウェア動的解析システムを提案している．しかし，Yoshioka らの手法は未知の脆弱性を突く外部への攻撃を抑制できない．

そこで，本章では，MW ダウンロード通信を判定し，当該 MW ダウンロード通信のみをインターネットに接続させるマルウェア通信制御システムを提案する．提案手法の新規性は，ダウンローダ型マルウェアの挙動に着目して MW ダウンロード通信を判定するところにあり，従来技術では防げなかった外部への攻撃（HTTP ダウンロード通信を装った攻撃や，未知の脆弱性を突く攻撃）を抑制できることが提案手法の優位性である．提案手法と既存技術との比較を表 3.1 にまとめる．

表 3.1　既存技術との比較

| | 提案手法 | 青木ら[55] | Kreibich ら[57] | Yoshioka ら[58] |
|---|---|---|---|---|
| 概要 | MW ダウンロード通信のみを接続 | C&C 通信や HTTP ダウンロード通信を接続 | 管理者の定めるポリシに従い接続 | 危険性が低いと判断された通信を接続 |
| 許可する通信の判定方法 | MW ダウンロード通信の判定結果を利用 | プロトコル識別結果を利用 | 管理者の定めるポリシを利用 | 統計検査，セッション検査，脆弱性検査の結果を利用 |
| 外部攻撃の少なさ | ○<br>（攻撃なし） | ×<br>（例：HTTP ダウンロードを装った攻撃） | -<br>（管理者の定めるポリシに依存） | ×<br>（例：未知の脆弱性を突く攻撃） |

## 3.3. マルウェア通信制御システムの提案

本章では，MW ダウンロード通信を検出し，当該 MW ダウンロード通信のみをインターネットに接続させるマルウェア通信制御システムを提案する．

### 3.3.1. MW ダウンロード通信判定手法

情報通信処理機構 IPA は，ダウンローダ型マルウェアが，2015 年第 4 四半期にもっとも多く検出された不正プログラムと報告[59]している．ダウンローダ型マルウェアは，多くの場合メールに添付して送られ，本命とされる第二のマルウェアに感染させるために利用される．

ダウンローダ型マルウェアを用いた攻撃の流れを図 3.1 に示す．ダウンローダ型マルウェアを用いた攻撃では，まず攻撃者がダウンローダ型マルウェアを添付したメールを被害者に送付する（図 3.1①）．被害者が誤って添付されたダウンローダ型マルウェアを実行（図 3.1②）してしまった場合，ダウンローダ型マルウェアは，攻撃者が用意したマルウェア配布サーバへ接続（図 3.1③）し，第二のマルウェアをダウンロード（図 3.1④）する．さらに，ダウンローダ型マルウェアは，ダウンロードしてきた第二のマルウェアを実行（図 3.1⑤）し，第二のマルウェアに感染させる．マルウェア配布サーバへの接続から，第二のマルウェアの実行（図 3.1③～⑤）は，ダウンローダ型マルウェアの働きによるものであり，被害者が気付かないうちに，第二のマルウェアに感染してしまう．

提案するマルウェア通信制御システムでは，ダウンローダ型マルウェアが第二のマルウェアを取得し，実行する機能を備えている点[60]に着目し，この動きを利用して，MW ダウンロード通信の判定を行う．MW ダウンロード通信判定の流れを図 3.2 に示す．

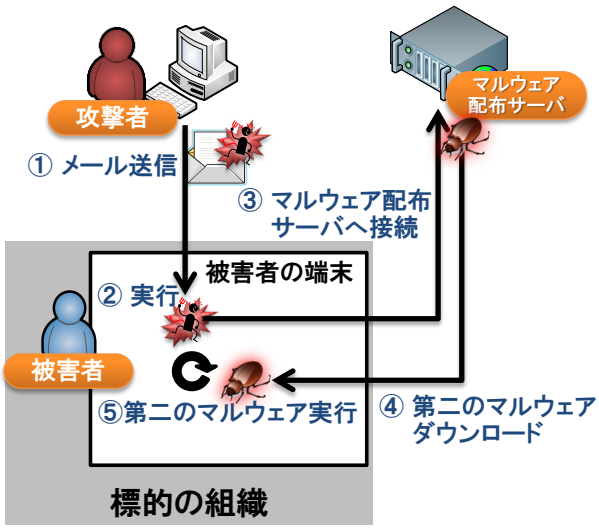



図 3.1　ダウンローダ型マルウェアを用いた攻撃の流れ

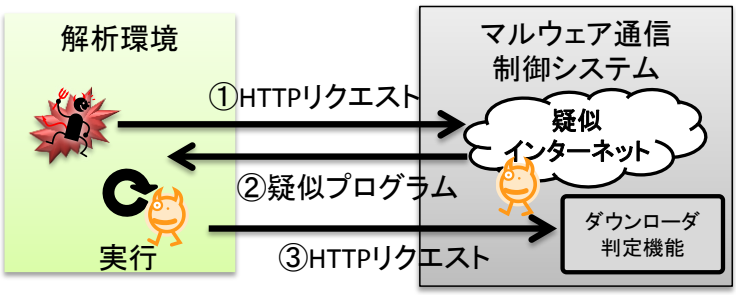



図 3.2　MW ダウンロード通信判定の流れ

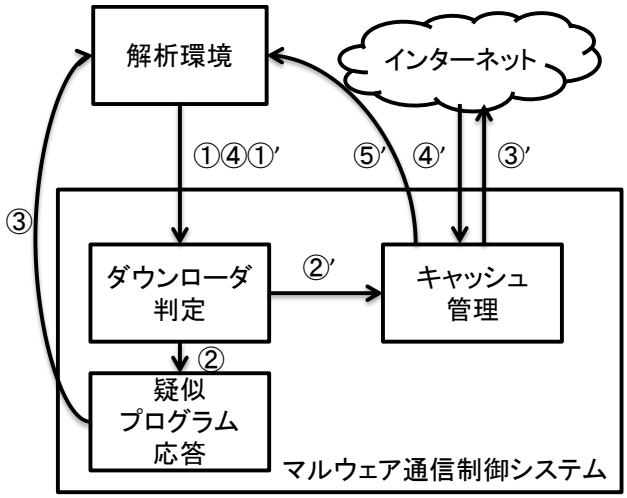



図 3.3　マルウェア通信制御システムの概要

マルウェア通信制御システムが，解析環境からインターネット向けの HTTP 通信を受信すると，マルウェア通信制御システムに通信を行う疑似プログラムを生成し，解析環境に応答する．解析環境上でこの疑似プログラムが実行されると，マルウェア通信制御システムへ新たな通信が行われるため，先ほどの HTTP 通信が MW ダウンロード通信であると判定できる．本判定手法により，2 章で提案した M3AS の多数の解析環境（サンドボックス）に変更を加えることなく，MW ダウンロード通信の判定が可能となる．

なお，3.1 節で述べたように，マルウェア配布サーバの中には，解析回避機能（クローキング等）を備えたものも存在する．このような解析回避機能への対応については 3.3.3 項で述べる．

### 3.3.2. 提案システムの概要

提案するマルウェア通信制御システムの概要を図 3.3 に示す．

提案するマルウェア通信制御システムは，以下 3 つの特徴を持った機能から構成される．なお，

これらの機能の詳細については，3.3.3 項で述べる．

(ア) ダウンローダ判定

マルウェアの通信がダウンローダ型マルウェアによる MW ダウンロード通信か否かを判定する

(イ) 疑似プログラム応答

MW ダウンロード通信の判定を行うために，疑似的なプログラムを生成し応答する

(ウ) キャッシュ管理

インターネットからダウンロードした第二のマルウェアをキャッシュ（一時保存）する

マルウェア通信制御システムは，2 つのフェーズを用いた解析により，マルウェアの通信を制御する．フェーズ 1 では，マルウェアが行う通信が MW ダウンロード通信か否かの判定を行い，フェーズ 2 では，インターネットから第二のマルウェアをダウンロードし，解析する．

まず，フェーズ 1 における解析の流れを述べる．ダウンローダ判定機能は解析環境から通信を受信（図 3.3①）すると，当該通信が MW ダウンロード通信か否かの判定を行う．MW ダウンロード通信でないと判定された場合には，疑似プログラム応答機能へ，疑似プログラム生成要求を送信（図 3.3②）する．疑似プログラム応答機能は，ダウンローダ判定機能から疑似プログラム生成要求を受信すると，マルウェア通信制御システムへ通信を行う疑似プログラムを生成し，解析環境へ応答（図 3.3③）する．解析環境は，疑似プログラム応答機能から応答された疑似プログラムを実行し，マルウェア通信制御システムへ通信（図 3.3④）を行う．マルウェア通信制御システムは，当該通信を観測することにより，先ほどの通信（図 3.3①）が MW ダウンロード通信であったことを検出する．

続いて，フェーズ 2 における解析の流れを述べる．ダウンロード判定機能は，解析環境から通信を受信（図 3.3①’）すると，当該通信が MW ダウンロード通信か否かの判定を行う．MW ダウンロード通信であると判定された場合には，通信をキャッシュ管理機能へ転送（図 3.3②’）する．キャッシュ管理機能は，ダウンローダ判定機能から MW ダウンロード通信を受信すると，当該通信に対する応答が，キャッシュ管理機能にキャッシュされているかどうかをダウンロード先の URL から判定する．キャッシュされていない場合は，MW ダウンロード通信をインターネットへ送信（図 3.3③’）する．キャッシュ管理機能は，インターネットからの応答を受信（図 3.3④’）すると，当該応答を自身の持つキャッシュデータに格納し，解析環境へ応答（図 3.3⑤’）する．

以上のように，2 つのフェーズに分けて解析を行うことにより，MW ダウンロード通信の検出と，当該 MW ダウンロード通信のみをインターネットに接続させるマルウェア通信制御システムを実現する．

### 3.3.3. 提案システムの詳細

提案システムを構成する 3 つの機能について，その詳細を述べる．

(ア) ダウンローダ判定

ダウンローダ判定機能では，表 3.2に例示するダウンローダ判定リストを用いて，マルウェアの通信がダウンローダ型マルウェアによる MW ダウンロード通信か否かの判定を行う．ダウンローダ判定リストは，判定対象 URL と判定結果の組み合わせからなり，判定結果が 1 の場合は該当する判定対象 URL への通信が MW ダウンロード通信であると判定し，判定結果が 0 の場合は該当する判定対象 URL への通信が MW ダウンロード通信ではないと判定することを表す．具体的には，解析環境からインターネット向けの HTTP 通信を受信すると，ダウンローダ判定リストの判定対象 URL にアクセス先の URL が存在するか否かを検査し，ダウンローダ判定リストにアクセス先の URL が存在し，かつ，判定結果が 1（MW ダウンロード通信であると判定される）の場合には，キャッシュ管理部へ通信を転送する．ダウンローダ判定リストの判定対象 URL にアクセス先が存在しない，或いは，アクセス先は存在するが判定結果が 0（MW ダウンロード通信ではないと判定される）の場合は，疑似プログラム応答機能へ疑似プログラム生成命令を送信する．なお，この時，ダウンローダ判定リストの判定対象 URL にアクセス先が存在しない場合には，当該アクセス先をダウンローダ判定リストに登録し，判定結果に 0 を設定する．

また，疑似プログラム実行による通信を受信すると，当該 URL のパス部分をダウンローダ判定リストの判定対象 URL と比較する．一致する URL が存在する場合に，判定結果を 1 に設定し，以後，当該 URL への通信は MW ダウンロード通信として判定する．

表 3.2 の例では，「http://example.com/example.exe」への通信が MW ダウンロード通信として判定される．

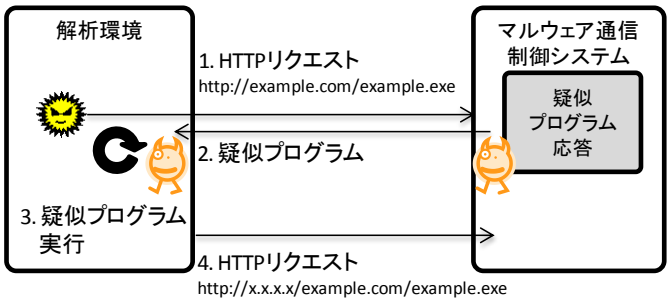
表 3.2　ダウンローダ判定リストの例

| 判定対象 URL | 判定結果 |
|---|---|
| http://example.com/example.exe | 1 |
| http://example.net/abc.exe | 0 |
| : | : |

(イ) 疑似プログラム応答

疑似プログラム応答機能では，マルウェアが行う通信に対して，マルウェア通信制御システムへ通信を行う疑似プログラムを生成し，解析環境に応答する．具体的には，マルウェアが行う通信に対して，その通信先の URL を識別可能な情報を実行コードに埋め込んだ疑似プログラムを通信（HTTP リクエスト）毎に自動生成し，解析環境へ応答する．

図 3.4 を用いて処理の流れを説明する。疑似プログラム応答機能は，マルウェアによるインターネット通信を受信すると，当該 URL と，マルウェア通信制御システムの IP アドレス（例えば x.x.x.x）を含む新たな URL（例えば http://x.x.x.x/example.com/example.exe）へアクセスする疑似プログラムを毎度自動生成し，解析環境に応答する．擬似プログラムは，マルウェア通信制御システムへ通信を行うベースプログラムを事前に用意しておき，ダウンロード要求があるたびに，ベースプログラムの一部をダウンロード要求内容に応じて動的に変更したファイルを生成して応答する．ベースプログラムのバイナリダンプと，動的に変更される箇所（図中枠内）を図 3.5 に示す．




※x.x.x.xは，マルウェア通信制御システムのIPアドレスを示す。

図 3.4　疑似プログラムの応答

```
          +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  0123456789ABCDEF
00AA60    39 3A 3B 3C 3D 3E 3F 40-41 42 43 44 45 46 47 48  9:;<=>?@ABCDEFGH
00AA70    49 4A 4B 4C 4D 4E 4F 50-51 52 53 54 55 56 57 58  IJKLMNOPQRSTUVWX
00AA80    59 5A 5B 5C 5D 5E 5F 60-61 62 63 64 65 66 67 68  YZ[¥]^_`abcdefgh
00AA90    69 6A 6B 6C 6D 6E 6F 70-71 72 73 74 75 76 77 78  ijklmnopqrstuvwx
00AAA0    79 7A 7B 7C 7D 7E 7F 00-68 74 74 70 3A 2F 2F 31  yz{|}~..http://1
00AAB0    2E 31 2E 31 2E 31 00 75-75 75 75 75 75 75 75 75  .1.1.1.uuuuuuuu
00AAC0    75 75 75 75 75 75 75 75-75 75 75 75 75 75 75 75  uuuuuuuuuuuuuuuu
00AAD0    75 75 75 75 75 75 75 75-75 75 75 75 75 75 75 75  uuuuuuuuuuuuuuuu
00AAE0    75 75 75 75 75 75 75 75-75 75 75 75 75 75 75 75  uuuuuuuuuuuuuuuu
00AAF0    75 75 75 75 75 75 75 75-75 75 75 75 75 75 75 75  uuuuuuuuuuuuuuuu
00AB00    75 75 75 75 75 75 75 75-75 75 75 75 00 00 00 00  uuuuuuuuuuuu....
00AB10    65 78 61 6D 70 6C 65 2E-63 6F 6D 2F 65 78 61 6D  example.com/exam
00AB20    70 6C 65 2E 65 78 65 00-76 76 76 76 76 76 76 76  ple.exe.vvvvvvvv
00AB30    76 76 76 76 76 76 76 76-76 76 76 76 76 76 76 76  vvvvvvvvvvvvvvvv
```

図 3.5　疑似プログラムの生成

解析環境で実行されたマルウェアがダウンローダ型マルウェアであった場合，疑似プログラム応答機能で生成された疑似プログラムが解析環境上のダウンローダ型マルウェアによって実行され，マルウェア通信制御システムへの HTTP リクエストが送信される．当該 HTTP リクエストをマルウェア通信制御システムのダウンローダ判定機能で観測することにより，MW ダウンロード通信か否かを判定する．

(ウ) キャッシュ管理

3.1 節で述べたように，マルウェア配布サーバの中には，同一の IP アドレスによる 1 回目のアクセスに対してのみしか第二のマルウェアを配布しないものが存在する．M3AS では個々のサンドボックスにグローバル IP アドレスを割り振ることは現実的でないことから，1 つのグローバル IP アドレスを全てのサンドボックスで共有することになる．このため，最初に顕現（マルウェア配布サーバにアクセス）したサンドボックスしか第二のマルウェアをダウンロードできない問題が発生し，解析の精度に悪影響を及ぼす．また，解析を複数回実施したい場合に，2 回目以降の解析では第二のマルウェアがダウンロードされず，解析がうまく行われないという問題も発生する．

この問題を解決するため，1 回目のアクセスに対してインターネットからダウンロードした第二のマルウェアをキャッシュする，キャッシュ管理機能を用いる．解析環境から MW ダウンロード通信が発生した際に，当該通信先の応答ファイルがキャッシュされていれば，キャッシュ管理機能が応答する．具体的には，インターネットからの応答を URL ごとにファイルとして保存し，表 3.3 に示すキャッシュリストを用いて管理する．

表 3.3　キャッシュリストの例

| 通信先 URL | 応答保存先 |
|---|---|
| http://example.com/example.exe | /data/example.com/example.exe |
| http://example.net/abc.exe | - |
| : | : |

## 3.4. マルウェア通信制御システムの実装

本節では，マルウェア通信制御システムの実装について述べる．マルウェア通信制御システムの各機能と，それらの実装に用いたソフトウェアの関係を図 3.6 に示す．

マルウェア通信制御システムの機能のうち，ダウンローダ判定機能，疑似プログラム応答機能，キャッシュ管理機能は man-in-the-middle 型のプロキシサーバである mitmproxy[61]を用いて実装した．また，ダウンローダ判定リスト，キャッシュリストは，SQLite を用いて実装した．なお，ベースプログラムとして，Windows で実行可能な実行ファイルを用意した．

1 番目の NIC（Network Interface Card）で受信した HTTP パケット（80 番向けポート）は，iptables[9]の機能を用いて mitmproxy が監視しているポート(8080 番)に転送される．なお，mitmproxy からインターネットへの接続は，2 番目の NIC を用いて行われる．

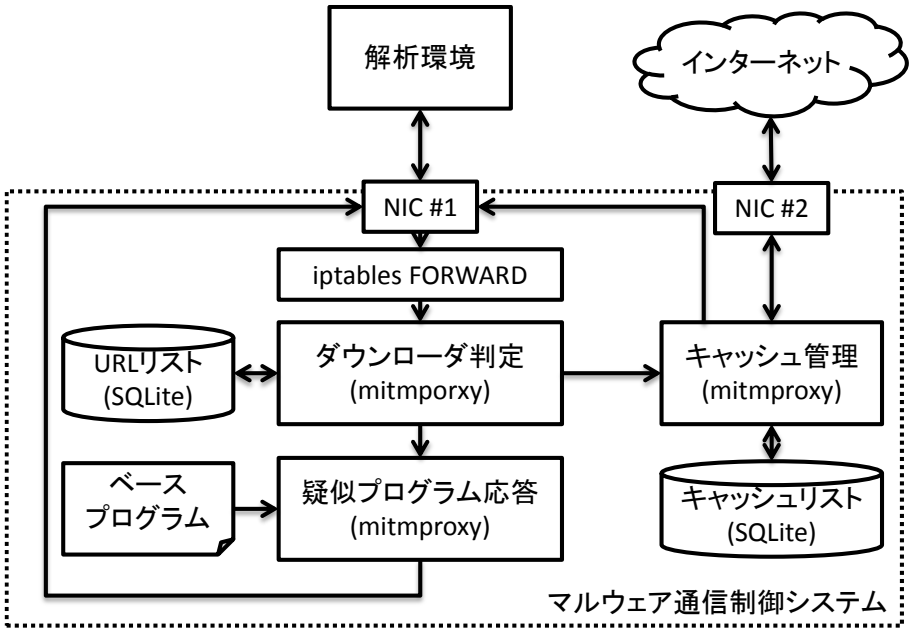



図 3.6　マルウェア通信制御システムの機能構成

[9] Linux に標準搭載されたパケットフィルタリング設定を操作するコマンド

## 3.5. 評価実験参照

本節では，実装したマルウェア通信制御システムを用いて評価し、結果について述べる．

### 3.5.1. 評価目的

マルウェア通信制御システムは，マルウェアの MW ダウンロード通信を検知し，当該 MW ダウンロード通信のみをインターネットに接続するシステムである．マルウェア通信制御システムを以下の観点で評価する．

(ア) 検知性能

提案手法により，世の中に存在するダウンローダ型マルウェアの MW ダウンロード通信を検知できるかを評価する．

(イ) 有効性

ある組織に届いた検体を解析し，どの程度ダウンローダ型マルウェアが存在するのか，また，提案手法により，閉塞環境での解析や青木ら[55]の手法に基づく解析と比較し，どの程度新たな脅威が明らかになるのかを評価する．

### 3.5.2. 評価方法

本項では前項に示した評価の目的を達成するための評価方法について述べる．

(ア) 検知性能

実在するダウンローダ型マルウェアを入手し，マルウェア通信制御システムによってマルウェアの MW ダウンロード通信を検知できるかを評価する．なお，Symantec 社のウイルス対策ソフトでダウンローダ型マルウェアと判定された以下マルウェアを評価に用いた（表 3.4）．

(イ) 有効性

2015 年 2 月に著者の所属する組織に届いたファイルの内，動的解析によって実行ファイルのドロップや，他のプロセスへのインジェクションなどの不審な挙動（マルウェアと思われる挙動）を観測した検体（644 検体）を用いて，ダウンローダ型マルウェアの数を評価する．また，当該ダウンローダ型マルウェアが通信するマルウェア配布サイトに接続し，第二のマルウェアを取得できるかを評価する．なお，評価では，2 章で提案および実装した M3AS を利用してマルウェアの解析を行った．

