

マルウェア対策ユーザサポートシステムの キューイングネットワークモデル

川口 信隆^{1,a)} 余田 貴幸¹ 山口 演己¹ 笠木 敏彦²

受付日 2012年2月9日, 採録日 2012年9月10日

概要: 今日, 日々数千から数万の新種のマルウェアが発生している. これにともない, シグニチャを用いたマルウェア検知方式では, シグニチャの更新が新種の発生頻度に追いつかなくなりつつある. この問題を解決するために, 動的解析によりマルウェアを検知して駆除する「マルウェア対策ユーザサポートシステム」の研究開発が進められている. 本稿では, マルウェア対策ユーザサポートシステムをキューイングネットワークモデルでモデル化する. そして, 大規模なマルウェア感染が発生した場合の, ユーザサポートシステムの性能を, コンピュータシミュレーションを用いて評価した.

キーワード: マルウェア, 動的解析, シミュレーション

Queuing Network Model of Anti-malware User Support System

NOBUTAKA KAWAGUCHI^{1,a)} TAKAYUKI YODA¹ HIROKI YAMAGUCHI¹ TOSHIHIKO KASAGI²

Received: February 9, 2012, Accepted: September 10, 2012

Abstract: With the increasing number of new malware species, traditional malware detection approaches relying on signature files are being less and less effective, since it is quite difficult for anti-virus vendors to keep up with the frequent appearance of new malware species. To address this problem, a system called Anti-malware User Support System, which detects malware files using dynamic analysis and removes them from user PCs, is developed. In this paper, we model this system as a queuing network model. Then, using this model, we evaluate the performance of Anti-malware User Support System against the large scale malware epidemics.

Keywords: malware, dynamic analysis, simulation

1. はじめに

今日, 日々数千から数万規模のマルウェアの新種が出現している. これにともない, シグニチャファイルを基にユーザ PC 内からウイルスを検知するアンチウイルスソフトは, シグニチャファイルの更新が新種マルウェアの出現頻度に間に合わず, 検知率が低下している.

一方, 近年では多数のセキュリティベンダや研究機関が, マルウェア動的解析システム [1], [2], [3], [6] や非マルウェアのホワイトリストデータベース [4] など独自のマルウェア

対策機能を開発して公開している. これらの中には最新のセキュリティ技術を用いた優れたものが多数ある. 特に, マルウェア動的解析システムは, マルウェアの挙動を基に検知を行うため, シグニチャファイルに依存せずに, マルウェアを発見することができる. しかし, 個々の機能のみでは包括的なマルウェア対策とはならず, セキュリティに関する知識が乏しい一般ユーザが活用するのは難しい.

このような背景を受けて, これらのマルウェア対策機能を連携させることで, 新種マルウェアの発見から駆除までの包括的なマルウェア対策を実現する「マルウェア対策ユーザサポートシステム」の研究・開発が推進されている.

マルウェア対策ユーザサポートシステム [14] では, マルウェア擬陽性ファイル検知機能を用いて, ユーザ PC 内から擬陽性ファイル (マルウェアである可能性があるファイ

¹ 株式会社日立製作所
Hitachi, Ltd., Yokohama, Kanagawa 244-0817, Japan

² KDDI 株式会社
KDDI Corporation, Chiyoda, Tokyo 102-8460, Japan

^{a)} nobutaka.kawaguchi.ue@hitachi.com

ル)を発見する。発見されたファイルは、既知マルウェア判定機能、マルウェア解析機能により解析される。ファイルがマルウェアと判断された場合、駆除ツール生成機能が駆除ツールを自動生成する。最後に、駆除ツールをユーザPC上で実行して、マルウェアの駆除を完了する。個々の機能は包括的対策には不十分であっても、本システムを介して複数の機能が連携することで、一般ユーザを対象とした包括的なマルウェア対策を実現することが可能となる。

本稿では、マルウェア対策ユーザサポートシステムをキューイングネットワークモデルでモデル化する。そして、大規模感染型マルウェア感染が発生した場合の、ユーザサポートシステムの検知・駆除性能を、コンピュータシミュレーションを用いて評価する。これまでの既存研究の多くは、端末がマルウェアに感染してから駆除されるまでの時間を確率的に求めてきたが[9], [10], [11], [21], [22], [23], 本稿ではキューイングネットワークモデルに基づき、解析待ち時間や解析に費やされるリソース量などを考慮して算出する。

特に、本モデルでは、マルウェア検知・駆除に必要な処理を複数の独立した処理に分割してモデル化することで、各処理の設定や並列数などがシステム全体のパフォーマンスに与える影響を正確に求めることができる。本稿は、マルウェア動的解析・駆除ツール生成に関する主要な技術を扱った、5種類のモデルを提示している。