表 3.4　検知性能評価用マルウェア

| 検知名 | ハッシュ値（MD5） |
|---|---|
| Downloader | ee5f956efb93e2981b9ce9b75680c299 |
| W97M.Downloader | cf9443e43b990077a3862aa4f9337fb2 |
| JS.Downloader | 20de9a1fc71d4654f980cee8e7ce84f2 |

図 3.7 に評価環境を示す．多種環境マルウェア動的解析システムは，環境によって振る舞いの異なるマルウェアを解析するために，様々な OS やアプリケーションを組み合わせた解析環境を約 70 種類備えている．マルウェアの解析を行う際には，これらの解析環境のすべてにマルウェアを振り分けて実行する．

また，比較対象として，インターネットへの接続を行わない閉塞環境を用いた解析と，青木ら[55]の手法に基づく解析も実施する．なお，閉塞環境での解析では，INetSim[62]を用いて，ネットワーク環境をエミュレートする．

### 3.5.3. 評価結果

(ア) 性能評価結果

評価用マルウェアに対して，MW ダウンロード通信として判定された通信先を表 3.5 に示す．性能評価に用いた全てのマルウェアすべてにおいて，MW ダウンロード通信を検出することを確認した．なお，ここでは安全性への配慮から通信先の一部をマスキングして記載する．

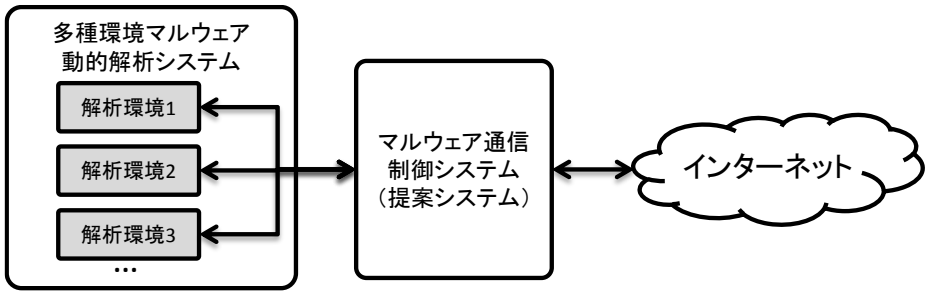



図 3.7　評価環境

表 3.5　検知性能評価結果

| ハッシュ値（MD5） | 通信先 URL |
|---|---|
| ee5f956efb93e2981b9ce9b75680c299 | http://k[snip].com/putty3.exe |
| cf9443e43b990077a3862aa4f9337fb2 | http://91.[snip]/upd2/install.exe |
| 20de9a1fc71d4654f980cee8e7ce84f2 | http://d[snip]/document.php?id=xxx |

(イ) 有効性評価結果

評価用の 644 検体について，複数種類のウイルス対策ソフトを用いてファイルを検査するオンラインサービス（VirusTotal）を利用して得た検知結果及び，M3AS を用いた不審ホストへの通信有無判定結果を表 3.6 に示す．なお，VirusTotal に関しては，複数種類のウイルス対策ソフトのうち一つでも検知したものを「VT 検知検体」と定義し，すべてのウイルス対策ソフトで検知していない，または VirusTotal に検体が登録されていないものを「VT 未検知検体」と定義した．ここで，解析環境にインストールしたソフトウェアが自動アップデートで通信する可能性のあるホスト（microsoft.com，windows.com，java.sun.com，adobe.com）を正規のホストとし，正規以外のホストを不審ホストとした．

なお，正規サイトが改ざんされていた場合，本評価では，本来は危険なサイトと判断すべきところを，誤って正規サイトとして分類している可能性がある．このため，表 3.6 に示す不審ホストへの通信検出結果（545 件）の値は、さらに多数となる可能性もある．

提案手法によって世の中で未知とされる検体（VT 未検知検体）を不審なホストへ通信した検体として検出することを確認した．

提案手法を用いた解析結果と，閉塞環境を用いた解析結果との比較を表 3.7 に示す．

表 3.6　マルウェア検知結果

| 項目 | VT 検知検体 | VT 未検知検体 | 総計 |
|---|---|---|---|
| 不審ホストへの通信あり | 309 | 236 | 545 |
| 不審ホストへの通信なし | 68 | 31 | 99 |
| 総計 | 377 | 297 | 644 |

表 3.7　閉塞環境での解析結果との比較

| 項目 | 提案手法 | 閉塞環境 |
|---|---|---|
| 不審ホストへ通信した検体 | 545 | 545 |
| （ダウンローダ型マルウェア） | (58) | (-) |
| HTTP コネクション数 | 17,092 | 16,719 |
| GET メソッド数 | 15,212 | 15,188 |
| POST メソッド数 | 1,880 | 1,531 |
| HTTP 通信先 | 449 | 396 |
| （マルウェア配布サイト） | (58) | (-) |
| インターネット接続数 | 58 | - |
| 実行ファイルダウンロード数 | 5 | - |

表 3.8　青木らの手法に基づく解析結果との比較

| 項目 | 提案手法 | 青木らの手法 |
|---|---|---|
| インターネット接続数 | 58 | 15,212 |
| 実行ファイルダウンロード数 | 5 | 5 |

提案手法を用いた解析結果と，青木ら[55]の手法に基づく解析結果の比較を表 3.8 に示す．なお，青木らの手法に基づく解析においては，HTTP の GET メソッドを HTTP ファイルダウンロード通信と判定し，インターネットと接続させた．

結果，多種環境マルウェア動的解析システムで不審ホストへの通信を観測した 545 検体のうち，58 検体がダウンローダ型マルウェアであることが，449 件の HTTP 通信先のうち，58 件がマルウェア配布サイトであることが判明した．また，提案手法の方が閉塞環境での解析よりも，多くの HTTP 通信先を観測した．さらに，青木らの手法に基づく解析と比較し，インターネット接続数は少ないものの，実行ファイルダウンロード数は同じ 5 つとなり，実行ファイルのダウンロード効率は高いことがわかった．なお，Snort を用いて提案手法のインターネット接続を監視し，外部への攻撃は発生しなかったことを確認した．これによって，攻撃者に加担していないことも確認した．

以上，マルウェア通信制御システムを用いることで，ダウンローダ型マルウェアや，当該マルウェアが通信するマルウェア配布サイトが特定できた．また，マルウェア配布サイトからマルウェアをダウンロードして解析することにより，今まで確認できなかった新たな不審ホストの情報を得ることも確認した．

## 3.6. 考察

(ア) 提案手法による外部への影響

例えば SQL インジェクション等は，HTTP の GET メソッドを利用した攻撃を行うため，提案手法では防ぐことができるが，青木らの手法に基づく解析では防ぐことができない．提案手法では，ダウンローダ型マルウェアが，第二のマルウェアを取得する挙動に着目し，当該 MW ダウンロード通信のみを許可している．このため，通信先は攻撃者の用意したマルウェア配布サイトに限られ，外部への攻撃は発生しない．

一方，MW ダウンロード通信のみをインターネットに接続する提案手法は，第二のマルウェアの一部をダウンロードできないという問題が発生する．

(イ) 第二のマルウェアダウンロード

評価実験では存在しなかったが，マルウェアの中には www.windowsupdate.com 等に接続し，疎通が確認できない場合は第二のマルウェアの取得を行わず，動作を停止するものも存在する[63]．このようなマルウェアに対して，青木ら[55]や Yoshioka ら[58]の手法では第二のマルウェアを取得できるが，提案手法では第二のマルウェアを取得することができない．

また，マルウェア配布サイトの中には，別のマルウェア配布サーバに通信をリダイレクトし，第二のマルウェアを配布するものも存在する．提案手法ではこのようなリダイレクトを行うマルウェア配布サイトに通信を行うダウンローダ型マルウェアも解析することができない．

さらに，マルウェアの中には，ダウンロードしたファイルのハッシュ値を，マルウェアの中にあらかじめ保持しているハッシュ値と比較し，ダウンロードの成否を確認するものも存在する[60]．このようなマルウェアは，疑似プログラム応答機能で生成した疑似プログラムを実行しないため，マルウェア通信制御システムではダウンローダ型マルウェアとして判定されず，第二のマルウェアを取得することができない．

これまで述べたように，提案手法では一部でダウンロードが行われない検体が存在する可能性がある．しかし，インターネットへの接続を許可する通信を MW ダウンロード通信に限定することで，他の手法よりもインターネット接続の制限が厳しくなる提案手法は，外部サイトへの攻撃が発覚すると社会的信用が失墜してしまう組織にとって，外部のサイトを攻撃しないという点で優れる．

(ウ) 提案手法と閉塞環境の差

評価実験では，提案手法の方が閉塞環境での解析よりも，多くの HTTP コネクション数を観測した．これは，提案手法がインターネットから第二のマルウェアをダウンロードし，当該ダウンロードしたマルウェアを解析環境上で実行したためである．なお，HTTP の通信先に関して，GET メソッドの増加分よりも，POST メソッドの増加分が多いことが確認できた．これは，ダウンロードした第二のマルウェアがホスト上の情報を収集し，HTTP サーバ上にアップロードしようとしていたためである．

(エ) インターネットからのダウンロード

評価実験で，インターネットに接続した通信は 58 件存在した．これら 58 件の通信先からのダウンロード結果を表 3.9 に示す．58 件の通信先のうち，5 件の通信先に関して実行ファイルがダウンロードできたが．HTML がダウンロードされたものが 36 件，応答がなかったものが 17 件存在した．これは，2 月に入手した検体に対して，マルウェア解析を 3 月に実施したため，時間経過により検体が削除されてしまったことによりダウンロードに失敗したと考えられる．検体を入手したすぐに解析を行えば，マルウェアのダウンロード成功率は向上すると考える．

表 3.9　ダウンロード結果

| ダウンロードファイル | 件数 |
|---|---|
| 実行ファイル | 5 |
| HTML | 36 |
| 応答なし | 17 |

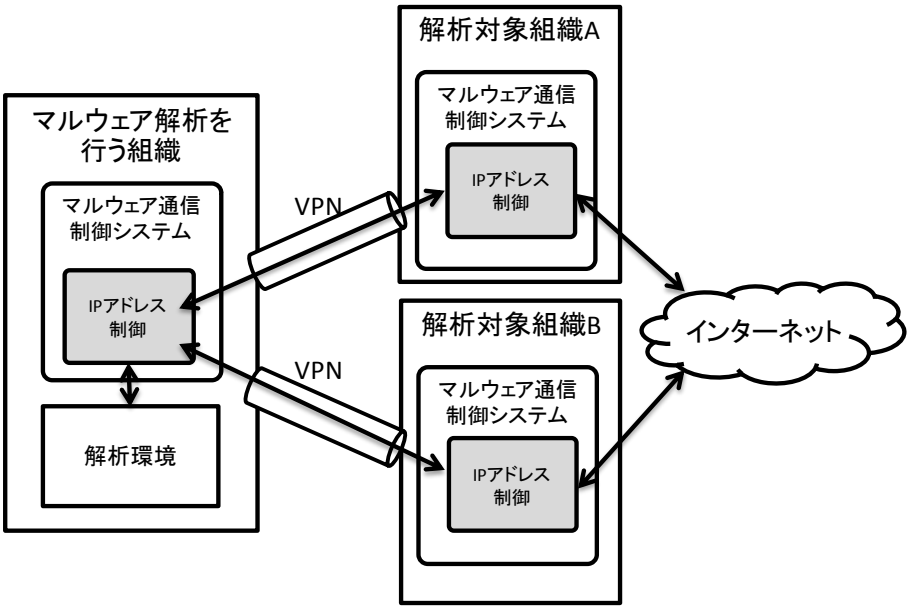



図 3.8　IP アドレス制御

(オ) クローキングへの対応

3.3.1 項で述べたように攻撃者は，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マルウェア解析を回避するクローキングという手法を利用することがある．このようなクローキングへ対応するには，マルウェア解析システムを攻撃対象の組織内に設置する必要がある．しかし，解析対象すべての組織内にマルウェア解析システムを設置するのはコスト面から困難である．このような攻撃を解析するために，マルウェア通信制御システムに，送信元の IP アドレスを制御し，攻撃対象の組織内ネットワークから通信する，IP アドレス制御機能を持たせることで解決する．

図 3.8 に，IP アドレス制御の仕組みを示す．マルウェア解析を行う組織と，解析対象のマルウェアを入手した組織が異なる場合に，これらの組織を VPN（Virtual Private Network）で接続し，マルウェアの MW ダウンロード通信を解析対象の組織からインターネットへ接続する．これにより，クローキングへも対応が可能となる．

## 3.7. 結論

本章では，ダウンローダ型マルウェアの MW ダウンロード通信を検出し，当該 MW ダウンロード通信のみをインターネットに接続させることにより，マルウェアによる外部への攻撃を抑制しつつマルウェア解析を行うマルウェア通信制御手法を提案し，提案手法を実装したプロトタイプシステムを開発した．また，代表的なダウンローダ型マルウェアを用いた評価実験により，提案システムによってダウンローダ型マルウェアを検知できることを確認した．さらに，実検体を用いた評価実験により，644 検体のうち，58 検体がダウンローダ型マルウェアであること，また，解析を通じて外部への攻撃が発生しなかったことを確認した．以上の結果より，本提案手法によって，外部への攻撃を行うことなく，組織に侵入したマルウェアの特性を解明できることを明らかにした．

## 参考文献


[51] 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, pp848-853, 2013

[52] Hitachi Solutions: 正規の Web サイトを改ざんしてウイルスを仕込む「Nine-Ball」攻撃に注意, 入手先＜http://securityblog.jp/news/757.html＞(参照 2016-11-24)

[53] Emurasoft: 今回のハッカーによる攻撃の詳細について, 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 2016-11-24)

[54] Google: Trends in Circumventing Web-Malware Detection, 入手先＜http://static.googleusercontent.com/media/research.google.com/ja//archive/papers/rajab-2011a.pdf＞(参照 2016-11-24)

[55] 青木一史, 川古谷裕平, 岩村誠 , 伊藤光恭: 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 (CSS2009) 論文集, pp 1-6, 2009

[56] S. Miwa, T. Miyachi, M. Eto, M. Yoshizumi, and Y. Shinoda : "Design and Implementation of an Isolated Sandbox with Mimetic Internet Used to Analyze Malwares," Proc. DETER Community Workshop on Cyber Security Experimentation and Test, 2007

[57] Kreibich C, Weaver N, Kanich C, Cui W, Paxson V. GQ: practical containment for measuring modern malware systems. In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pp. 397-412, 2011

[58] K. Yoshioka, T. Kasama, T. Matsumoto: Sandbox Analysis with Controlled Internet Connection for Observing Temporal Changes of Malware Behavior, The Fourth Joint Workshop on Information Security, 2009

[59] 情報通信処理機構: コンピュータウイルス・不正アクセスの届出状況および相談状況 [2015 年第 4 四半期 (10 月～12 月) ], 入手先＜https://www.ipa.go.jp/security/txt/2015/q4outline.html＞(参照 2016-11-24)

[60] 本城信輔: PC のウイルスを根こそぎ削除する方法, 技術評論社(2011).

[61] Aldo Cortesi: mitmproxy, 入手先＜https://mitmproxy.org/＞(参照 2016-11-24)

[62] Thomas Hungenberg, Matthias Echert: INetSim Internet Services Simulation Suite, 入手先＜http://www.inetsim.org/index.html＞(参照 2016-11-24)

[63] 情報通信処理機構: 脆弱性を利用した新たなる脅威の分析による調査, 入手先＜https://www.ipa.go.jp/files/000017747.pdf＞(参照 2016-11-24)

# 4. プロキシアクセス型マルウェア解析

## 4.1. 背景と目的

これまでも述べてきたように，マルウェアは既存のマルウェア対策ソフトでは検知できないため，組織へ侵入してしまうことを前提とした「多層防御型」のセキュリティ対策が重要となっている．これに対し，情報処理推進機構 IPA では多層防御の一環として出口対策に認証付きプロキシを設置することを推奨している[64]．しかしながら，プロキシを経由したインターネットアクセス機能を有するマルウェア（以降，プロキシアクセス型マルウェアと呼ぶ）が増加傾向にあり，さらには，2014 年頃より新型 PlugX[65]をはじめ，認証付きプロキシをも突破するマルウェアの出現も報告[66]されている．

本章では，プロキシ認証を突破するマルウェアを解析できるように M3AS を拡張したシステムを提案する．また，2014 年 10 月に取得したマルウェアのうち 629 種類のマルウェアを解析し，プロキシアクセス型マルウェアの数とその傾向を分析した結果について述べる．

## 4.2. 関連研究

前述した新型 PlugX や PoisonIvy のように，本来は認証情報で保護されているはずのプロキシを突破するマルウェアが出現しはじめており，それらの突破手法について分析，報告されている[66][67] [68]．ただしこれらの報告は，プロキシアクセスや認証突破手法の仕組みについて述べたものであり，その手法を実装したマルウェアの検出には触れられていない．一方で，前節で述べたように，IPA からは認証付きプロキシの設置による対策が推奨されており，本体策を突破するこれらのマルウェアの到達・侵入を把握することは，サイバー攻撃から組織を守らなければならない対策側にとって喫緊の課題である．

本章で述べる技術は，上記課題を解決する点で従来の研究とは異なる．

## 4.3. プロキシアクセス型マルウェア

本節ではプロキシアクセス型マルウェア出現までの歴史的背景と，プロキシ認証突破型マルウェアの認証情報窃取方法について述べる．

まず，プロキシアクセス型マルウェアをはじめとするネットワーク通信型マルウェアの攻撃手法の変遷について述べる．ネットワーク通信型マルウェアの出現当初，企業の外部にいる攻撃者が，攻撃対象の脆弱性を狙ってバックドアを仕掛け，ここから攻撃者との通信チャネルを開設し，攻撃対象の企業に対して直接的な攻撃を行っていた．これに対し，企業は，ファイアウォール（FW）を設置したり，ネットワークアドレス変換（NAT）したりするなどして，外部からの直接接続（インバウンド）を禁止して，企業からインターネット（アウトバウンド）への限られたサービス（メールやウェブ）のみ許可を与える対策を行ってきた．そこで攻撃者はボットや遠隔操作型マルウェアをメールやウェブ経由で企業に送付し，組織内の端末への感染，および感染端末が組織内部

からインターネットに接続して指令サーバとコミュニケーションをとるような仕組みを採用するようになった．さらに企業はこの対策としてプロキシを設置したり，プロキシの認証機能を設定したりすることによって，万が一マルウェアに感染したとしても，マルウェアが組織内部から容易に外部に出られないような出口対策を導入した．しかし，前述したように，プロキシでの認証機能による対策をも突破するマルウェアの出現が確認され，企業の最終砦を知らず知らずのうちに突破されている懸念が高まってきているのが現状である．日々多くのマルウェアが送られてくる企業等の組織にとって，現状の対策を破る能力を有するマルウェアを把握して，優先的に対処しなければならない．しなしながら，従来の手法ではマルウェアか否かを判定できたとしても，それがプロキシ認証を突破する能力を有するか否かを判定することはできなかった．

次に，プロキシ認証の仕組みについて述べる．企業の内部のネットワークと外部のネットワークとで通信する際，多くの企業ではプロキシサーバを介して，インターネットへのアクセス権限のあるユーザであるか否かの認証を行う．プロキシサーバを利用するには，プロキシサーバやディレクトリサーバでユーザ名とパスワードを設定するとともに，ブラウザには利用するプロキシサーバのアドレスを設定する．そして，ユーザがブラウザを使って外部の Web サイトにアクセスする際に出現する認証確認画面（図 4.1）で認証情報を入力することで，本人認証や権限確認を行い，外部 Web サイトへの通信を許可する．

コールバック通信[10]を行うマルウェアは，あらかじめ設定していた攻撃用のサイトと通信（チャネル）を確立させ，そのチャネルを使って遠隔操作を行う．ここで，上記で設定した認証の手段を講じることで，マルウェアは外部サーバへアクセスする際に認証情報を要求されるため，マルウェアは外部サーバへアクセスできずチャネルの確立を防止できる．これにより認証はコールバック通信を行うマルウェアの対策として有効とされてきた．しかし，前述の通り，最近のマルウェアにはこの認証情報を窃取して，プロキシ認証の際に窃取した認証情報をプロキシに認識させ，外部サーバ（C&C サーバ）へとアクセスしてしまうものも確認されている．ただし，プロキシ認証を突破するには，必ずしも認証情報の窃取が必要とは限らない．以降，このような能力を有するマルウェアを「プロキシ認証突破型マルウェア」と呼ぶ．

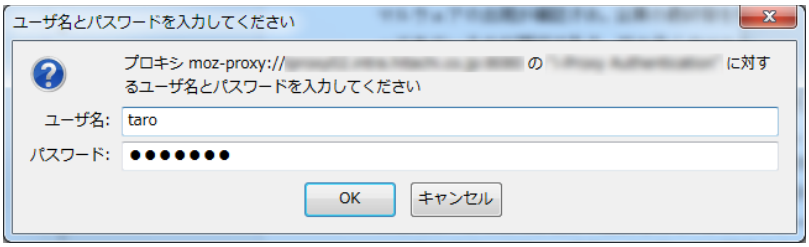



図 4.1　プロキシ認証確認画面（Firefox の例）

[10] 標的端末から攻撃端末へアクセスさせて，標的端末と通信する方式．FW（FireWall）等の導入によりインターネットから標的組織のイントラネットワークに直接接続できなくなったが，イントラネットワークからインターネットへは接続できることから，攻撃者は本方式を利用するようになった．

著者は認証付きプロキシを突破手法について言及されている解析報告[65][66][67][69][70]を参考に，突破方法を以下の 5 つの手法に整理した．

(ア) 認証情報格納ファイル・レジストリから窃取
(イ) キーロギング
(ウ) 画面取得による窃取
(エ) Internet Explorer（IE）へのコードインジェクション
(オ) ネットワーク盗聴

続けて，上記 5 つの認証付きプロキシ対策をマルウェアが突破する手法について述べる．

(ア) 認証情報格納ファイル・レジストリから窃取

マルウェアが PC 中のプロキシ認証格納ファイル・レジストリにアクセスし，プロキシ認証情報を盗み見る方法が考えられる．具体例としてはマルウェアが FindFirstFile/FindNextFile API を用いて，指定したディレクトリ内のプロキシ認証格納ファイルを走査する．

(イ) キーロギング

ユーザのキーボード入力を監視して，プロキシ認証情報を窃取する方法．キーボード入力を盗み取る方法としては以下の 2 種類がある．

i. システムフックを用いた方法

システムフックを用いたキーロギングでは, システム中の GUI プロセスに対して DLL を埋め込み，キーボード入力に伴うメッセージをフックする．フックされたメッセージは，マルウェアが用意した処理関数に入力される．

DLL の埋め込みには SetWindowsHookEx API を用いる．この API は様々なウィンドウメッセージに対するフック機構を提供する．キーロガーの多くがこの SetWindowsHookEx API を利用している．

ii. システムフックを用いない方法

システムフックを用いないキーロギングでは，GetAsyncKeyState API や AttachThreadInput API が用いられる．

GetAsyncKeyState API は特定のキー入力がされているかどうかを判定する API で，一定時間ごとにこの API を呼び出して各キーに適用することでキーロギングを行う．

AttachThreadInput は他スレッドへのウィンドウメッセージを取得する API で，別ウィンドウへのキー入力を取得する．

また，マルウェアは GetForegroundWindow API を用いて，ユーザがどのアプリケーションに対してキー入力を行っているかを調べることがある．

(ウ) 画面取得による窃取

マルウェアがデスクトップ画面を取得する際に用いる 2 つの API について述べる．一つ目は GetDC API で，これによりデスクトップのデバイスコンテキストを取得する．二つ目は BitBlt API で，GetDC API で取得したデバイスコンテキストの内容（画面データ）を別のデバイスコンテキストに出力する．その後，出力先のコンテキストの内容をファイル等に出力することで，デスクトップ画面の取得が完了する．通常，プロキシの認証情報の入力内容はマスキングされており，入力画面のみの窃取は困難であると考えられるが，認証情報入力の際に利用者がパスワード管理ソフトウェアを利用している場合，認証情報確認時に認証情報を映し出した画面情報が窃取される可能性がある．

(エ) Internet Explorer（IE）へのコードインジェクション

プロキシを突破するタイプのマルウェアが有する機能や動作の特性に関しては，様々なセキュリティベンダから調査レポートが公開されている[65][69]．新型 PlugX と呼ばれる RAT（Remote Administration Tool）では，プロキシ認証を突破する方法として Internet Explorer（IE）にインジェクションし，プロキシの認証情報を自動的に窃取する仕組みがある．これにより，IE を起動するたびに新型 PlugX が IE に対してコードインジェクションを仕掛け，4 つの API（HttpSendRequestA API，HttpSendRequestW API，HttpSendRequestExA API，HttpSendRequestExW API）にフックをかけることでプロキシの設定情報や認証情報を窃取する．

(オ) ネットワーク盗聴

新型 PlugX はネットワーク盗聴機能によりプロキシの認証情報を窃取する機能も具備する．ネットワーク盗聴により，プロキシ認証突破型マルウェアはプロミスキャスモードになってネットワークを盗聴し，HTTP の通信内容からプロキシの設定情報や認証情報を窃取する．具体的には，“Authorization: BASIC”に続く文字列（ユーザ名とパスワードが base64 エンコードされている文字列）をデコードして，ユーザ名やパスワードを取得し，それを新型 PlugX の設定に追加，再利用することで，プロキシの BASIC 認証を突破して C&C サーバと通信を行う．

次節では，上記機能を備えたプロキシ認証突破型マルウェアのプロキシ認証突破能力の有無を自動判定する手法について述べる．

## 4.4. プロキシ認証突破判定システム

本節では，プロキシ認証を突破するようなマルウェアが，実際にプロキシ認証を突破したか否かを判定するプロキシ認証突破判定手法およびシステムの概要を述べる．本システムでは，図 4.2 の通りマルウェアを M3AS（解析エンジン上のサンドボックス）で解析し，マルウェアの挙動を解析する．解析環境はプロキシを介して擬似インターネットへ接続する構成とする．

次に，プロキシによって保存されるアクセスログの一部を図 4.3 に示す．図中の丸で囲まれた数字はステータスコードと呼ばれ，Web アクセスや認証の成否に関する状態を示す．“200” はユーザ（マルウェア等のプログラムも含む）によるプロキシ認証が成功し，ユーザによって要求された HTML コンテンツ等の情報が Web サーバからユーザに正常に返されたことを示しており，プロキシを介して外部との接続確立が成功した（してしまった）ことを示す．またステータスコード “407” は認証が必要であることをプロキシ側がユーザ側に要求したことを示しており，ステータスコード”200”を伴わない”407”（つまり特定のユーザによる特定の Web サーバへのアクセス時に”407”のみプロキシログに残った場合）は，プロキシを介した外部との接続確立に失敗したことを示す．

提案手法では，あるマルウェアを解析した際に，そのマルウェアを M3AS で解析した解析時間帯（解析開始時間と終了時間）にプロキシに残されたアクセスログを基に，解析時間内にプロキシ認証が成功（ステータスが “200”）していることが確認されれば，そのマルウェアはプロキシ認証突破型マルウェアと判定する．

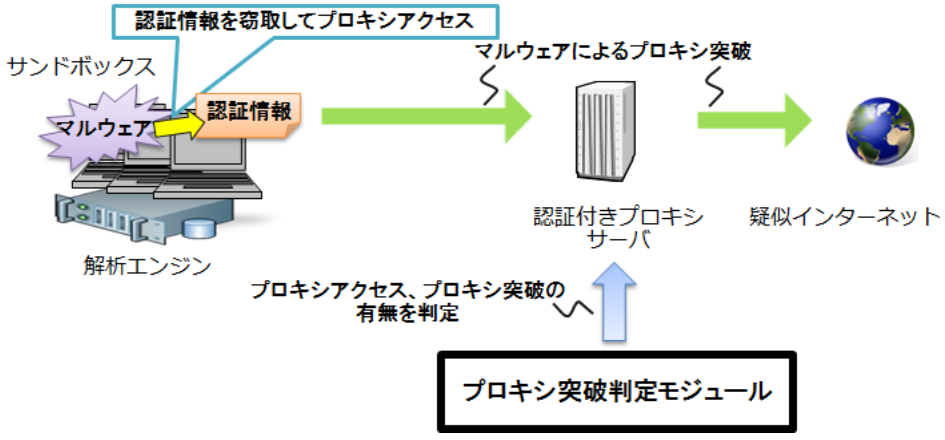



図 4.2　プロキシ認証突破判定システム

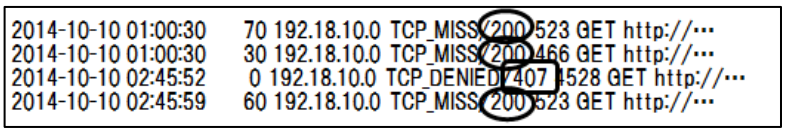



図 4.3　プロキシアクセスログ

マルウェアの認証付きプロキシ突破能力の有無を見極めることを目的としていることから，マルウェアがプロキシ認証を突破しやすい環境の準備が重要である．このため，4.3 節で述べた手段を用いて認証を突破するマルウェアが，認証情報を取得可能な環境をサンドボックスとして準備する．具体的には，M3AS が解析開始してサンドボックスを起動した直後に，自動的にブラウザを起動して，プロキシが要求してくる認証情報（ID とパスワード）をユーザが入力する行動を模擬したり，その認証結果をブラウザに保存させたりする．さらにネットワークを盗聴するマルウェアのために，認証情報をネットワーク上に敢えて平文で流すために，プロキシ認証には DIGEST 認証ではなく BASIC 認証方式を採用している[11].

## 4.5. 評価実験

本節では 2014 年 10 月の一ヶ月間に取得した 629 検体のマルウェアを用いて，プロキシ認証突破判定システムを評価し，プロキシアクセス型マルウェア（プロキシ認証突破型マルウェア，プロキシ利用マルウェア（プロキシ認証失敗）を含む）の分析結果とその考察を述べる．

なお，プロキシ認証突破判定結果の分析と考察にあたり，プロキシ認証が成功したマルウェアであっても，プロキシ認証に成功したアクセス先 URL が明らかに不正サイトでない通信先のみであった場合には（たとえば Windows の更新機能による WindowsUpdate サイトへのアクセスなど)，プロキシ認証突破判定の結果は「プロキシ認証突破」でなく「プロキシ利用（プロキシ認証失敗)」として分類した．

### 4.5.1. プロキシアクセス型マルウェアの検体数

１カ月間に取得した 629 種類の検体（マルウェア）にプロキシアクセス型マルウェアがどの程度の割合で存在するのかを把握するため，既存の M3AS を用いて，プロキシアクセスした検体（プロキシアクセス型マルウェア）の数と，プロキシアクセスしなかった検体（プロキシアクセスなしのマルウェア）の数を集計した．結果を表 4.1 に示す．

表 4.1　プロキシアクセス型マルウェアの検体数

| | 検体数 | 割合 |
|---|---|---|
| プロキシアクセス型 | 92 | 14.6% |
| プロキシアクセスなし | 537 | 85.4% |

[11] DIGEST 認証は，ID とパスワードをハッシュ関数 MD5 でハッシュ化して送信する．本方式では，クライアント端末とプロキシ間でハッシュ化した認証情報をやり取りすることになるため，盗聴や改ざんを防ぐことができる．本方式は，パケットを盗聴して認証突破を試みるマルウェアへの対策になるが，マルウェアの突破能力を試すという提案システムの目的においては不適となるため，本方式を採用すべきでない．

表 4.2　プロキシ認証突破型マルウェアの検体数

| | 検体数 | 割合 |
|---|---|---|
| プロキシ認証突破型（プロキシ認証成功） | 8 | 1.3% |
| プロキシ利用型(プロキシ認証失敗) | 84 | 13.3% |

次に，プロキシアクセス型マルウェア（92 検体）について，M3AS を拡張したプロキシ認証突破判定システムを用いて評価した．本システムは 4.4 節で述べたように，認証情報を取得可能なサンドボックスを用いる．

これにより，プロキシ認証突破した検体（プロキシ認証突破型マルウェア）を 8 検体確認した．

本評価では解析環境が外部ネットワークから隔離された（攻撃者から画面情報を盗み見ることができない）環境で行っている．このため，上記で確認した 8 種のプロキシ認証突破型マルウェアは全て，画面窃取以外の方法で認証情報を窃取するタイプのマルウェアである．

### 4.5.2. プロキシアクセス型マルウェアの検体数（拡張子別）

4.5.1 項で分類したプロキシ認証突破型マルウェア，プロキシ利用型マルウェア，プロキシアクセスなしマルウェアのそれぞれについて，各マルウェアのファイル形態（拡張子）の傾向を調査した．結果を表 4.3 に示す．

本調査の結果，プロキシ認証突破型マルウェアには exe 形式やマクロが埋め込まれた doc 形式，実行可能なファイルが埋め込まれた rtf 形式のファイルが特に多い傾向があることが分かった．また，拡張子が doc や exe のマルウェアはプロキシ利用型（プロキシ認証失敗）の数に対してプロキシ認証突破型のマルウェアが一定の割合で（それぞれ約 20%，約 10%）で存在したが，拡張子が rtf のマルウェアについてはプロキシ利用型（プロキシ認証失敗）14 検体に対してプロキシ認証突破型 0 検体という結果を得た．

表 4.3　プロキシアクセス型マルウェアの検体数（拡張子別）

| | 拡張子 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | bat | dll | doc | exe | pdf | jar | rtf | scr |
| プロキシ認証突破型 | 0 | 0 | 1 | 7 | 0 | 0 | 0 | 0 |
| プロキシ利用型<br>(プロキシ認証失敗) | 1 | 0 | 4 | 65 | 0 | 0 | 14 | 0 |
| プロキシアクセスなし | 1 | 1 | 4 | 410 | 1 | 1 | 113 | 6 |
| 検体の総数 | 2 | 1 | 9 | 482 | 1 | 1 | 127 | 6 |

### 4.5.3. VirusTotal データベースとの比較

プロキシ認証突破型マルウェアは認証を突破可能な最新の機能を具備しているため，プロキシ利用マルウェア（プロキシ認証失敗）やプロキシアクセスなしマルウェアよりもウイルス対策ソフトによる検知が難しい，という仮説を立て，これを検証した．

プロキシ認証突破型マルウェアのアクセス先 URL と，プロキシ利用マルウェア（プロキシ認証失敗）のアクセス先 URL を，VirusTotal のデータベースと照合し，VirusTotal データベース上で未登録なアクセス先 URL（つまり VirusTotal にとって未知の不正サイト）を調査した．その結果を表 4.4 に表す．

分析の結果，プロキシ認証突破型マルウェアのアクセス先 URL が VirusTotal で未登録であった（もしくは不正サイトとして判定されなかった）比率は全体の 64.7%，プロキシ利用マルウェアのアクセス先 URL の比率は 29.6%であり，仮説で示した傾向のあることを確認した．

表 4.4　プロキシ突破型マルウェアによるアクセス先 URL の VirusTotal 登録率

| 検体 | アクセス先 URL 数 | VirusTotal 未登録数（割合） |
|---|---|---|
| プロキシ認証突破型（プロキシ認証成功）（17 サイト） | 17 | 11 (64.7%) |
| プロキシ利用型（プロキシ認証失敗）（90 サイト） | 90 | 27 (29.6%) |

## 4.6. 考察

プロキシ認証突破能力を有するマルウェアの突破手法を検証するために，該当マルウェアのバイナリデータから文字列部分を抽出して追加調査を実施した．抽出した文字列の一部抜粋を図 4.4 に示す．本文字列から，マルウェアのバイナリファイル中に 4.2 節で述べたキーロギングに用いられる API（たとえば“GetAsyncKeyState API”，“GetForegroundWindow API”）が含まれていることが分かった．

## 4.7. 結論

本章では，多種環境マルウェア動的解析システム（M3AS）をベースにプロキシアクセス型マルウェアを解析するプロキシ認証突破判定システムを提案し，実装した．

本システムを用いて 2014 年 10 月からの 1 カ月間で取得したマルウェアを解析し，解析した全 629 検体のマルウェアのうち 84 検体がプロキシ利用するマルウェアで，8 検体がプロキシ認証突破するマルウェアであることを確認した．本システムを用いることにより，IPA より推奨されている認証付きプロキシをも突破してしまう能力を有するマルウェアの自動解析および特定が可能となる．この結果を用いて優先対処すべきマルウェアの選定に役立てることができ，インシデント対応の効率化を図ることが期待できる．

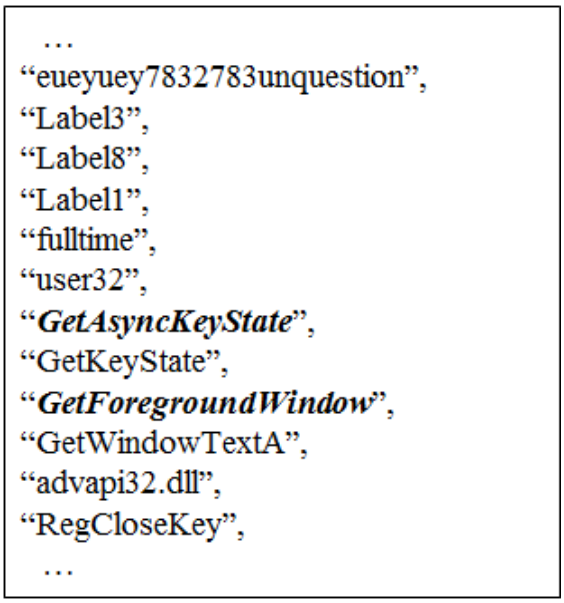



図 4.4　M3AS 解析ログ（抜粋）

## 参考文献


[64] 情報通信処理機構：「高度標的型攻撃」対策に向けたシステム設計ガイド,
入手先＜http://www.ipa.go.jp/files/000042039.pdf＞(参照 2016-11-24)

[65] IIJ：新型 PlugX の出現，2013 年 11 月，入手先＜https://sect.iij.ad.jp/d/2013/11/197093.html＞(参照 2016-11-24)

[66] JPCERT/CC：認証プロキシに対応した PoisonIvy(2015-07-08),
入手先＜https://www.jpcert.or.jp/magazine/acreport-poisonivy.html＞(参照 2016-11-24)

[67] Cobalt Strike Blog: HTTP Proxy Authentication for Malware，入手先＜http://blog.cobaltstrike.com/2014/06/25/http-proxy-authentication-for-malware/＞(参照 2016-11-24)

[68] Michael Sikorski , Andrew Honig : Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press, 2012

[69] トレンドマイクロ： 標的型攻撃に利用される PlugX の脅威とは，入手先＜http://about-threats.trendmicro.com/relatedthreats.aspx?language=jp&name=Pulling%20the%20Plug%20on%20PlugX＞(参照 2016-11-24)

[70] 情報通信処理機構：「標的型サイバー攻撃対策」，2014 年 2 月，入手先＜http://www.ipa.go.jp/security/event/2013/isec-semi/documents/2013videosemi_targeted_cyber_attacks_v1a.pdf＞(参照 2016-11-24)

[71] マクニカネットワークス：既存セキュリティ対策をすり抜ける標的型攻撃を検知し適切な防御策を支援，入手先＜http://diamond.jp/articles/-/19410?page=2＞(参照 2016-11-24)

[72] すべてわかるセキュリティ大全―基礎知識から最新の攻撃手法や対策まで，日経 BP 社，2014

[73] 三井物産セキュアディレクション：サイバーセキュリティ事件簿,
入手先＜http://www.mbsd.jp/casebook/20130212.html＞(参照 2016-11-24)

[74] VirusTotal：入手先＜https://www.virustotal.com/ja/＞(参照 2016-11-24)

[75] NISC：標的型攻撃等の脅威について，入手先＜http://www.nisc.go.jp/conference/suishin/ciso/dai18/pdf/2.pdf＞(参照 2016-11-24).