最後に、評価結果を基に、マルウェア対策ユーザサポートシステムが大規模感染型マルウェアに対して有効であることを示した。

今後、ますます高度化するマルウェアへの対抗策として、マルウェア動的解析システムの利用は必須と考えられる。しかし、著者らが知る限りマルウェア動的解析システムを用いた検知・駆除対策のモデル化・性能評価を行った研究は、これまでにない。

以下、2章では本稿の関連研究を、3章では、マルウェア対策ユーザサポートシステムの概要を説明する。続いて、4章ではマルウェア対策ユーザサポートシステムの基本モデルについて述べる。5章では基本モデルを基に導出された5種類のモデルに対する、シミュレーション評価を行う。6章を本稿のまとめとする。

2. 関連研究

2.1 マルウェア対策

複数のマルウェア対策機能を組み合わせることで、高度なマルウェア対策を実現する手段に関する既存研究は数少ない[5], [7]。CloudAV[5]は、一般的なアンチウイルスソフトや動的解析システムなどの複数のマルウェア検知機能を統合して検知を行うプラットフォームである。CloudAVでは、ユーザPCを監視し、アクセスが発生したファイルを解析対象ファイルとして、解析システムに送信する。解

析システムでは、複数のマルウェア検知機能を用いてファイルを分析する。そして、分析結果を統合して、最終的な検知結果を求める。

しかし、CloudAVでは、ファイルがマルウェアと判断された場合にも、駆除ツールは生成されないため、マルウェアに対する包括的対策を実現していない。また、擬陽性ファイルを発見する機能を有しておらず、PC中の全ファイルが解析対象となり、解析システムやネットワークに大きな負荷がかかるという問題がある。

2.2 マルウェアの感染シミュレーション

インターネットや大規模ネットワークを対象とした、マルウェアの感染シミュレーションモデルは、Epidemiological Model[8]を代表に、様々に提案されている[9], [10], [11]。これらのモデルでは、感染端末数やセキュリティパッチが適用される端末数の時間変化を、微分方程式や差分方程式で表現する。本稿では、マルウェアの感染シミュレーションモデルとして、Analytical Active Worm Propagation (AAWP)モデル[9]を用いた。AAWPモデルは感染端末数の時間推移を離散的に求めるため、マルウェアが駆除された端末数の時間推移を離散的に求めるユーザサポートシステムのキューイングネットワークモデルとの組合せが容易なためである。

また、数百から数千規模の端末を対象としたマルウェア対策手法を、シミュレーションにより評価した研究としては文献[12], [13]などがある。

しかし、著者らが知る限り、マルウェア対策ユーザサポートシステムのように、マルウェア動的解析システムを含むマルウェア対策方式のシミュレーションモデルを構築し、大規模感染型マルウェアに対する性能評価を行っている研究はこれまでにない。これまでに研究されている、マルウェア駆除をとともう感染シミュレーションの多く[9], [10], [11], [21], [22], [23]では、検知・駆除時間を、確率的に求めている(ある1秒間に駆除される確率 = 0.001など)。これに対して本稿で提案するモデルは、キューイングネットワークモデルを導入し、検体の解析待ち時間や解析に費やされるリソース量などを考慮することで、マルウェア対策にかかる時間を、より正確に算出することができる。

3. マルウェア対策ユーザサポートシステム

マルウェア対策ユーザサポートシステムは、様々なマルウェア対策機能を連携させることで、マルウェアの検知から駆除までの包括的なマルウェア対策を行う。図1に本システムの概要を示す。

本システムは、マルウェア対策に必要な手順を4種類のマルウェア対策機能(マルウェア擬陽性ファイル検知機能、既知非マルウェア判定機能、マルウェア解析機能、駆

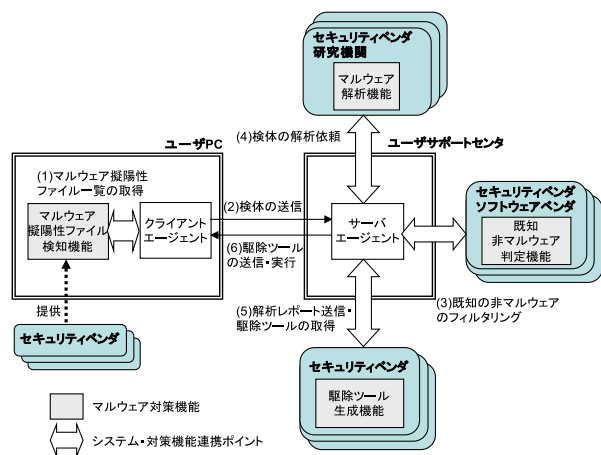


図 1 マルウェア対策ユーザサポートシステムの概要

Fig. 1 Overview of Anti-malware User Support System.