# 5. サンドボックス最適化

## 5.1. 背景と目的

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が，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が増えている．このような背景から，著者は，前章までに述べてきたように，最新のサイバー攻撃の性質をいち早く把握し，防御に生かすことを目的として， M3AS の提案を行ってきた．本技術では，多種環境のサンドボックス上でマルウェアの並列解析を実施することで，動作環境に応じて挙動を変化させるマルウェア（環境選択型マルウェア）の挙動解明が可能である．マルウェアを多種環境上で並列解析するシステムにおいては，サンドボックスの構築コストとの兼ね合いから，並列解析を行うサンドボックスの個数と検知精度との兼ね合いが重要である．そこで本章では，M3AS の保有するサンドボックス 77 種類に対して，構成サンドボックス数と検知精度に関する分析を行い，検知精度を維持しながらサンドボックス数を削減できる可能性について検証する．

## 5.2. 関連研究

環境選択型マルウェアの解析にあたり，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その挙動が明らかにできないという問題があった．この問題に対して，複数種類のサンドボックスによりマルウェアを解析する技術[22][31][32]は存在するが，サンドボックスの構成は利用者（解析者）の判断に任されており，数と精度との関係を明らかにして示してサンドボックス最適化にアプローチした研究はない．

## 5.3. 環境選択型マルウェアの抽出

どのような環境でも動作するマルウェアであればサンドボックスは 1 種類で十分であるが，実際には特定の環境でしか動作（顕現）しない環境選択型マルウェアが多数存在する．そこで環境選択型マルウェアの存在を確認するため，ここでは M3AS を用いた環境選択型マルウェアの抽出方法を述べるとともに，本手法をもちいて環境選択型マルウェアを抽出した結果についても述べる．

2 章で述べたように M3AS は，M3AS を構成する各サンドボックスからマルウェアの挙動を観測して，それらの結果を出力する．2.1.6 項で定義したように M3AS では，被解析検体がインターネット上の不審な接続先（あらかじめ定めたホワイトリストのいずれにも合致しない通信先）へ通信を行う挙動をマルウェアの挙動とみなし，この挙動がみられたマルウェアを顕現マルウェアと呼ぶ．

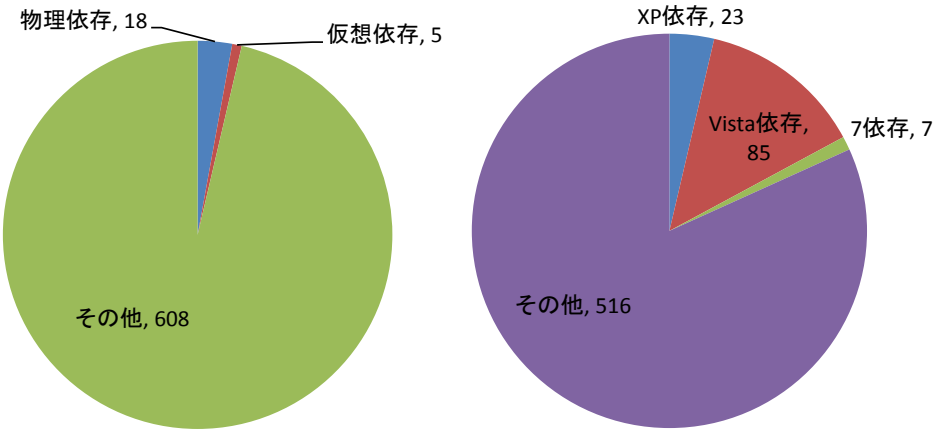



図 5.1　物理/仮想環境依存性検体（左）と OS 依存検体（右）の分布

また，本章では，環境選択型マルウェアの判定基準に，環境ごとに導出するピアソンの相関係数を用いる．M3AS を構成するサンドボックス集合に対して，あるマルウェアの動作有無と，サンドボックスの環境（例えば OS の種類，アプリケーションインストールの有無等）との相関性をピアソンの相関係数により求め，その値が 0.4 を上回った場合に，そのマルウェアを環境選択型マルウェアと判定する．

この環境選択型マルウェア判定方法を用いて，2 章で扱ったマルウェア検体 631 件を判定した結果，図 5.1 に示す結果となった．動作環境が，物理環境，もしくは仮想環境に依存する検体を 23 検体，また，いずれかの OS 環境への依存性が確認された検体を 115 検体抽出した．このように，プラットフォームやソフトウェア環境に依存した環境選択型マルウェアが少なからず存在する．

## 5.4. サンドボックス選定手法の提案

M3AS では，様々な環境選択型マルウェアの挙動を解明するために，多種類の環境をサンドボックス上で構築しており，その種類は 76 種にも及ぶ．M3AS を構成するサンドボックスの動作環境は，解析エンジンとして Threat Analyzer および Cuckoo Sandbox をもつなど，2 章に示した表 2.1 に示すとおりである．

M3AS を構成するサンドボックスの効率的な選定手法を考えるにあたり，M3AS の各サンドボックスのマルウェア挙動解析結果を 3,000 件超取得した．M3AS を構成する各サンドボックスで顕現したマルウェアの数を表 5.1 に示す．ここで顕現率を以下に定義する．

顕現率　＝　解析して顕現したマルウェアの数／解析した検体×100

M3AS を用いてこれまでに解析した検体の中で，不審なネットワークへの通信を検知した検体の総数は 2,117 件であり，表 5.1 の第 3 列には，各サンドボックスにおける顕現マルウェア数 2,117 件に対する顕現率を示している．これにより最も検知率の高いサンドボックスは単体で 2,117 件のうち 50%程度を顕現可能であることがわかる．

M3AS を実際の企業等に導入するにあたっては，構築時のコスト（CAPEX[12]）と運用時のコスト（OPEX[13]），および予算の都合によりサンドボックスの規模に制約を受ける．このため，より少ない数のサンドボックスで構成することが求められることもある．

本節では，より少ないサンドボックスの組合せでも高い顕現率が得られるようなサンドボックスの選定を試みる．

表 5.1　サンドボックス毎のマルウェア顕現率

| サンドボックス ID | 顕現 MW 数 | 顕現 MW 率 |
|---|---|---|
| #1 | 1,146 | 54.13% |
| #2 | 1,145 | 54.09% |
| #3 | 1,132 | 53.47% |
| #4 | 1,131 | 53.42% |
| #5 | 1,128 | 53.28% |
| : | : | : |
| #73 | 196 | 9.26% |
| #74 | 167 | 7.89% |
| #75 | 147 | 6.94% |
| #76 | 1 | 0.05% |

[12] CAPital EXpenditure（資本的支出，設備投資）
[13] OPErating EXpenditure（運用維持費，運用コスト）

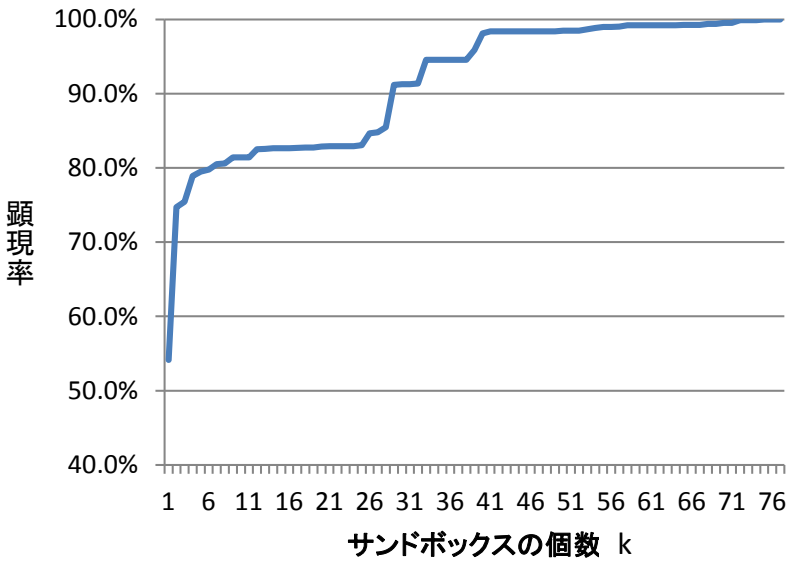



図 5.2　顕現率とサンドボックス数の関係

表 5.1 の結果を用いて，単純に顕現率の高いサンドボックスを順（降順）に選択して組合せを作成する方式「①顕現率順選択方式」を提案する．検知率順選択方式でサンドボックスの組合せを選定した際の，選定サンドボックスの個数に対する顕現率の推移を図 5.2 に示す．

図 5.2 より，①顕現率順選択方式では，サンドボックスの個数に対する全体顕現率の増加の停滞が断続的に発生している事象を確認できる．これは，類似のサンドボックス間では顕現する検体の種類に重複が見られるため，単純に顕現率順に選択した場合には類似のサンドボックスが連続して選択されてしまい，累計した顕現マルウェア数が増加しないことに起因する．そこで，これらの問題を解決するサンドボックスの組合せ選定の方法として，「②最大顕現器選択方式」と「③重複排除選択方式」を新たに提案する．

②最大顕現器選択方式は，顕現率の高いサンドボックスを優先的に選ぶという方針のもと，他のサンドボックスとの顕現マルウェアの重複を考慮するよう，前述した①顕現率順選択方式を改良した方式である．単純に顕現マルウェア数の大きなサンドボックスを順に選ぶ手法に対して，選択済のサンドボックスのいずれでも顕現していないマルウェアに着目し，それらマルウェアをより多く顕現可能なサンドボックスを優先的に選定する．

③重複排除選択方式は，各サンドボックスにおいて，固有に顕現可能な検体を多く持つサンドボックスに着目し，これを優先的に選定する．顕現可能な全検体を漏れなくカバーすることを前提とする場合は，③重複排除選択方式がより効率的にサンドボックスを選定できる．

以降に各方式におけるアルゴリズムの内容とその評価結果を述べる．なお，アルゴリズムの説明にあたっては表 5.2 に示す表記記号を用いる．

表 5.2　表記記号

| 記号 | 定義 |
|---|---|
| U | マルウェアの全体集合. |
| $m_i$ | マルウェア. $i$ はマルウェアの序列番号を表す. $(\mathrm{U} = \{m_1, \cdots, m_{2117}\})$ |
| $s_j$ | サンドボックス. $j$はサンドボックスの序列番号を表す. |
| $M_j$ | $s_j$によって顕現可能なマルウェアの集合. |
| $\lvert M \rvert$ | 集合$M$の要素の個数. たとえば，$\lvert M_j \rvert$は$s_j$によって顕現可能なマルウェアの個数を表す. $(0 \leq \lvert s_j \rvert \leq 2117)$ |
| $Y$ | 選定したサンドボックスの集合. |

(ア)②最大検知器選択方式のアルゴリズム

1. $\mathrm{n} = 1$，$\mathrm{A}^{(1)} = \mathrm{U}$，$\mathrm{Y}^{(1)} = \emptyset$
2. $\mathrm{j}^* = \underset{j \in [1,\cdots,76]}{\arg\max} \left| M_j \cap \mathrm{A}^{(n)} \right|$
3. $\mathrm{Y}^{(n+1)} = \mathrm{Y}^{(n)} \cup \mathrm{s}_{\mathrm{j}^*}$
4. $\mathrm{A}^{(n+1)} = \mathrm{A}^{(n)} - \mathrm{M}_{\mathrm{j}^*}$
5. もし、$\mathrm{A}^{(n+1)} = \emptyset$であれば終了し，$\mathrm{Y}^{(n+1)}$ を出力
6. $n = n + 1$ して Step. 2 に戻る.

(イ)③重複排除選択方式のアルゴリズム

1. $\mathrm{n} = 1$，$\mathrm{A}^{(1)} = \mathrm{U}$，$\mathrm{Y}^{(1)} = \emptyset$
2. $M_k \in \mathrm{A}^{(1)}$ に対し、$\mathrm{j}^* = \underset{j \in [1,\cdots,76]}{\arg\max} \left| M_j \cap \overline{\bigcup_{j \neq k} \mathrm{M}_k} \right|$
3. $\mathrm{Y}^{(n+1)} = \mathrm{Y}^{(n)} \cup \mathrm{s}_{\mathrm{j}^*}$
4. $\mathrm{A}^{(n+1)} = \mathrm{A}^{(n)} - \mathrm{M}_{\mathrm{j}^*}$
5. もし、$\mathrm{U}' = \emptyset$であれば終了し，$\mathrm{Y} = \{x(1), \cdots, x(n)\}$ を出力
6. $n = n + 1$ して Step. 2 に戻る.

## 5.5. 評価実験

②最大顕現器選択方式と③重複排除選択方式とをそれぞれ用いて，サンドボックスの組合せを決定した結果を図 5.3 に示す．なお比較対象として，5.4 節で述べた，①顕現率順選択方式を用いた場合の全体顕現率の推移も同図に示す．②最大顕現器選択方式では，30 個のサンドボックスで全 2,117 検体の顕現を確認した．一方，③重複排除選択方式では，27 個のサンドボックスで全 2,117 検体が顕現し，②最大顕現器選択方式と比べて 3 つ少ないサンドボックスの組合せとなった．なお，サンドボックスの数が少ない場合では，③重複排除選択方式による顕現率が②最大顕現器選択方式を下回る結果となった．

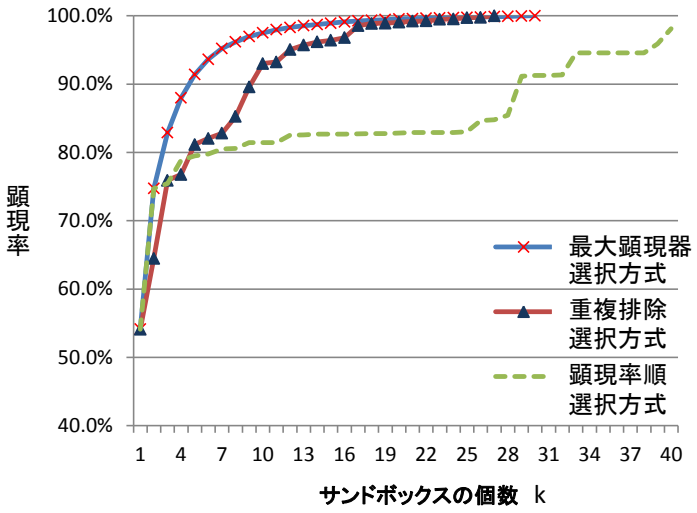



図 5.3　提案手法による全体顕現率の評価

## 5.6. 考察

②最大顕現器選択方式は，サンドボックスの数が少ない場合に顕現率の大きな増加を示すものの，残存する未顕現マルウェアの数が減少するにしたがって，顕現率の増加率は単調に減少する．このため②最大顕現器選択方式は，すべての検体を漏れなく顕現させることを前提とせず，限られたサンドボックスで高い顕現率を持つ組合せを選定する場合に有効である．

③重複排除選択方式の場合，②最大顕現器選択方式に対してサンドボックスの組合せ個数に対する顕現率の増加の程度が不安定であった．これは本方式が，単純に顕現数の大きなサンドボックスを順番に選ぶ方法ではないことに起因する．さらに③重複排除選択方式においては，固有の顕現マルウェアを持つサンドボックス（一部のサンドボックスでしか顕現しない検体を顕現させる能力のあるサンドボックス）がより早い段階で選択される．このために，②最大顕現器選択方

式，③重複排除選択方式の双方による選択サンドボックスの組合せからそれぞれ上位のみを抽出した場合，前者に対し後者の方が，より多くの環境選択型マルウェアを検知するサンドボックス群となる．

本章で実施した②重複排除選択方式の評価では，あるマルウェア解析時にサンドボックスのエラーが発生した場合でも，顕現していないサンドボックス（顕現なし）として扱った．これは，あるマルウェアが固有に顕現するサンドボックスを選定する際に，誤った選定となる原因となりうる．このため「解析エラー」と「顕現なし」とを区別して選定方法を評価することが必要となる．③重複排除選択方式では，他のサンドボックスでは顕現しなかった固有の顕現マルウェアを持つサンドボックスを選定し，それらによって顕現したマルウェアを全体から取り除くという処理を繰り返しているが，前述のサンドボックスが存在しない場合は，アルゴリズムが破綻してしまう可能性がある．いずれのサンドボックスにも固有の顕現マルウェアが存在しない場合は，顕現したサンドボックスの種類数が最も少ないマルウェアについて，それを顕現したサンドボックスを選定する，といった方針での改良が考えられる．また，本章で実施したサンドボックス選定の評価では，これまでに M3AS で顕現が確認されたマルウェアを標本としたため，本選定方法によって構成されたサンドボックスで今後現れる検体を 100%検知できるとは限らない．このため，マルウェアのトレンドに合わせて逐次評価してサンドボックスの再選定をしていく必要がある．

## 5.7. 結論

2 章で述べた M3AS について，構成するサンドボックスの個数に対する顕現率をより大きくするサンドボックス選定方法の策定に取り組み，複数のサンドボックス間での顕現マルウェアの重複を考慮した，サンドボックスの選定方式を 2 方式提案した．まず②最大顕現器選択方式は，少ないサンドボックス数でより顕現率の高いサンドボックスの組合せを選定する目的で利用するのに有用であることを確認した．次に③重複排除選択方式は，②最大顕現器選択方式よりも少ない数のサンドボックスで全検体を顕現させることができる手法として有効であることを確認した．

評価の結果，76 個のサンドボックスで構成される M3AS によってこれまでに解析した中に，顕現した検体（不正なネットワーク通信を行う検体）は 2,117 件存在したが，これらの顕現マルウェアのすべてを顕現させるのに，②最大検知器選択方式の場合は合計 30，③重複排除選択方式の場合は合計 27 のサンドボックスがそれぞれ必要であることが分かった．このため，③重複排除選択方式を採用した場合，M3AS を構成するサンドボックスの数を，65%削減できることを確認した．

# 6. 不正サイト挙動解明

## 6.1. 背景と目的

これまでに標的型攻撃の事例としてあげてきた日本年金機構へのサイバー攻撃では，マルウェアの感染にドライブバイダウンロード（Drive-by download）[76]という手法が利用された[77]．ドライブバイダウンロードとは，Web ブラウザなどを介して，ユーザに気付かれないようにマルウェアをダウンロードさせる手法である．

ドライブバイダウンロードによる攻撃は，主に Web ブラウザや，OS，その他のサードパーティ製のソフトウェアの脆弱性を突いて行われることが多い．このため，ある解析環境で不正サイトにアクセスしたとしても，攻撃者が狙っている脆弱性が存在しない解析環境ではマルウェア感染が起きず，不正サイトの解析を行えないといった課題が存在する．最近では，javascript を用いたブラウザフィンガープリントによってクライアント環境に応じた攻撃を仕掛ける不正サイトも確認されている[78]．M3AS は，マルウェアという実体のあるファイルの解析できるが，Web サイトのようなファイルの実体のない URL 等の情報は，サンドボックスでそのまま実行することができないため解析ができなかった．

そこで本章では，2 章で述べた M3AS を拡張することで，多種環境を用いた不正サイト解析システムを開発し，環境に応じて応答を変化させる不正サイトの解析を行う．また，実際の不正サイトを用いて，このような不正サイトの解析に有効に動作する解析環境を考察する．さらに，実環境を用いた評価実験により，多種環境を用いた不正サイト解析の有効性を示す．

## 6.2. 関連研究

不正サイトの解析を行う手段として，クライアントハニーポットに関する研究が存在する．クライアントハニーポットは，Web ブラウザ，または，それを模擬したシステムにより不正サイトにアクセスすることで不正サイトを解析する技術である．実際のブラウザを用いたクライアントハニーポットを高対話型[79][80]と呼び，ブラウザをエミュレートして解析を行うクライアントハニーポットを低対話型[81][82]と呼ぶ．高対話型クライアントハニーポットの方が実際の環境を用いることで難読化処理を行うような攻撃に対しても攻撃の検知が可能であるが，網羅的な解析を行うためには，攻撃対象となる様々な解析環境を用意する必要があり，それだけ多くのリソースが必要となってしまう．

高対話型クライアントハニーポットの方が実際の環境を用いることで難読化処理を行うような攻撃に対しても攻撃の検知が可能であるが，環境に応じて応答を変化させる不正サイトの解析を行うためには，攻撃対象となる様々な解析環境を用意する必要があり，それだけ多くのリソースが必要となる．

## 6.3. 不正サイト解析システムの提案

本章では，環境に応じて応答を変化させる不正サイトを解析するため，様々な解析環境を用いて不正サイトに接続し，不正サイト解析に適した解析環境の組み合わせを見つける．

ドライブバイダウンロードなどに用いられる不正サイトを解析する不正サイト解析システムは，攻撃者が攻撃に利用する脆弱性を備えた複数の解析環境を用いなければならない．このため，様々な OS やアプリケーションを備えた解析環境を構築し，不正サイトの解析に用いる．なお，不正サイト解析システムに実装する解析環境の構成については 6.3.1 項で検討する．不正サイト解析システムの機能構成は，基本的に 2 章で述べた図 2.1 および図 2.2 と同様であるが，図 6.1 に示すように入力する検体がファイルではなく，Web サイトの URL である点，そしてネットワーク再現機能が存在せずインターネットに直結している点で異なる．

不正サイト解析システムの具体的な処理の流れを説明する．

1. 解析者は，不正サイト解析システムの URL 投入画面を通じて URL 振り分け機能に解析対象の URL を投入する．
2. URL 振り分け機能は，投入された URL を Web ブラウザ挙動観測機能の複数の解析エンジンに振り分けて投入する．
3. 動的解析エンジンは，多種多様な解析環境（サンドボックス）において，Web ブラウザを起動し，指定された URL に対して通信を行う．
4. 動的解析エンジンは，解析ログを観測ログ分析機能に集約する．
5. 観測ログ分析機能は，複数の解析環境から得た観測ログを分析し，解析者に解析結果を提供する．

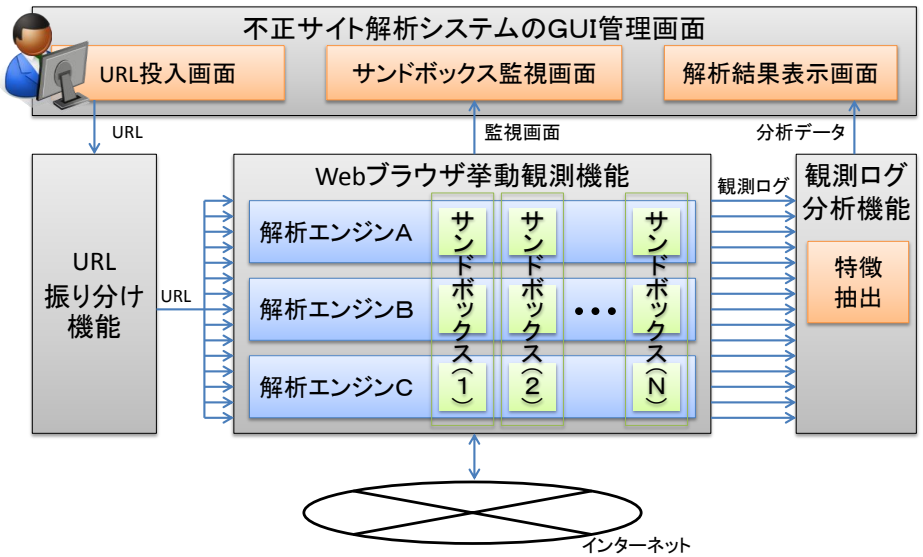



図 6.1　機能構成と URL 解析の流れ

以上のように，不正サイト解析システムは，複数の動的解析エンジン，解析環境を用いて不正サイトの解析を行う．

### 6.3.1. 解析環境の設計

6.2節で述べたように,高対話型クライアントハニーポットである不正サイト解析システムは,攻撃対象となる様々な解析環境を用意する必要があり,多くのリソースが必要となる.本項では,この課題を解決するために，効率的な解析環境の組み合わせについて検討する．

(ア) 方針

解析の成功率を上げるためには，攻撃を受けやすい解析環境を構築する必要がある．攻撃者は世の中で普及しているOSやアプリケーションを攻撃対象として選ぶ可能性が高いため，世の中で普及しているOSやアプリケーションを解析環境の候補とする．さらに，多くの脆弱性を持つ解析環境ほど攻撃が成功する可能性が高まる．つまり，多くの脆弱性を持つ解析環境を構築することで攻撃の成功率を向上させることが期待できるため，OSやアプリケーションが持つ脆弱性の数に着目し，解析環境候補の絞込みを行う．続いて，実際の不正サイトに接続して攻撃者が攻撃対象としている環境を調査し，解析環境候補の組み合わせについて検討する．

なお，以下の理由により，2章で述べたM3ASのサンドボックス環境（2.2.2項）の構成とは異なっている．

- M3ASの解析エンジンの1つであるThreat Analyzerが2015年以降に新しいOS（Windows8）に対応したこと
- 不正サイト解析システムは環境選択型マルウェア（ファイル）の挙動の解明が目的ではなくURLアクセス時のブラウザの挙動の解明が目的であることからブラウザのバリエーションの確保が重要であること

(イ) 普及率の調査

OSに関しては，企業で利用されるOSの普及状況[83]を考慮し，Windowsを対象に解析環境を候補とする．表 6.1に解析環境候補のOSを示す．また，Webブラウザに関しては，普及状況[84]を考慮し，Internet Explorer(IE)，Firefox，chrome，Safariを対象に解析環境を候補とする．表 6.2に解析環境候補のWebブラウザを示す．

表 6.1　解析環境候補（OS）

| OS 種類 | バージョン |
|---|---|
| Windows XP | SP なし，SP1，SP2，SP3 |
| Windows Vista | SP なし，SP1，SP2 |
| Windows 7 | SP なし，SP1 |
| Windows 8 | SP なし |
| Windows 8.1 | SP なし |

表 6.2　解析環境候補（Web ブラウザ）

| Web ブラウザ種類 | バージョン |
|---|---|
| Internet Explorer | ver.6～ver.11 |
| Firefox | ver.0.1～ver.34 |
| chrome | ver.0.1～ver.40 |
| Safari | ver.1～ver.8 |

(ウ) 脆弱性情報の調査

(イ)に挙げた解析環境候補に対して，2012 年～2014 年に脆弱性情報データベース（NVD）[85]に公開された脆弱性情報 17,804 件の中から，CVSS[86]基本値の深刻度レベル High の脆弱性を調査した．

ここでは OS を例にとり，解析環境の優先順位の考え方について説明する．図 6.2 は，OS の脆弱性情報の集合を表した図である．まず，最も多くの脆弱性を持つ OS A を優先順位の 1 番目とする．次に OS の脆弱性の集合から優先順位 1 で選択された OS が持っている脆弱性の集合を除いた脆弱性集合のなかで，最も多くの脆弱性を持つ OS B を優先順位の 2 番目とする．以下同様に優先順位付けを行う．

上記の方法で算出した OS の優先順位を表 6.3 に，OS と同様にして算出した Web ブラウザの優先順位を表 6.4 に示す．なお，Web ブラウザの優先順位は 5 番までを表示した．ここで，脆弱性を識別するための規格である CVE[38]を用いて CVE カバー数と CVE カバー率を定義する．CVE カバー数は，これまでに選択された OS または Web ブラウザが持つ脆弱性の累計数を表し，CVE カバー率は，OS または Web ブラウザに存在する脆弱性の数に対する CVE カバー数（CVE カバー数／OS または Web ブラウザに存在する脆弱性の総数）の割合を表す．なお，本調査では，表 6.1 に示す OS に存在する脆弱性の総数は 185 件，表 6.2 に示す Web ブラウザに存在する脆弱性の総数は 1399 件であった．

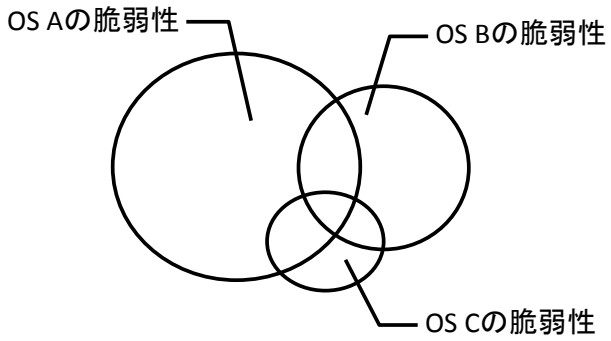



図 6.2　OS 選択の優先順位

表 6.3　　OS の優先順位

| 優先順位 | OS | CVE カバー数 | CVE カバー率 |
|---|---|---|---|
| 1 | Windows Vista (sp2,x64) | 142 | 76.76% |
| 2 | Windows XP (sp3,x86) | 167 | 90.27% |
| 3 | Windows 7 (sp1,x86) | 179 | 96.76% |
| 4 | Windows Vista (sp2,x86) | 184 | 99.46% |
| 5 | Windows Vista (-,x86) | 185 | 100.00% |

表 6.4　　Web ブラウザの優先順位

| 優先順位 | Web ブラウザ | CVE カバー数 | CVE カバー率 |
|---|---|---|---|
| 1 | Internet Explorer 9 | 260 | 18.58% |
| 2 | Firefox 4 | 515 | 36.81% |
| 3 | Safari 4 | 601 | 42.96% |
| 4 | chrome 25 | 635 | 45.39% |
| 5 | Firefox 19 | 728 | 52.04% |

表 6.5　ユーザエージェント

| UA-ID | OS | Web ブラウザ | ユーザエージェント |
|---|---|---|---|
| 1 | Windows XP (sp3,x86) | Internet Explorer 7 | Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0) |
| 2 | Windows XP (sp3,x86) | Internet Explorer 8 | Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0) |
| 3 | Windows XP (sp3,x86) | Firefox4 | Mozilla/5.0 (Windows NT 5.1; rv:2.0) Gecko/20100101 Firefox/4.0 |
| 4 | Windows XP (sp3,x86) | Safari4 | Mozilla/5.0 (Windows; U; Windows NT 5.1; ja-JP) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17 |
| 5 | Windows XP (sp3,x86) | Chrome25 | Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.19 (KHTML, like Gecko) Chrome/25.0.1323.1 Safari/537.19 |
| 6 | Windows Vista (sp2,x64) | Internet Explorer 7 | Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; WOW64; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.30729) |
| 7 | Windows Vista (sp2,x64) | Internet Explorer 8 | Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.30729) |
| 8 | Windows Vista (sp2,x64) | Internet Explorer 9 | Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; WOW64; Trident/5.0) |
| 9 | Windows Vista (sp2,x64) | Firefox4 | Mozilla/5.0 (Windows NT 6.0; WOW64; rv:2.0) Gecko/20100101 Firefox/4.0 |
| 10 | Windows Vista (sp2,x64) | Safari4 | Mozilla/5.0 (Windows; U; Windows NT 6.0; ja-JP) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17 |
| 11 | Windows Vista (sp2,x64) | Chrome25 | Mozilla/5.0 (Windows NT 6.0; WOW64) AppleWebKit/537.19 (KHTML, like Gecko) Chrome/25.0.1323.1 Safari/537.19 |
| 12 | Windows 7 (sp1,x86) | Internet Explorer 9 | Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) |

(ア) 不正サイト応答の調査

実際の不正サイトに複数の環境で接続し，応答の有無から不正サイト解析システムに構築する解析環境を検討する．ここでは，2013 年 6 月から 2016 年 1 月の期間に収集した不正な接続先（2,439 件）に対して応答の有無を調査した．不正サイト応答の調査に用いるユーザエージェントは，(ウ)の脆弱性調査結果及び，これまでのマルウェア解析の実績をもとに選定した．ユーザエージェントの情報を表 6.5 に示す．

2,439 件の不正サイトのうち，応答があった不正サイトは 318 件（応答率 13%（318 件/2,439 件））であった．表 6.6 に，ユーザエージェント毎に応答があった不正サイトの数を示す．応答のあった不正サイトのうち、取得したコンテンツの MD5 ハッシュ値が異なる不正サイトは 86 件（応答の変化率 27%（86 件/318 件））存在した．表 6.7 に，ユーザエージェントの違いにより取得したコンテンツのサイズに顕著な差が見られた不正サイトを示す．

Case 644 及び Case1054 は，Internet Explorer（7, 8, 9）から，Case1120 は，Internet Explorer (7, 8)からリクエストに対してのみコンテンツを応答している．

以上の検討結果より得られた解析環境の優先順位を表 6.8 に示す．OS 選択の優先順位と同様，最も多くの応答があった解析環境を優先順位の 1 番目とし，次に不正サイトの集合から優先順位 1 で選択された解析環境で応答があった不正サイトの集合を除いた不正サイトの中で，最も多くの応答があった解析環境を優先順位の 2 番目とした．なお，応答数が同じ場合は，脆弱性の数が多い環境を優先して選択した．

表 6.6　応答があった不正サイトの数

| UA-ID | 応答不正サイト数 | UA-ID | 応答不正サイト数 |
|---|---|---|---|
| 1 | 318 | 7 | 317 |
| 2 | 318 | 8 | 316 |
| 3 | 315 | 9 | 314 |
| 4 | 314 | 10 | 315 |
| 5 | 315 | 11 | 315 |
| 6 | 318 | 12 | 317 |

表 6.7　不正サイトの応答サイズ（bytes）

| UA-ID | Case 644 | Case 1054 | Case 1120 |
|---|---|---|---|
| 1 | 2,684 | 322 | 36,839 |
| 2 | 2,684 | 322 | 36,839 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 6 | 2,684 | 322 | 36,839 |
| 7 | 2,684 | 322 | 36,839 |
| 8 | 2,684 | 322 | 0 |
| 9 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 |
| 12 | 2,684 | 322 | 0 |

表 6.8　解析環境の優先順位

| 優先順位 | OS | Web ブラウザ |
|---|---|---|
| 1 | Windows XP (sp3,x86) | Internet Explorer 8 |
| 2 | Windows Vista (sp2,x64) | Internet Explorer 9 |
| 3 | Windows XP (sp3,x86) | Chrome25 |
| 4 | Windows Vista (sp2,x64) | Internet Explorer 8 |
| 5 | Windows Vista (sp2,x64) | Chrome25 |
| 6 | Windows Vista (sp2,x64) | Internet Explorer 7 |
| 7 | Windows 7 (sp1,x86) | Internet Explorer 9 |
| 8 | Windows Vista (sp2,x64) | Firefox4 |

## 6.4. 評価実験

本節では，多種環境による不正サイト解析の有効性を評価するために実施した評価実験の結果について述べる．

### 6.4.1. 評価目的

6.3 節で導出した解析環境を用いて実際の不正サイトに接続し，単一の解析環境ではどの程度不審な挙動を見逃しているかを評価する．具体的には不正サイトに接続した際に発生する不審なホストへのリダイレクト（以下，接続先ホスト）を観測し，解析環境の違いによる接続先ホストの違いを評価する．

### 6.4.2. 評価方法

評価に用いた解析環境の構成を表 6.9 に示す．なお，評価には，6.3 節で導出した優先順位 1～3 の解析環境に加え，ブラウザの多様性を確保するために優先順位 8（Firefox4）の解析環境も用いた．

表 6.9　解析環境

| Sandbox ID | OS | Web ブラウザ |
|---|---|---|
| 1 | Windows XP (sp3,x86) | Internet Explorer 8 |
| 2 | Windows Vista (sp2,x64) | Internet Explorer 9 |
| 3 | Windows XP (sp3,x86) | Chrome25 |
| 4 | Windows Vista (sp2,x64) | Firefox4 |

### 6.4.3. 評価結果

実際の不正サイト 218 件に接続し，解析環境の違いによる振る舞いの違いを評価した．4 種類の解析環境で観測した接続先ホストの数及び，接続先のうち VirusTotal[87]によって悪性ホストとして判断されたホストの数を表 6.10 に示す．なお，悪性ホストの総数は 66 ホストであった．

表 6.10 より，観測された悪性ホストの数は解析環境ごとに異なり，Windows XP, Internet Explorer 8 の環境が最も多くの悪性ホストを観測できることが分かった．表 6.11 に，各解析環境で観測できた悪性ホストのカバー率を示す．これにより一つの解析環境で，平均 82.2%（最小 74.2%）の悪性ホストをカバーしていることがわかる．つまり，一つの解析環境を用いた解析では，平均で 17.8%（最大 25.8%）の悪性ホストを見逃してしまう可能性があるとも言える．

4 つの解析環境に共通して観測された悪性ホストの数を組み合わせごとに表 6.12 に示す．また図 6.3 に悪性ホストの包含関係を示す．ある特定の解析環境で観測された悪性ホストの数を円の大きさで示し，円が重複している部分は共通して観測された悪性ホストの数を表している．本図より，Windows XP, Internet Explorer8 でのみ観測された悪性ホストは 6 ホスト，Windows XP, Chrome でのみ観測された悪性ホストは 1 ホスト，Windows Vista, Firefox でのみ観測された悪性ホストは 3 ホストであることが分かった．また，4 つの解析環境で共通する悪性ホストの数は 42 ホストであったことから，36.4%（24 ホスト/66 ホスト）は環境によって振る舞いが変化する悪性ホストであると言える．

表 6.10　不正サイトの数

| Sandbox ID | ホスト数 (総数) | ホスト数 (ユニーク) | 不正ホスト数 (総数) | 不正ホスト数 (ユニーク) |
|---|---|---|---|---|
| 1 | 1,643 | 456 | 228 | 58 |
| 2 | 1,066 | 321 | 208 | 55 |
| 3 | 1,875 | 464 | 229 | 49 |
| 4 | 2,945 | 714 | 213 | 55 |

表 6.11　悪性ホストカバー率

| Sandbox ID | カバー率 |
|---|---|
| 1 | 87.9% |
| 2 | 83.3% |
| 3 | 74.2% |
| 4 | 83.3% |

表 6.12　悪性ホスト数と解析環境の関係

| Windows XP InternetExplorer 8 | Windows XP Chrome25 | Windows Vista InternetExplorer 9 | WindowsVista FireFox4 | 共通する悪性ホスト数 |
|---|---|---|---|---|
| ○ | | | | 58 |
| | ○ | | | 49 |
| | | ○ | | 55 |
| | | | ○ | 55 |
| ○ | ○ | | | 44 |
| ○ | | ○ | | 51 |
| ○ | | | ○ | 49 |
| | ○ | ○ | | 47 |
| | ○ | | ○ | 46 |
| | | ○ | ○ | 51 |
| ○ | ○ | ○ | | 43 |
| ○ | ○ | | ○ | 43 |
| ○ | | ○ | ○ | 48 |
| | ○ | ○ | ○ | 45 |
| ○ | ○ | ○ | ○ | 42 |

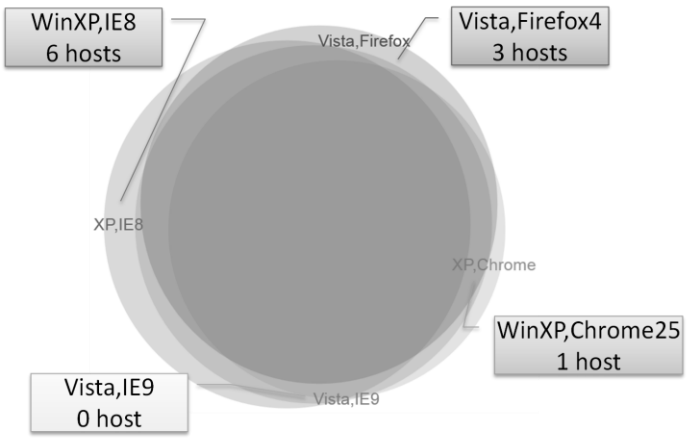



図 6.3　悪性ホストの包含関係

以上の評価結果より，多種環境で不正サイトを解析することで，単一解析環境だけでは見過ごす可能性のあった，悪性ホストを明らかにできることがわかり，多種環境を用いた不正サイト解析の有効性を確認した.

## 6.5. 考察

(ア) ブラウザの脆弱性を突く攻撃

本章で実施した評価で用いた不正サイトの中には，ブラウザの脆弱性を突いてマルウェアを実行するドライブバイダウンロード攻撃を観測することが出来なかった．Nappa らの調査によると，改ざんされたサイトの生存期間は短く，約 60%のサイトが 1 日以内に閉鎖されてしまう[88]．このため，ドライブバイダウンロード攻撃を観測するためには，不正サイトの入手と同時に解析（評価）することが望ましい．

(イ) ユーザエージェントの相違による応答変化

ここではユーザエージェントの違いによる応答の変化が見られた Case1054 に関して考察する．解析環境を用いて Case1054 の不審ホストに接続した際に Windows XP, Internet Explorer 8 の解析環境でのみに応答があり，その他の解析環境では応答がなかった．この結果より，Case1054 の不正サイトが備えている解析回避機能として，以下の 2 つが考えられる．

・サーバ側で環境を確認し，応答を変化させている

・同じ IP アドレスからの 2 回目の接続には応答しない

一つ目の解析回避機能は，多種環境で解析する本システムによって解析することができるが，二つ目の解析回避機能に関しては，本システムをそのまま適用するだけでは解析できない．これに対しては，一度目の通信で応答されたコンテンツをキャッシュし，別の解析環境で解析させることで，解析回避機能による影響を回避することが可能になる．

(ウ) 解析環境

本章で提案した不正サイト解析システムは，Windows 環境の解析環境のみを対象として構成している．しかし，個人保有の携帯端末を職場に持ち込みそれを業務に使用する BYOD[14]の普及が進んでいることなどから，今後企業内で業務に活用される OS やブラウザの種類や組合せは変かしていくと考えられる．この問題に対しては，企業で業務に活用される OS 等のシェアを継続的に調査し，随時解析環境を更新していくことで対応する．

## 6.6. 結論

本章では，多種環境を用いた不正サイト解析システムを提案し，環境に応じて応答を変化させる不正サイトの解析を行った．さらに，このような不正サイトの解析に有効に動作する解析環境を考察した．また，実際の不正サイトを用いた評価実験により，不正サイトの中には解析環境の違いにより応答するコンテンツを変化させるものが存在し，単一の解析環境では平均 17.8%（最

[14] BYOD (Bring your own device)は，組織に属する人が個人で所有する機器を組織に持ち込み，それを組織における作業にも使用すること．

大 25.8%）の不正サイトを見逃していることを明らかにした．これより，多種環境で解析を行う不正サイト解析システムの有効性を確認した．今後は，不正サイトの解析を継続して行い，攻撃者の不正活動の実態を明らかにする．

## 参考文献


[76] SECURELIST : ドライブバイダウンロード: 危険にさらされる Web，入手先＜http://www.viruslistjp.com/analysis/?pubid=204792056＞(参照 2016-07-28)

[77] Kaspersky : APT「ブルーターマイト」:新たな手口で感染拡大，入手先＜https://blog.kaspersky.co.jp/blue-termite-apt-targeting-japan/8412/＞(参照 2016-07-28)

[78] 高田雄太, 秋山満昭, 針生剛男 : ドライブバイダウンロード攻撃に使用される悪性な JavaScript の実態調査, 電子情報通信学会技術研究報告, vol. 113, no. 502, pp. 59-64, 2014

[79] Yi-Min Wang , Doug Beck, Xuxian Jiang, Chad Verbowski, Shuo Chen, Sam King : Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities, In Proceedings of Network and Distributed System Security Symposium (NDSS), pp. 35-49, 2006

[80] Mitsuaki AKIYAMA, Makoto IWAMURA, Yuhei KAWAKOYA, Kazufumi AOKI, Mitsutaka ITOH : Design and Implementation of High Interaction Client Honeypot for Drive-by-Download Attacks, IEICE TRANSACTIONS on Communications, Vol.E93-B No. 5 pp. 1131-1139

[81] The Honeynet Project : Capture-HPC, 入手先＜https://projects.honeynet.org/capture-hpc＞(参照 2016-07-28)

[82] Sourceforge : HTML Unit, 入手先＜http://htmlunit.sourceforge.net/＞(参照 2016-07-28)

[83] ソフトバンクグループ : 企業 PC の OS シェア，Windows 7 が 74.4%に，入手先＜http://www.sbbit.jp/article/cont1/28758＞(参照 2016-07-28)

[84] Webrage : Web ブラウザシェアランキング TOP10，入手先＜https://webrage.jp/mobile/data/pc_browser_share.html＞(参照 2016-07-28).

[85] NIST : National Vulnerability Database, 入手先＜https://nvd.nist.gov/＞(参照 2016-07-28)

[86] NIST:NVD-CVSS, 入手先＜https://nvd.nist.gov/cvss.cfm＞(参照 2016-07-28)

[87] Google : VirusTotal, 入手先＜https://www.virustotal.com/ja/＞(参照 2016-07-28)

[88] A. Nappa, M. Z. Rafique, and J. Caballero : Driving in the cloud: An analysis of drive-by download operations and abuse reporting, In dimva, 2013

# 7. 挙動に基づく自動対策

## 7.1. 背景と目的

著者はこれまでにマルウェアを解析して、その特徴（マルウェアによる接続先等）を取得する技術を提案してきた．しかしながら，2 章でも述べたように，自動解析によって得られたそれらの特徴や，世の中で共有されているセキュリティ情報（インテリジェンス）には不確実な情報が含まれている。このため，その情報を対策に利用すると誤った対策を実施してしまう可能性があり，実務で自動対策を実施することが困難であった．そこで，著者は，これらの不確実な情報であっても自動対策に適切に活用することで攻撃者らが連携を深めて高度化するサイバー攻撃に対して，守る側の集団防御を実現する自律進化型防御システム（AED：Autonomous Evolution of Defense）の研究を進めている．本技術は信頼関係のない他の組織から共有された不確実なインテリジェンスであっても，本来業務への悪影響を最小限に抑えつつ，対策に活用できるようにするものである．これにより，共有されたインテリジェンスに基づくシームレスな対策を実現し，EMDIVI で二の舞を演じた同様の被害の発生を未然に防ぐ．

標的型攻撃はサイバーキルチェーン[89][90]と呼ばれる「偵察（Reconnaissance）」「武器化（Weaponization）」「デリバリー（Delivery）」「エクスプロイト(Exploitation)」「インストール（Install）」「遠隔操作（Command & Control）」「目的実行（Objectives）」の 7 つのステップに整理できると考えられており，最終ステップである目的実行までのいずれかのステップで防御を成功させることによって実質的な被害を抑えられる，という考え方が重要となる．

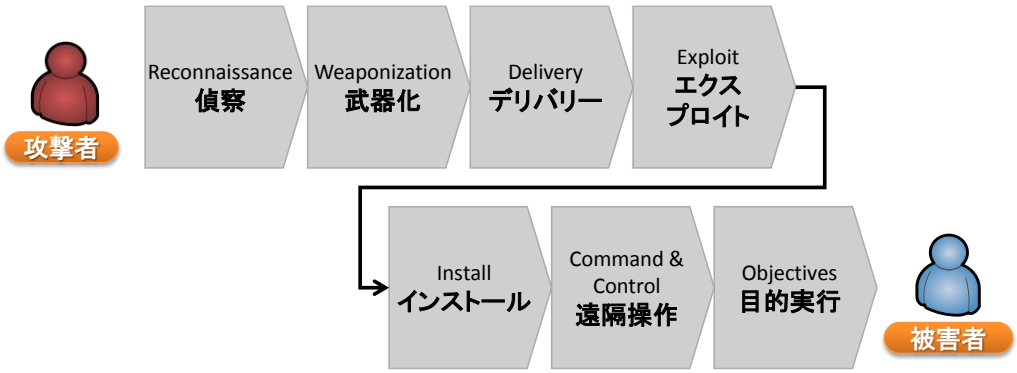



図 7.1　サイバーキルチェーン

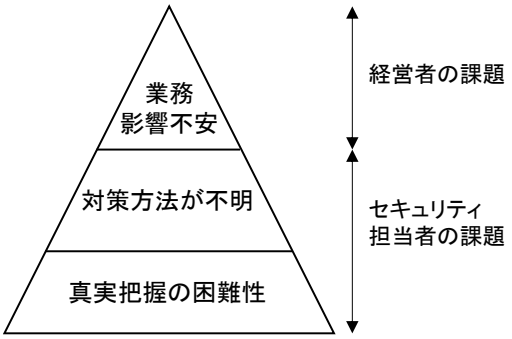



図 7.2　セキュリティ対策の 3 段階阻害要因

2 章で述べた日本年金機構を端緒とする 44 件の連続的な漏えい事件では, サイバーキルチェーンの初～中期段階として位置づけられる高度なマルウェアの侵入, 感染といった「偵察」から「インストール」までは防げなかったとしても，被害が発覚してすぐにインテリジェンスを共有および活用できていれば「遠隔操作」（内部感染拡大や情報詮索）や「目的実行」（個人情報のアップロード）のステップで情報漏えいを防げた可能性が高い．例えば，マルウェアを解析したり，プロキシログ等から得られたりする情報漏えい先等のインテリジェンスを迅速に共有して対策に活用できていればこのような被害拡大は防げた．このような目前の脅威の存在に気付きながらも，自組織の防衛に有用であるはずのインテリジェンスを活用した対策に踏み込めない原因として，図 7.2 に示すセキュリティ対策の 3 段階阻害要因があると考える．

阻害要因 1：真実把握の困難性
阻害要因 2：対策方法が不明
阻害要因 3：対策による既存業務への悪影響の懸念

阻害要因 1 は現在起きている脅威が何か，どのようなリスクが自組織に内在するのかを把握できない問題に起因する．阻害要因 2 は自組織のリスクを把握できたとしても，専門性の欠如等からリスクを軽減する方法がわからない問題に起因する．阻害要因 3 はリスク軽減方法がわかったとしても，対策によるリスク軽減効果や本来業務への悪影響の事前把握が難しく意思決定が遅れてしまう問題に起因する．阻害要因 1，2 はサイバー攻撃に起因するリスク排除を優先するセキュリティ担当者の抱える課題であり，阻害要因 3 は事業利益の最大化を優先する経営者の課題である．このように立場の異なる人間の合意形成の難しさが迅速な対策を困難にしている．

著者らは阻害要因 1 を解決するために，標的型攻撃等で悪用されるマルウェアを自動的に解析し，マルウェア感染による影響を把握する多種環境マルウェア動的解析システム（M3AS）を開発し 2 章で述べた．前述した日本年金機構の情報漏えい問題で悪用された EMDIVI もそうで

あるように，マルウェアの 82.8%はマルウェア感染後にインターネットに接続して新たなマルウェアのダウンロードや，遠隔操作者（C&C サーバ）との通信，機密情報の窃取のためのアップロード通信等，外部サーバとの通信が発生することが知られている．つまりマルウェアに感染したとしても，上記の外部サーバとの通信を検知，遮断することによって実害の発生リスクを軽減できる．情報処理通信機構はインターネットへの出口に設置したプロキシのユーザ認証機能を有効化することでマルウェア感染による被害拡大を抑止できるとし，同設定を推奨している[91]．また，マルウェアのアクセス先をブラックリストとして定義し，プロキシや FW 等で遮断とすることも阻害要因 2 の解決策となる．ブラックリストとしては Spamhaus[92]や MalwareDomainList[93]等，リストそのものが提供されているものから，FireEye EX[94]や上記 M3AS 等のサンドボックス解析手法を用いてマルウェアから動的に作成されるものがある．このように著者らは M3AS の研究開発を通じて，セキュリティ担当者の抱える課題（阻害要因 1，2）の解決を図ってきた．

しかし，プロキシに対応したマルウェアのうち 8.7%（92 件対中 8 検体）がプロキシ認証を突破すると 4.5.1 項でも報告したとおり，プロキシ認証が万全とはいえない状況となってきている．また，マルウェアの実行環境がインターネットと通信可能かを判断するためにマルウェア実行初期に正規なサーバに対して疎通確認を行う場合もあり，マルウェアのアクセス先の全てが不正なサーバとはいえない．これらに対してプロキシ認証に加えて毎回新たな作業を利用者に強いたり，解析の結果得られたアクセス先を一様にブロックしたりする対策は，情報システムの利便性の低下や正規なサーバへのアクセスの遮断等，業務への悪影響（可用性の低下）に繋がるため，事業継続や利益最大化を任務とする経営者からは問題視されていた．これが経営者の抱える課題（阻害要因 3）の代表例である．

前述したように阻害要因 1，2 はこれまでに解決が図られてきたが，阻害要因 3 に対する有効な解決策は図られていない．そこで，本章では上記阻害要因 3 を解決するため，これまでに提案した M3AS 等から得られるインテリジェンスを活用して自動対策を実現するとともに，業務に影響を及ぼしにくい集団防御を実現する AED を提案する．

## 7.2. 関連研究

マルウェアを解析して得たレポートを対策に活かして集団的に防御する技術が Colajanni[95]らによって提案されている．本提案手法では，複数の組織にハニーポット等のセンサーを配置して，中央のサーバでマルウェアを収集する．そして収集したマルウェアをマルウェア動的解析サービスなどで解析して，マルウェアの解析レポートを得る．その解析レポート（たとえばアクセス先やポート，プロトコル等）を，各組織とシェアすることによってマルウェアの通信を遮断して対策する．また，Tsai らの研究では，クライアントによる SNS ウェブサイトへのアクセスの前に，そのウェブサイトをスキャンして，危険があった場合にプロキシで遮断および警告メッセージを表示する技術が提案されている[96]．本手法は，マルウェア解析と連携して対策に活用す

る点で類似している．しかし，本手法はマルウェアが正常な動作（例えば検索エンジンへのアクセス等）をした場合には，誤った自動対策がとられてしまい，利用者の業務を妨げる可能性がある．

マルウェアに感染したクライアントによる社外サーバへの情報漏えいを防止する一つの手法として，社外サーバへアクセスする際に CAPTCHA 認証を求める方式が提案されている[97]．CAPTCHA は機械と人とを判別する逆チューリングテストであり，マルウェアのようなプログラム（機械）では CAPTCHA を解読することができない特性を利用する．CAPTCHA 認証を行うことで，クライアントに感染したマルウェアがブラウザを乗っ取り，社外サイトへアクセスすることを防ぎつつ人間による意図的なアクセスを許可することができる．しかし，CAPTCHA 認証は人間にとっても認証困難であることが多いため,外部サイトへのアクセスの度に CAPTCHA 認証を行っていては日々の業務の妨げとなる．

悪性な社外サイトへのアクセスを防ぎつつも，安全性の高いサイトへのアクセス時には CAPTCHA 認証を省略することで業務への影響を軽減する手法の一つに，ブラックリストとホワイトリスト，これら 2 つのリスト以外に一定の基準を満たした不審な外部サイトはグレーリストへ振り分けておき，グレーリストへのアクセスに対してのみ CAPTCHA 認証を行う方法が提案されている[98]．しかし，本方式は不審な外部サイトを決定付ける基準が静的であるため日々進化する脅威に追従するのは難しい．

セキュリティ担当者と経営者の合意形成の困難さを解決するために，セキュリティを高めた対策にすべきか業務効率等の利便性を重視した対策にすべきかを，セキュリティと利便性との間のトレードオフを評価して対策を選定する方式が提案されている[99]．

上記対策の選定手法ではトレードオフを前提としてそれらのバランスを数理的に解決するものであり，リスクとコスト低減のいずれかの妥協が必要となる．

著者の提案手法は，不確実な情報が含まれる解析レポートを用いる対策として，いきなり遮断するのではなく認証を追加することによって，人間による意図的な通信は許可するとともに，その認証結果を用いて不確実な情報を確実性の高い情報に振り分けていくことによって，認証の頻度など，対策の精度を運用していく中で自律的に高めていく点で，優れている．

## 7.3. 自律進化型防御システム

本節では，前述した問題を解決しリスク低減と利便性とを両立するために，2 つの技術を提案する．1 つは，正当性や信頼性の欠如するインテリジェンス，ここでは不審な URL リスト（グレーリスト）を活用して，業務へ与える悪影響を最小限に抑えつつ悪性サイトへの接続リスクをも低減し，阻害要因 3「対策による既存業務への悪影響の懸念」を解決するリスクベースプロキシ制御技術である．もう 1 つは，グレーリストから不審な URL ルールを動的に生成し，リストに無い未知の URL への追加認証をも実現することで，さらなるリスク低減を実現する認証条件最適化技術である．

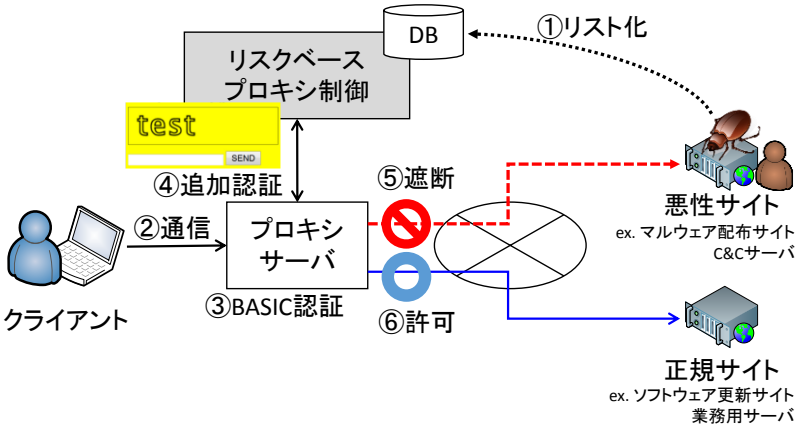



図 7.3　リスクベースプロキシ制御の概要

### 7.3.1. リスクベースプロキシ制御

グレーリストに登録された URL へのアクセスに対し，マルウェアでは解決困難な認証手段（本提案手法でも CAPTCHA 認証を用いるが，他の手段でも構わない）を追加することで，ユーザの本来の業務の可用性を損なわず，標的型攻撃で多用されている遠隔操作型マルウェアに対策可能なリスクベースプロキシ制御機能のアーキテクチャを提案する．本機能の概要を図 7.3 に示す．

プロキシはユーザ認証機能（BASIC 認証や LDAP[15]認証，AD[16]認証連携等）をサポートしており，多くの組織で外部サーバへアクセスするユーザのアクセス履歴を記録するとともに，パスワードを知らないマルウェアのアクセスをブロックすることで遠隔操作型のマルウェアのリスクを低減していた．しかしながら前述したように様々な方法でプロキシ認証を突破するマルウェアが出現してきている．具体的にはブラウザにキャッシュされている認証情報を詐取するものや，認証済みのブラウザプロセスに悪性なコードをインジェクションしたりするマルウェアが確認されている．これらの最新の脅威に対抗するため，M3AS 等の解析結果に基づき悪性サイトをグレーリストに管理①しておき，クライアントの通信②に対して既存のユーザ認証③に加えて，新たな認証（例えば CAPTCHA 認証）を追加④する．この認証により，マルウェアからの外部アクセスを排除⑤し，かつ業務の可用性を保持⑥することが可能となる．

続いてリスクベースプロキシ制御機能の構成を図 7.4 に示す．リスクベースプロキシ制御機能は，プロキシに機能拡張するための標準仕様 ICAP（Internet Content Adaptation Protocol）[100]を利用して，URL チェックや CAPTCHA 認証機能との連携を実現する．

---

[15] Lightweight Directory Access Protocol の略で，ディレクトリ・サービスに接続するために使用される通信プロトコルの一つ．

[16] Active Directory の略で，マイクロソフトによって提唱されているディレクトリ・サービス・システム．Windows 2000 Server から導入された，ユーザとコンピュータリソースを管理するコンポーネント群の総称．

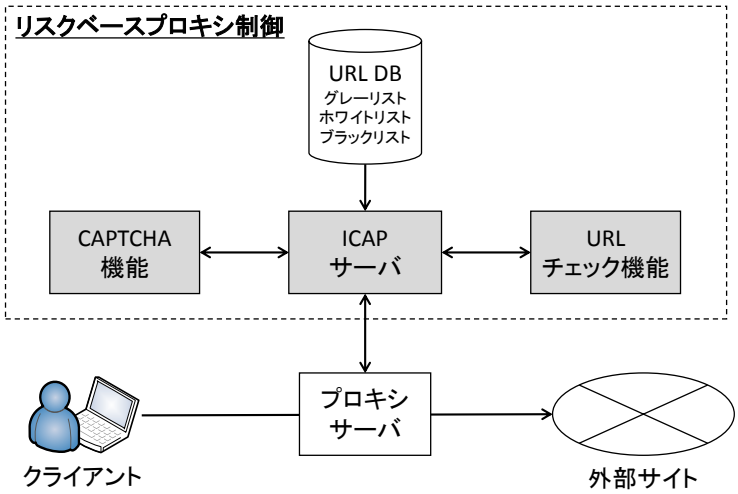



図 7.4　リスクベースプロキシ制御機能の構成

リスクベースプロキシ制御機能は以下の 5 つの機能およびデータベースから構成され，クライアントのアクセス先のリスクに応じて, CAPTCHA 認証を追加する. 各構成要素を以下に述べる.

(ア) プロキシサーバ

本機能は，ユーザによるインターネットアクセスを代理するとともに，ICAP コアと連携してユーザのアクセス先に応じた制御を行う.

(イ) ICAP サーバ

本機能はプロキシサーバに届いたクライアントからのリクエスト情報を ICAP に従ってプロキシサーバから受信し，URL チェック機能や CAPTCHA 機能と連携して，ユーザからのリクエストに対して認証の追加やアクセスの遮断といった制御を実施する.

(ウ) URL チェック機能

本機能は URL データベースと連携して，ICAP サーバが制御しようとしている URL にどの程度のリスクがあるかを応答する.

(エ) CAPTCHA 機能

本機能は CAPTCHA 認証に必要な歪み画像および認証フォームの生成を行う．また ICAP サーバからの要求に応じてユーザからの CAPTCHA 追加認証応答の正当性を判定して返答する.

(オ) URL データベース（URLDB）

本データベースには，正当性や信頼性の欠如する不確実な URL 群をグレーリストとして，不正である確度の高い URL 群をブラックリストとして，安全性の高い URL 群をホワイトリストとしてそれぞれ管理する.

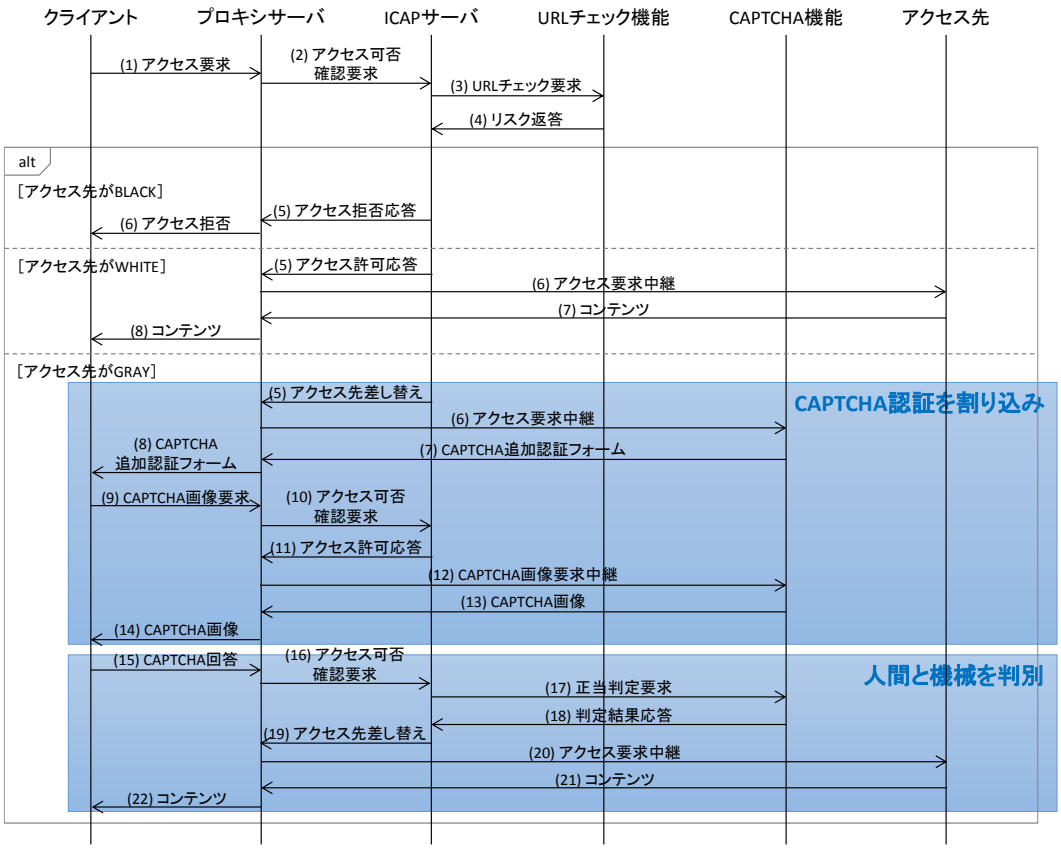



図 7.5　シーケンス図

上記機能によるリスクベースプロキシ制御機能の処理シーケンスを図 7.5 に示す．本機能は不確実な情報（グレーリスト）へのユーザ（クライアント）のアクセスに対して CAPTCHA 認証を割り込ませ，人間（ユーザ）と機械（マルウェア）とを判別して通信の制御を行うことが特徴である．

1. クライアントがプロキシサーバに対してアクセス要求を送信．
2. プロキシサーバはクライアントからのアクセス要求に対し，ICAP サーバへアクセス可否の確認要求を送信．
3. ICAP サーバは URLDB と連携する URL チェック機能に対し，アクセス要求に含まれるアクセス先のリスクを確認．
4. URL チェック機能は URL（あるいはドメインや FQDN[17]）のリスクを送信．

上記リスクがブラックの場合

5. ICAP サーバはアクセス拒否応答をプロキシサーバに送信．
6. プロキシサーバがアクセスを拒否するメッセージをクライアントに送信．

上記リスクがホワイトの場合

[17] Fully Qualified Domain Name の略で，完全に記述されたドメイン名．

5. ICAP サーバはアクセス許可応答をプロキシサーバに送信.
6. プロキシサーバがアクセス先サーバへのアクセス要求中継を依頼.
7. アクセス先サーバがプロキシサーバに要求に対応するコンテンツを送信.
8. プロキシサーバがクライアントにコンテンツを送信.

上記リスクがグレーの場合

5. ICAP サーバは，アクセス先を CAPTCHA 機能に差し替えた応答をプロキシサーバに送信.
6. プロキシサーバは CAPTCHA 機能に対してアクセス要求中継を依頼.
7. CAPTCHA 機能は CAPTCHA 追加認証フォームをプロキシサーバに送信.
8. プロキシサーバが CAPTCHA 追加認証フォームをクライアントに送信.
9. クライアントが CAPTCHA 追加認証フォームに含まれる CAPTCHA 画像を取得するためのアクセス要求をプロキシサーバに送信.
10. プロキシサーバは ICAP サーバへアクセス可否の確認要求を送信.
11. ICAP サーバはアクセス先が CAPTCHA 機能であるためアクセス許可レスポンスをプロキシサーバに送信.
12. プロキシサーバは CAPTCHA 機能に対するアクセス要求中継を依頼.
13. CAPTCHA 機能は CAPTCHA 画像をプロキシサーバに送信.
14. プロキシサーバは CAPTCHA 画像を取得しクライアントに送信.
15. クライアントはユーザから入力された CAPTCHA 認証への回答（テキスト値）を含んだアクセス要求をプロキシサーバに送信.
16. プロキシサーバは ICAP サーバへアクセス可否の確認リクエストを送信.
17. ICAP サーバは認証の正当性を CAPTCHA 機能に確認.
18. CAPTCHA 機能は正当性を判定し，ICAP サーバに結果を送信.
19. 正当性が確認できた場合，ICAP サーバはアクセス先を本来クライアントが要求していた URL に差し替えた返答をプロキシサーバに送信．正当性が確認できなかった場合は処理 5 に遷移.
20. プロキシサーバが本来のアクセス先へのアクセス要求中継を依頼.
21. アクセス先サーバがプロキシサーバにコンテンツを送信.
22. プロキシサーバがクライアントにコンテンツを送信.

上記の処理シーケンスによりアクセス先のリスクに応じて，遮断，アクセス許可，CAPTCHA 認証追加の制御を行う.

### 7.3.2. 認証条件最適化

前項で述べたリスクベースプロキシ制御機能によって，例えグレーリストに誤って登録された業務上アクセスが必要な正規サイトへのアクセスも，ユーザの認証が成功すれば支障なくアクセスできるようになる．これにより悪性サイトに接続してしまうリスクを低減しつつ，業務への悪影響を抑え，阻害要因 3 の解決に寄与する．しかしながらグレーリストの拡充に伴い，誤って登録される正規サイトの数も増えてくることが想定され，その度にユーザに追加認証をさせることは，業務効率の低下につながり望ましくない．また，URL データベースに事前に登録されていない URL（未知の URL）へのアクセスに対しての防護策が存在しない．

そこで本項では，グレーリストに登録された不審サイトの安全性を，ユーザの認証結果に基づいて評価するリスト良質化機能を提案する．さらに危険性あるいは安全性の高いサイトに共通する属性を学習することにより，未知の URL に対するアクセス時でもリスクベースプロキシ制御機能で防護可能とさせる不審属性学習機能も提案する．本章では防御システムの自律進化を支援するこれら 2 つの機能をまとめて認証条件最適化機能と呼ぶ．

(ア) リスト良質化

リスト良質化機能はグレーリストに含まれる URL に対して，クライアントによる認証結果を表 7.1 に示す 3 種類のステータスに分類して認証実績として記録する．これによって，URL 毎の認証成否傾向が集計できる．

こうやって集計した URL 毎の認証実績の統計値に基づいて危険度の高い URL（ブラック），安全な URL（ホワイト）に分別する．具体的には CAPTCHA 認証要求数に対し，認証成功となった数を認証成功数，認証失敗となった数を認証失敗数，認証無試行となった数を認証無試行数として，それぞれの値を各 URL の認証実績値として定義する．

本章では単純に，認証要求数に対する認証無試行数の割合が一定値以上あるいは一定数以上の認証実績値を持つ URL をブラックリストに追加する．また，認証要求数に対する認証成功数の割合が一定値以上あるいは一定数以上に達した認証実績値となった場合はホワイトリストに追加する．本機能により，認証を繰り返すことでグレーリストの URL がホワイトリストおよびブラックリストに機械的に振り分けられる．

表 7.1　認証ステータス

| ステータス | 説明 |
|---|---|
| 認証成功 | CAPTCHA の回答が正しく入力された状態 |
| 認証失敗 | CAPTCHA が入力されが，回答が正しくなかった状態 |
| 認証無試行 | CAPTCHA 認証に一定時間（例えば 10 秒等）回答が無かった状態 |

(ア) 不審属性学習

不審属性学習機能の構成を図 7.6 に示す.

不審属性学習機能は，プロキシサーバから得られるアクセス先 URL やグレーリストに格納された URL に関連する情報を調査して，その結果を「属性」として付与する属性付与機能を持つ．また，機械学習を用いてルールを生成する学習機能，そして生成したルールを用いて認証を追加するか否かを判断する予測機能を持つ．それぞれの機能について以下に述べる．

属性付与機能は，表 7.2 に示す情報を，グレーリストに URL が追加されたタイミング，あるいはクライアントがアクセス先 URL への代理接続をプロキシに依頼したタイミングで，アクセス先 URL を管理している Web サーバや，DNS，GeoIP[101]等の外部サービスを用いて取得する．表中の#9 は，DNS ラウンドロビン機能を悪用した Fast-Flux 攻撃[102]の場合に不一致になる可能性が高いことを，また#10 や#11 は正規のサイトが HTTP と HTTPS の両方のコンテンツを同等にメンテナンスされている可能性が高いことを想定した属性である．

学習機能はプロキシの BASIC 認証（図 7.3 ③）を試行した認証成功や認証失敗，あるいは CAPTCHA 認証（図 7.3 ④）を試行した URL への認証成功，認証失敗，認証無試行結果を目的変数として，また付与した属性を説明変数として決定木学習アルゴリズムによりルールを生成する．これにより生成された決定木の一部を抜粋して図 7.7 に示す．予測機能は，クライアントがグレーリストに登録されていない URL へアクセスする際に CAPTCHA 認証を表示させるか否かを上記ルールに基づき予測する．

本学習機能を定期的に実行することで変化を再学習させることができ，日々進化する脅威に追従することができるようになる．

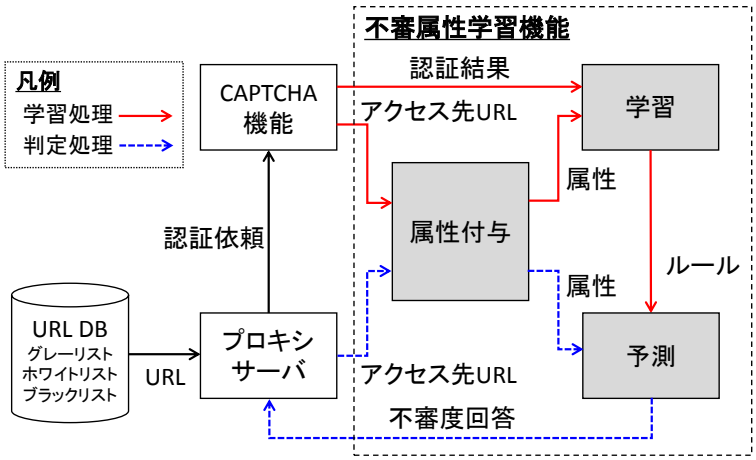



図 7.6　不審属性学習機能の構成

表 7.2　URL へ付与する属性

| # | 属性 | 取得先 | 説明 |
|---|---|---|---|
| 1 | HTTP ステータスコード | アクセス先 URL | HTTP アクセスした際のステータスコード |
| 2 | HTTP コンテンツサイズ | アクセス先 URL | HTTP アクセスした際のコンテンツサイズ |
| 3 | HTTPS ステータスコード | アクセス先 URL | HTTPS アクセスした際のステータスコード |
| 4 | HTTPS コンテンツサイズ | アクセス先 URL | HTTPS アクセスした際のコンテンツサイズ |
| 5 | DNS A レコード | DNS サーバ | FQDN から正引きした IP アドレス |
| 6 | DNS 逆引き | DNS サーバ | IP アドレスから逆引きした FQDN |
| 7 | カントリーコード | GeoIP サービス | IP アドレスから所在国を推定した国／地域 |
| 8 | AS 番号 | GeoIP サービス | IP アドレスが属する AS の番号 |
| 9 | 逆引き一致 | － | 上記#5 と#6 の整合性 |
| 10 | ステータスコード一致 | － | 上記#1 と#3 の整合性 |
| 11 | コンテンツサイズ差 | － | 上記#2 と#4 の差 |

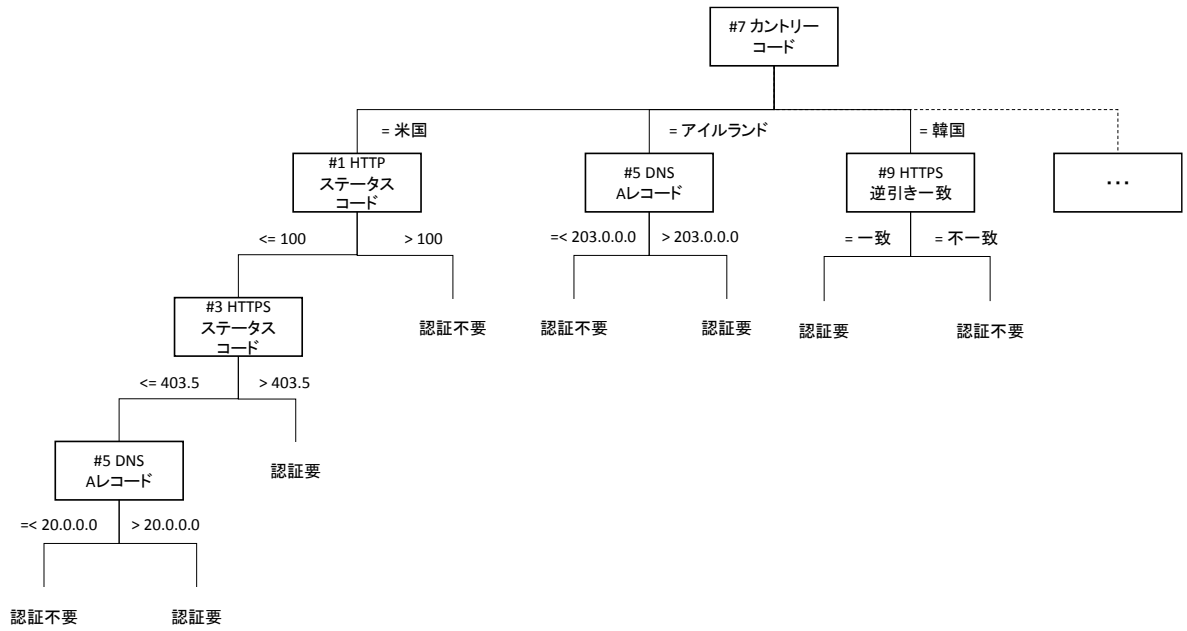



図 7.7　生成された決定木（一部）

## 7.4. 評価実験

本節ではAEDの主要機能であるリスクベースプロキシ制御機能と認証条件最適化機能の実装を行い，評価を行った．以降に評価の目的と結果を示す．

### 7.4.1. 評価目的

主に追加認証によるマルウェア感染端末のリスク低減効果や業務悪影響抑制効果，ユーザビリティ，不審属性学習機能による未知のURLへの効果を評価する．

評価1

追加認証によるリスク低減効果を評価するため，マルウェア解析結果等のインテリジェンス情報を元にグレーリストを生成し，当該グレーリストを用いてリスクベースプロキシ制御機能によるマルウェア感染端末のリスク低減効果を検証する．

評価2

追加認証による業務悪影響抑制効果を評価するため，評価1で用いたグレーリストとリスクベースプロキシ制御機能を実際のユーザに利用してもらい，認証条件最適化機能のリスト良質化効果を検証する．

評価3

追加認証によるユーザビリティへの影響を評価するため，グレーリストに正規サイトが数多く誤登録されている状況を再現した上でリスクベースプロキシ制御機能を実際のユーザに利用してもらい，コンテンツ表示画面への影響や認証頻度等を検証する．

評価4

グレーリストに登録されていない未知のURLに対して追加認証が適切に表示できるか評価するために，不審属性学習機能の評価を行う．本機能はクライアントによるアクセスの度に属性付与機能が呼び出されることから，本機能がボトルネックになる可能性がある．このため本機能の処理性能を評価する．加えて本機能により属性を学習させて得られたルールを用いて，未知のURLの予測精度を評価する．なお評価ではURLとしてPATHを含まないFQDNを用いる．

### 7.4.1. 評価結果

評価1では，表 7.3に示す52,653種類のユニークな不審URLをグレーリストとして用いる．本グレーリストにはマルウェア（著者の組織で2014年から2015年6にかけて独自に入手した2,064種類）をM3ASで解析して得た不審なURLや，VirusTotalやThreatConnectサービスから得た不審URLも含まれる．またブラックリストおよびホワイトリストは空にしている．

表 7.3　グレーリストに含まれる不審 URL の分類と数

| 情報源 | 数量 |
|---|---|
| M3AS | 16,384 |
| VirusTotal | 31,937 |
| ThreatConnect | 6,257 |
| 計 | 54,578 |
| ユニーク数 | 52,653 |

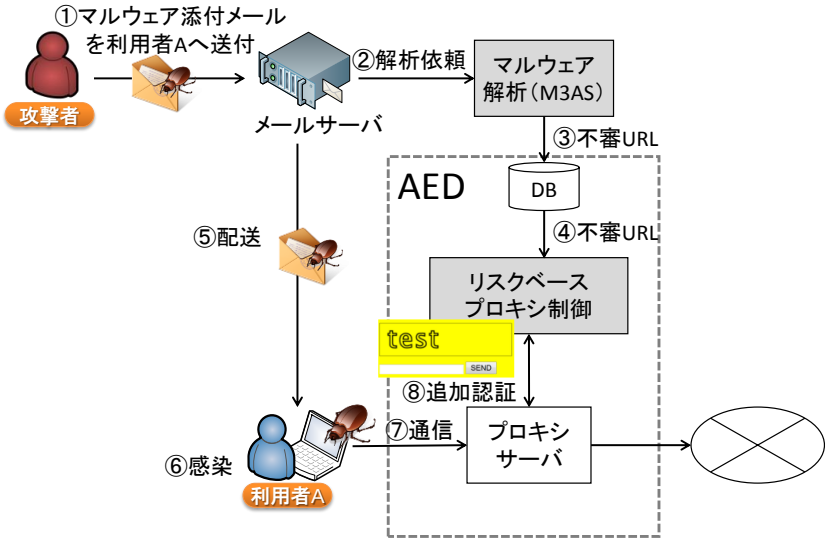



図 7.8　評価 1 の前提条件

また，攻撃者によりメールに添付されたマルウェア（①）は，図 7.8 に示すように，M3AS で事前に解析（②）してグレーリストを取得（③④）してからユーザ（被験者）に配送（⑤）されることを想定する．評価では，ユーザ端末で実行したマルウェアがインターネットにアクセスに成功するか否かを確認して，AED（リスクベースプロキシ制御機能）による遮断の失敗，成功を判断し，その精度を評価する．

評価結果を表 7.4 に示す．本表は前述した 2,064 種類のマルウェアのうち遮断に成功したマルウェアの数と，遮断に失敗したマルウェアの数を「マルウェア」の欄に示す．マルウェアの中には 1 つの検体が複数の URL にアクセスする種も存在するため，あるマルウェアが 1 回の実行で 3 種類の URL へアクセスする種であった場合に，本実験によって 2 種類の URL へのアクセスのブロックに成功し，残り 1 種類の URL へのアクセスのブロックに失敗した場合であっても，そのマルウェアの遮断は失敗として扱う．「URL」は，全マルウェアが実験中にアクセスした 1,007 種類の URL に対する遮断の成功数と失敗数等を示す．先ほどと同様に，1 つの URL に対して 1 つ以上遮断に失敗したケースが存在すれば，その URL の遮断は失敗として扱う．「アクセス」は，

実験中にマルウェアがアクセスした総数に対する遮断成否の数等を示している．

評価 2 では，被験者が日常的に業務で利用している情報システムに AED を導入して評価しており不正な通信は全く行われないクリーンな状態で被験者 35 名の協力を得て 9 日間かけて実施した．評価期間中の Web アクセス総数は 543,489 件で，接続先 URL（FQDN）のユニーク数は 5,146 件であった．認証追加判断機能による追加認証頻度及び評価機能によるリストの分別結果について表 7.5 に示す．評価の結果，約 19.9%の確率で追加認証に成功し，失敗率は 3.2%であった．また，図 7.9 に示すように予め用意したグレーリスト 52,653 件のうち，実験中にユーザが実際にアクセスした URL は 147 件, それらの URL に対して追加認証を実施した結果を用いて認証条件最適化機能（リスト良質化）により分別されたグレーリストは 28.5%程度であった．

なお，後に被験者に対して実施したアンケート結果では，本システムの導入により不便を感じていると答えた被験者は 0 であった．

表 7.4　マルウェアに対する AED の効果

| | 成功数 | 失敗数 | 精度 |
|---|---|---|---|
| マルウェア | 2,057 | 7 | 99.66% |
| URL | 1,000 | 7 | 99.30% |
| アクセス | 212,094 | 43 | 99.98% |

表 7.5　追加認証傾向およびグレーリストの分別結果

| No | 項目 | 結果 |
|---|---|---|
| 1 | 認証追加総数. | 186 (100.0%) |
| 2 | 追加認証を誰も思考しなかったドメインの数 | 149 (80.1%) |
| 3 | 追加認証に一人以上成功したドメインの数 | 37 (19.9%) |
| 4 | 追加認証に一人以上失敗したドメインの数 | 6 (3.2%) |
| 5 | グレーリストからホワイトリストに分類された数 | 7 (3.8%) |
| 6 | グレーリストからブラックリストに分類された数 | 46 (24.7%) |

評価 3 では，意図的にグレーリストに正規なサイトをノイズとして追加し，その結果得られる認証ステータスを集計するとともに，コンテンツ表示画面への影響を検証した．評価のために用意したグレーリストは，被験者が過去にアクセスした URL をアクセス数の多い順にソートし，その下位 30%（33,990 個）をグレーリストに追加して作成した．

被験者 10 名に対して約 11 日間実験した結果を表 7.6 に示す．グレーリストへの一致率は約 13.6%で，追加認証は 651 回実施された．この時の認証成功率は 27.3%となった．一方で，認証無試行が 274 件と，認証要求全体の 42.1%を占めた．認証無試行となったケースの多くは，アクセス先 URL の HTML ファイルが外部サーバに格納された CSS ファイルを外部参照していて，かつその外部サーバがグレーリストに登録されている場合であった．このため，グレーリストに誤登録されたサーバから CSS ファイル等を読み込む Web サイトでは，図 7.10 のようなレイアウトの乱れが生じた．これは，CSS ファイルや外部データファイル等がグレーリストに含まれるサーバに登録され CAPTCHA 認証の追加対象となっていたとしても，ユーザに対して認証画面を表示する手段が無いため，結果的に認証無試行となることに起因する．この問題はグレーリストへの誤登録が多い場合に顕在化するため，グレーリストの精緻化やホワイトリストの充実化が重要となる．

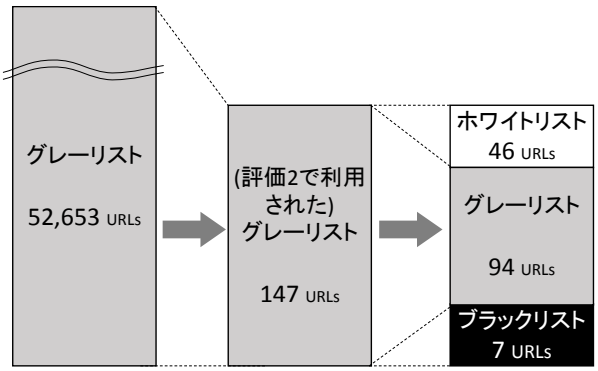



図 7.9　グレーリストの分別イメージ

表 7.6　リスクベースプロキシ制御機能に関わる統計

| 項目 | 数 | 割合 |
|---|---|---|
| 処理リクエスト総数 | 92,101 | 100.00% |
| グレーリスト一致数 | 12,479 | 13.55% |
| 認証要求数 | 651 | 100.00% |
| 認証成功数 | 178 | 27.34% |
| 認証失敗数 | 199 | 30.57% |
| 認証無試行 | 274 | 42.09% |

図 7.10　表示不具合の例（左：適用前，右：適用後）

表 7.7　属性付与機能実装サーバのスペック

| 項目 | スペック |
|---|---|
| CPU | Intel Core i7-2600 |
| Memory | 2GB |
| NIC | 1Gbps（光ネクストビジネス） |
| OS | Ubuntu 14.04 |

評価 4 では本来，実際のマルウェア感染端末による CAPTCHA 認証（図 7.3-④）試行データを用いることが望ましいが，前述の評価では感染事象が発生しなかったこと，グレーリスト一致数（URL ユニーク）680 種，CAPTCHA 認証失敗/無試行数（URL ユニーク）が 74 種と，学習および予測評価に十分な量でなかったことから，プロキシの BASIC 認証（図 7.3-③）試行データを用いて評価することとした．BASIC 認証はブラウザ起動直後に一度しか認証しないため CAPTCHA 認証のように URL 毎に認証実績が付与されない．しかしマルウェア等のプログラムが BASIC 認証に失敗する URL は CAPTCHA 認証にも失敗すると考えられるため，認証失敗実績の傾向は類似すると考える．

属性付与機能を実装したサーバのスペックを表 7.7 に示す．このサーバで属性付与した結果，1 つの URL あたり 0.22 秒の処理時間を要した．

表 7.8　不審属性学習機能による予測精度

| 項目 | 数 | 割合 |
|---|---|---|
| 全体 | 50,000 | 100.00% |
| 認証成功（ホワイト） | 49,174 | 98.35% |
| 認証失敗（ブラック） | 826 | 1.65% |
| 予測成功 | 49,411 | 98.82% |
| ホワイト予測成功 | 48,983 | 99.61% |
| ブラック予測成功 | 428 | 51.82% |
| 予測失敗 | 589 | 1.18% |
| ホワイト予測失敗 | 191 | 0.39% |
| ブラック予測失敗 | 398 | 48.18% |

次に独自に入手した実際の約 2500 万件の BASIC 認証ログを URL 単位に集計して「BASIC 認証に全て失敗」と「BASIC 認証を一度でも成功」の 2 種に分類した URL を目的変数として，さらに表 7.2 に示した属性を説明変数として生成した 100,000 件の学習用データを決定木学習アルゴリズムで学習させてルールを生成した．さらに，予測精度の評価用 URL を別途 50,000 件用意して，上記ルールと属性から BASIC 認証の成否を予測させて精度を測定した．結果，606 のルールが生成され，表 7.8 に示す予測精度となった．

以上の結果より，既存の認証結果を学習することで，未知の URL に対しても 99%近い精度で認証の成否を予測できることを確認した．

## 7.5. 考察

評価 1 では，2,064 種類のマルウェアを事前に解析して得た結果をグレーリストに用いていることから，本来であれば，同じマルウェアを用いて実験した場合は全ての通信が遮断できることを期待する．しかしながら実際には 0.3%程度のマルウェアを遮断できなかった．この理由としては，実行されるたびにアクセス先を変更するマルウェアが存在することに起因すると考える．このような，アクセス先を変えるマルウェアは DGA（Domain Generation Algorithm）[103]を用いている可能性がある．このような毎回アクセス先を変更するようなマルウェアに対応するには，グレーリストだけではなく，不審属性学習機能による属性ベースの遮断が重要となる．

評価 3 において用いた BASIC 認証は，ブラウザが一度認証に成功した後，そのブラウザが起動している限りはその認証結果（成功）がキャッシュに保存されるため，他の URL へ遷移した際に，再び BASIC 認証を加えることはできない．一方で CAPTCHA 認証は URL 単位で認証を追加できる．このため，不審属性学習機能の予測結果を用いて CAPTCHA 認証を追加／非表示することによって，BASIC 認証後の URL データベースに存在しない不審な URL（BASIC 認証を追加できなかった可能性の高い URL）へのアクセスを CAPTCHA で遮断（約 52%）することが

できると考える．また本来業務で必要な正規サイトを誤予測してしまった場合（約 0.4%）には，ユーザが CAPTCHA 認証を意識的に入力することによって，業務を阻害することなく意図した Web サイトへアクセスすることができる．一方で，本来 BASIC 認証で遮断されるはずの未知の URL の約 48%に対しては本機能の検知漏れにより CAPTCHA 認証が表示されることなくアクセスできてしまう．このため本項目の精度改善が今後の課題となる．

## 7.6. 結論

本稿では，普及するインテリジェンスが不確実な状態であった場合でも，業務に悪影響を与えることなく対策に活かすことが可能な AED を提案し，4 つの観点で評価した．

本システムでは，マルウェア動的解析システムやインテリジェンスサービスベンダが提供する不審な URL 情報をもとにプロキシに追加認証を加えることで，一様にアクセスを遮断するのではなく，ユーザの意思や認証行為を明確に示させることで，セキュリティ対策の 3 段階阻害要因の最終段階である「対策による既存業務への悪影響の懸念」を解決した．また認証結果と，その URL に関係する属性情報とを決定木学習アルゴリズムを用いて学習させることで，未知の URL に対して認証結果を 99%近い精度で予測できるルールの生成に成功した．本ルールに基づく予測機能を CAPTCHA 認証の追加判断条件に利用することで，52%程度の精度で不審な URL へのアクセス時に認証を追加できる見込みを得た．さらに本システムを運用することで，不確実な情報（不審な URL 情報）を 9 日間で約 28.5%，意味のある情報に良質化することを確認した．

一方で，不審な URL へのアクセスを 48%程度の確率で CAPTCHA 認証を経ずに許可しまう点は今後の課題であるが，URL 単位に認証実績が付与された CAPTCHA 認証の試行データを学習することにより，学習精度および予測精度が高まると考える．今後はリスクベースプロキシ制御機能の実証範囲を拡大し，上記 URL 単位の試行データを取得して再度評価したい．

AED は，不確実な情報でも自動対策に活用できる点で冒頭に述べたセキュリティ対策の阻害要因 3 を解消し得る．また，運用を継続することで URL リストやルールが自律的に進化するため，ネットワーク効果や運用コスト低減が期待できる．例えば複数の組織にそれぞれ導入した AED 間でグレーリストや認証ステータスを共有することにより，特にばらまき型メール攻撃[104]のような組織を跨る攻撃に対して集団防御の効果を発揮することができると考える．

## 参考文献


[89] Eric M. Hutchins, et al.: Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, ICIW2011, 2011.

[90] LOCKHEED MARTIN : Cyber Kill Chain, 入手先<http://cyber.lockheedmartin.com/solutions/>（参照 2016-11-25）

[91] 情報通信処理機構 :『高度標的型攻撃』対策に向けたシステム設計ガイド, 入手先< https://www.ipa.go.jp/files/000046236.pdf>（参照 2015-12-7）

[92] The Spamhaus Project, 入手先<https://www.spamhaus.org/>（参照 2015-12-7）

[93] Malware Domain List, 入手先< http://www.malwaredomainlist.com>（参照 2015-12-7）

[94] FireEye, Inc : E メール・セキュリティ, 入手先<https://www.fireeye.jp/products/ex-email-security-products.html>（参照 2015-12-7）

[95] Michele Colajanni, Daniele Gozzi, and Mirco Marchetti : Collaborative architecture for malware detection and analysis, IFIP International Federation for Information Processing, vol. 278, Proceedings of the IFIP TC 11 23rd International Information Security Conference, pp. 79-93, 2008

[96] Dwen-Ren Tsai, Allen Y. Chang, Sheng-Chieh Chung, You Sheng Li : A Proxy-based Re al-time Protection Mechanism for Social Networking Sites, Security Technology (ICCST), pp. 30-34, 2010

[97] The Official CAPTCHA Project, 入手先＜http://www.captcha.net/＞. （参照 2016-12-27）

[98] 角田朋, 大鳥朋哉, 藤井康広他 : グレーリストを用いたホワイトリストブラックリストの自動生成によるマルウェア感染検知方法の検討, 情報処理学会研究報告, Vol.66, No.16, 2014

[99] 芝口誠仁, 稲場太郎, 中山佑輝, 岡田謙一 : 仕事量及び利便性低下度に着目したセキュリティ対策選定手法. 情報処理学会研究報告, Vol.70, No.11, 2009

[100] Internet Engineering Task Force : Internet Content Adaptation Protocol (ICAP), 入手先<https://tools.ietf.org/html/rfc3507>（参照 2015-12-7）

[101] MaxMind, Inc : GeoIP2: 業界をリードする IP 情報収集, 入手先<https://www.maxmind.com/ja/geoip2-services-and-databases>（参照 2015-12-7）

[102] WIKIPEDIA : Fast flux, 入手先＜https://en.wikipedia.org/wiki/Fast_flux＞（参照 2015-12-7）

[103] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee and David Dagon : From Throw-Away Traffic to Bots:Detecting the Rise of DGA-Based Malware, Security'12 Proceedings of the 21st USENIX conference on Security symposium, pp. 491-506, 2012

[104] 情報通信処理機構 :【注意喚起】特定の組織からの注文連絡等を装ったばらまき型メールに注意, 入手先< https://www.ipa.go.jp/security/topics/alert271009.html>（参照 2015-12-7）

# 8. 結論

本論文では高度化するマルウェアを悪用した標的型攻撃に迅速対応するために，自動対策システムの提案を行った．

上記攻撃の自動対策を実現するにあたっては，マルウェア解析の困難性（問題１）と，解析結果という不確実な情報に起因する誤対策による業務悪影響懸念（問題２）が存在することを明らかにした．前者を解決するために，昨今問題となっている環境選択型マルウェアを自動解析し，攻撃者による遠隔操作や情報搾取を防止するために有用となる“マルウェアの接続先（不審サイト)”を抽出することで多層防御に必要な情報を得る目的１と，後者を解決するためにマルウェアによる被害発生リスクの軽減と誤った情報を用いて実施した自動対策による利用者の業務悪影響軽減の両立を実現する目的２の実現を目指した．

目的１を達成するために，動的解析を逃れるマルウェアの解析の実現（課題１）と，インターネット隔離環境では顕現しない機能の把握（課題２）を設定し，目的２を達成するために，マルウェアによる不正なアクセスの遮断（課題３）と，誤った情報による対策であったとしても業務上必要なアクセスは許可すること（課題４）をそれぞれ設定した．

課題１では，巧妙化が益々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多種環境マルウェア動的解析システム（Multi-modal Malware Analysis System, 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を特定した．この中には特定のアプリケーション「一太郎」の特定のバージョンがインストールされている環境でしか動作しないマルウェアや WindowsXP（ServicePack2）かつ物理環境（仮想環境ではない）でしか動作しないマルウェア，英語版の Windows でしか動作しないマルウェアの解析の自動解析および当該条件の自動導出に成功した．この他，全検体のうち 64%（357 検体）からマルウェアが依存する環境条件（顕現条件）を抽出できることを示した．

また，多層防御において重要な役割を果たす認証付プロキシを突破するマルウェアが増加している懸念から，M3AS をベースにプロキシアクセス型マルウェアを解析するプロキシ認証突破判定システムを提案し，実装した．本システムを用いてマルウェアがプロキシに対応しているか否か，プロキシに対応しているマルウェアが認証を突破するか否かを自動的に判断可能なことを確認した．さらに，2014 年 10 月に取得したマルウェア 629 検体を解析し，84 検体がプロキシ利用するマルウェアで，8 検体がプロキシ認証突破するマルウェアであることを確認した．

M3AS はサンドボックス数の拡充と解析精度にトレードオフの関係があり，サンドボックスの拡充（精度向上）がコスト高騰に直結するため，構成するサンドボックスの個数に対する全体顕現率をより大きくするサンドボックス選定方法の策定に取り組んだ．複数のサンドボックス間で

の顕現マルウェアの重複を考慮したサンドボックスの選定方式を 2 方式提案した．1 つめの最大顕現器選択方式は，少ないサンドボックス数でより顕現率の高いサンドボックスの組合せを選定する目的で利用するのに有用であることを確認した．次に重複排除選択方式は，最大顕現器選択方式よりも少ない数のサンドボックスで全検体を顕現させることができる手法として有効であることを確認した．評価の結果，76 個のサンドボックスで構成される M3AS によってこれまでに解析した中に，顕現した検体（不正なネットワーク通信を行う検体）は 2,117 件存在したが，これらの顕現マルウェアのすべてを顕現させるのに，最大検知器選択方式の場合は合計 30，重複排除選択方式の場合は合計 27 のサンドボックスがそれぞれ必要であることを確認した．これにより，重複排除選択方式を採用した場合，M3AS を構成するサンドボックスの数を，65%削減できることを確認した．

さらに，M3AS をベースに不正サイトを解析するシステムを提案し，解析対象サイトへのアクセスに用いるブラウザ等の環境に応じて応答を変化させる不正サイトの解析を行った．InternetExplorer や Chrome，Firefox，Safari といったブラウザの様々なバージョンをインストールしたサンドボックスから構成されるシステムを用いて，実際の不正サイトと思しき 2,439 サイトを用いて評価した．その結果，不正サイトの中には解析環境の違いにより応答するコンテンツを変化させるものが存在し，単一の解析環境では平均 17.8%（最大 25.8%）の不正サイトを見逃していることを明らかにした．M3AS はマルウェアなどのファイルの解析だけではなく，URL 等のサイトの解析にも有用であることを確認した．

課題 2 では，インターネット隔離環境では顕現しない機能の把握するため，インターネットエミュレーション環境を使った閉塞環境で解析していた前記 M3AS を拡張し，限定的にインターネットに接続させて解析精度を向上させる手法を提案した．本拡張では，インターネットに接続して新たなマルウェアをダウンロードして攻撃者の本来の目的を実行するようなダウンローダ型マルウェアによるダウンロード通信を検出し，当該ダウンロード通信のみをインターネットに接続させることにより，マルウェアによる外部への攻撃を抑制しつつマルウェア解析を行うマルウェア通信制御手法を実現しプロトタイプシステムを開発した．また，代表的なダウンローダ型マルウェアを用いた評価実験により，644 検体の実検体のうち，58 検体がダウンローダ型マルウェアであること，また，解析を通じて外部への攻撃が発生しなかったことを確認した．以上の結果より，本提案手法によって，外部への攻撃を行うことなく，組織に侵入したマルウェアの特性を解明できることを明らかにした．

課題 3 では，課題 1，2 で抽出したマルウェアの接続先（不審 URL）に基づくマルウェアによる不正なアクセスの遮断を実現するために，M3AS やインテリジェンスサービスベンダが提供する不審な URL（グレーリスト）へ端末がアクセスする際に，プロキシに CAPTCHA を用いて追加認証を加えることで，マルウェアによる機械的なアクセスのみ遮断する自律進化型防御システ

ム（AED）を提案した．M3AS 等から得た 52,653 種類のユニークな不審 URL が含まれるグレーリストから構成される AED を用いて評価した結果，2,064 種類のマルウェアの 99%以上の遮断に成功した．また，事前に把握している不審 URL に基づく遮断では，未知（未解析）のマルウェアへの効果が期待できないことから，前記不審 URL に付随する特徴（AS 番号やステータスコード等）を決定木学習させて，未知の URL を特徴から不審推定する機能を実装し，99%近い精度で追加認証すべき URL の推定に成功することを確認した．

課題４では，誤った情報による対策であったとしても業務上必要なアクセスは許可するため，課題３と同様に CAPTCHA 認証を用いることで，M3AS の解析結果に誤った情報（例えば www.google.com）等が含まれていたとしても，ユーザの意図的な追加認証（CAPTCHA）への入力により，業務への影響を抑えられることを確認した．さらに，追加認証の頻出も利用者へ負担をかけることから，利用者の認証成否等の結果からグレーリストを，認証をせずに遮断するブラックリストや認証をせず通過させるホワイトリストへと振り分けることにより認証頻度を抑制する機能を実装した．本機能の評価の結果，グレーリスト 52,653 件のうち，実験中 9 日間にユーザが実際にアクセスした URL は 147 件，それらの URL に対してブラックあるいはホワイトに分別されたグレーリストは 28.5%程度であった．これにより，運用をしていくことでグレーリストが分別され認証頻度が低下することを確認した．また，実証実験に参加した被験者に対して実施したアンケート結果では，本機能の導入により不便を感じていると答えた被験者は 0 であることも確認した．AED は，不確実な情報でも業務に影響を与えづらい自動対策できる点で，現場と経営者との意思決定時間短縮を実現する．また，運用を継続することで URL リストやルールが自律的に進化するため，ネットワーク効果や運用コスト低減が期待できる．例えば複数の組織にそれぞれ導入した AED 間でグレーリストや認証ステータスを共有することにより，特にばらまき型メール攻撃のような組織を跨る攻撃に対して集団防御の効果を発揮することができると考える．

これらの４つの課題を解決するための手法提案，実装，実験評価を通し，環境選択型マルウェア等の高度化するマルウェアを自動解析して対策に有効な特徴を抽出することで目的１を達成し，また，この特徴に対策に悪影響を与える情報が含まれていたとしても業務への影響を最低限に抑えて迅速，自動的に対策に適用することで目的２を達成した．

以上，本論文では目的１，２を技術によって解決する手法を提案した．

一方，技術ではなく人材育成によって解決する試みもなされている．情報通信処理機構 IPA の試算によると，国内の従業員 100 人以上の企業における情報セキュリティに従事する現状の技術者数は約 23 万人，不足人材数は約 2.2 万人，また既存の技術者約 23 万人中，必要なスキルを満

たしていないと考えられる人材は 14 万人弱と推計されている[105]．この状況を打破すべく，日本経済団体連合会が約 30 社からなる「サイバーセキュリティに関する懇談会」を 2014 年に設立し，さらに 2015 年には重要インフラ企業を含む各業界の大手企業も参画する「産業横断サイバーセキュリティ人材育成検討会」が発足し，本検討会ではセキュリティ人材育成に関する提言も行っている．政府もサイバーセキュリティ戦略のもと，教育や演習の充実を図るとともに，国家資格である情報処理安全確保支援士の整備もすすめてきた．

著者も，全てのサイバー攻撃が技術によって解決できるものではなく，人と技術が相互補完的に高度化，連携していくことが重要であると考える．本論文で提案したシステムやそれによって生成されるデータが，サイバー攻撃への対策力強化や人材育成の一助になることを願っている．

## 参考文献

[105] 独立行政法人 情報処理推進機構 : 情報セキュリティ人材の育成に関する基礎調査 -調査報告書-，入手先< https://www.ipa.go.jp/files/000014184.pdf>(参照 2016-11-24)

# 9. 実績

## 9.1. 学術論文

| No. | 発表年月 | 「タイトル」，単著・共著の別，掲載誌等・巻・号（発行所・出版社等），頁 |
|---|---|---|
| 1 | 2016年9月 | 「マルウェア解析向け通信制御システムの開発」，共著（重本倫宏，仲小路博史ほか4名），情報処理学会論文誌第57巻第9号（情報処理学会），pp. 2012-2020 |
| 2 | 2016年9月 | 「サイバー攻撃の侵入経路を考慮したセキュリティリスク評価技術」，共著（杉本暁彦，仲小路博史ほか1名），情報処理学会論文誌第57巻第9号（情報処理学会），pp. 2077-2087 |
| 3 | 2015年9月 | 「多種環境マルウェア動的解析システムの提案及び評価」，共著（仲小路博史，菊池浩明ほか4名），情報処理学会論文誌第56巻第9号（情報処理学会），pp. 1730-1744 |

## 9.2. 翻訳・書評・作品等

| No. | 発表年月 | 種　類 | 「タイトル」，単著・共著の別，掲載誌等・巻・号（発行所・出版社等），頁 |
|---|---|---|---|
| 4 | 2014年5月 | 技術論文誌 | 「Automatic Malware Analysis Technology to Defend against Evolving Targeted Attacks」，共著（仲小路博史，鬼頭哲郎ほか3名，Hitachi Review VOL. 63 No. 5, pp. 80-86 |

## 9.3. 学会発表

| No. | 発表年月 | 「タイトル」，単独発表・共同発表の別，発表学会，開催地 |
|---|---|---|
| 5 | 2016年10月 | 「セッショングラフ構造を用いた不審アクセスログの分析に関する検討」，共同（発表代表者：林直樹），マルウェア対策研究人材育成ワークショップ／コンピュータセキュリティシンポジウム，秋田キャッスルホテル（秋田） |
| 6 | 2016年10月 | 「多種環境を用いた不正サイトの解析」，共同（発表代表者：重本倫宏），マルウェア対策研究人材育成ワークショップ／コンピュータセキュリティシンポジウム，秋田キャッスルホテル（秋田） |
| 7 | 2016年9月 | 「Proposal and Evaluation of Cyber Defense System using Blacklist Refined Based on Authentication Results」，共同（発表代表者：仲小路博史），Network-Based Information Systems，Technical University of Ostrava（チェコ・オストラヴァ） |
| 8 | 2016年7月 | 「サイバーセキュリティ脅威対策のためのビジネスリスク評価システムの提案」，共同（発表代表者：磯部義明），コンピュータセキュリティ研究会，中市コミュニティホールNac（山口） |
| 9 | 2016年1月 | 「人間行動を用いた自律進化型防御システムの提案」，共同（発表代表者：仲小路博史），暗号と情報セキュリティシンポジウム，ANAクラウンプラザホテル熊 |

| No. | 発表年月 | 「タイトル」，単独発表・共同発表の別，発表学会，開催地 |
|---|---|---|
| | | 本ニュースカイ（熊本） |
| 10 | 2015年10月 | 「環境選択型マルウェア解析システムの機能向上に向けたサンドボックスの構成検討」，共同（発表代表者：徳山喜一），マルウェア対策研究人材育成ワークショップ／コンピュータセキュリティシンポジウム，長崎ブリックホール（長崎） |
| 11 | 2015年5月 | 「マルウェア解析向け通信制御システムの開発」，共同（発表代表者：重本倫宏），コンピュータセキュリティ研究会，別府国際コンベンションセンター（大分） |
| 12 | 2015年5月 | 「マルチモーダルマルウェア解析システムを用いたプロキシアクセス型マルウェアの解析結果の考察」，共同（発表代表者：下間直樹），コンピュータセキュリティ研究会，別府国際コンベンションセンター（大分） |
| 13 | 2014年10月 | 「複数の解析環境から取得したマルウェアの振る舞い情報の非類似性尺度に関する検討」，共同（発表代表者：林直樹），マルウェア対策研究人材育成ワークショップ／コンピュータセキュリティシンポジウム，札幌コンベンションセンター（北海道） |
| 14 | 2014年10月 | 「多種環境マルウェア動的解析システムの提案」，共同（発表代表者：仲小路博史），マルウェア対策研究人材育成ワークショップ／コンピュータセキュリティシンポジウム，札幌コンベンションセンター（北海道） |

# 謝辞

本論文は著者が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において，菊池研究室において行った研究をまとめたものです.

本研究に関して終始ご指導ご鞭撻を頂きました本学菊池浩明教授に心より感謝致します. また，本論文を精読いただき有益で建設的なご議論を頂いた本学二宮広和教授，齋藤孝道教授，池田幸太博士，立命館大学毛利公一教授に深謝致します．さらに，研究の各フェーズにて厳しくも示唆に富む温かいご指導を頂いた MIMS 前所長三村昌泰教授，萩原一郎所長に厚く感謝致します.

第 2 章から第 7 章記載の各システムの実装には，日立製作所システムイノベーション研究センタセキュリティ研究部の重本倫宏研究員，鬼頭哲郎研究員，林直樹研究員，下間直樹研究員にご協力いただきました．また，評価に用いるマルウェアの検体収集には同社 IT 統括本部 IT セキュリティ統括部の梅木久志氏にご協力いただきました．第 2 章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」の協力を得て実施しました．第 7 章の評価実験では同研究所セキュリティ研究部の皆様の本来の業務へ悪影響が出る可能性があるにもかかわらず多くの方にご協力いただきました．関係者の方々に心より感謝します.

在学期間中，研究内容について独特の発想でアイデア溢れる議論で新たな気付きを与えていただいた静岡大学西垣正勝教授，夜遅くまで熱く議論し，忌憚ない意見を頂いた明治大学菊池研究室の山口通智氏，新原功一氏に感謝致します.

本論文の一部は明治大学先端数理科学インスティテュート(MIMS) PhD プログラムの一環による研究奨励奨学金 A で実施した研究成果です．ご支援に感謝いたします。

最後になりますが，業務と学業を両立する中，最後まで私の研究にご理解ご協力をいただいた日立製作所の同僚の皆様、そして妻 亜希子，娘 侑花，研究中に誕生してくれた息子 知輝に心より感謝します．ありがとうございました.