除ツール生成機能)に分割する。

マルウェア擬陽性ファイル検知機能は、ユーザ PC 内を探索し、マルウェアである可能性があるファイル（擬陽性ファイル）を発見する [14]。具体的には、新規プロセスが起動する際に、その実行ファイルの内容を検査し、ファイルがマルウェアである可能性があるかどうかを判定する。判定では、マルウェアの見逃しを防ぐため、検査対象ファイルがマルウェアに見られる特徴を少しでも有する場合は、擬陽性ファイルと判断する。たとえば、マルウェアは静的解析を防ぐため、パッカを使いファイルの中身を圧縮・難読化する。擬陽性ファイル検知機能は、検査対象のファイルからパッカに固有な名称を持つセクションを発見した場合、ファイルを擬陽性ファイルと判断する。一方で、ファイルに信頼できる認証局から発行された、有効な AuthenticCode 署名が付与される場合などは、ファイルがマルウェアである可能性は低いと判断する。

既知非マルウェア判定機能は、与えられた検体が、既知の非マルウェアであるか否かを判定する。判定には、既知の非マルウェアのハッシュ値が格納されたホワイトリスト DB [4] などを用いる。

マルウェア解析機能は、サポートセンタから受信した検体を解析環境内で実行する [1], [2], [3], [6]。解析環境は、検体が実行される計算機と、検体のネットワーク活動を再現するためのネットワークエミュレータなどから構成される。解析環境には、検体が呼び出したシステムコールや送受信パケットを記録する機構が備えられており、検体の挙動を記録する。そして、実行開始から数分内の挙動記録を基に、検体がマルウェアであるか否かを判定する。判定では、検体の振舞いが、マルウェアに固有なものであるか、あるいは、非マルウェア（正規アプリケーション）の振舞いから大きく外れていないかをチェックする [3]。

判定完了後、マルウェア解析機能は、検体の挙動や判定結果などの一連の解析結果を解析結果レポートとして、サ

ポートセンタに返答する。最後に、検体が実行された計算機を実行前の状態に復旧する。

駆除ツール生成機能 [18] は、解析結果レポートを基に、ユーザ PC からマルウェアを駆除するための、駆除ツールを生成する。駆除ツールは、パターンファイルと駆除エンジンから構成される。パターンファイルは、どのファイルやレジストリを削除するのかといった、駆除手順を指定する。駆除エンジンは、パターンファイルに従い、ユーザ PC 上で駆除処理を行う。駆除ツール生成機能は、駆除ツールのパターンファイルを生成する。駆除エンジンはあらかじめユーザ PC 上にインストールされている。

一般的に、マルウェアに感染したことが判明したユーザ PC はネットワークから即時切り離し、被害拡大を防止することが重要である。しかし、警告を PC 画面に表示してもユーザが気付かない、または業務上の都合などが原因で切り離しが難しい場合がある。本システムでは、駆除ツールによって駆除処理を自動化することで、上記の場合であっても被害拡大を防止することができる。

また、これらの機能を連携させるために、クライアントエージェント (CA)、サーバエージェント (SA) という 2 つの機能を設ける。CA はユーザ PC 上で、SA はシステムを統括するユーザサポートセンタ上で動作する。ユーザサポートセンタは、組織ネットワークや ISP が自網内の端末を守るために設置したり、アプリケーションサービスプロバイダがセキュリティサービスとしてインターネット上の端末に提供したりするなど、様々な形態で導入されることを想定している。

マルウェア検知・駆除の手順を以下に示す。

- 最初に、ユーザはユーザサポートセンタから CA をダウンロードして、PC にインストールする。マルウェア擬陽性ファイル検知機能はプロセスの起動を監視し、擬陽性ファイルを発見すると、CA に渡す。
- CA はマルウェア擬陽性ファイルを検体として、SA に送信する。ファイルは送信前に、CA・SA 間の共通鍵または SA の公開鍵により暗号化される。また、検体と同時に、ユーザ PC 内の環境情報を送信する。環境情報には、ユーザ PC にインストールされている OS やアプリケーションの種類やバージョンなどが含まれる。
- 検体と環境情報を受信した SA は、検体を復号して既知非マルウェア検知機能に送信する。既知非マルウェア判定機能は、検体が既知の非マルウェアであるか否かの判定結果を出力する。

既知非マルウェア検知機能が検体を非マルウェアと判定した場合、SA は検体のファイル名とハッシュ値を CA に通知する。CA は、SA から通知されたファイル名とハッシュ値をローカルホワイトリストに保存する。ローカルホワイトリストは、ユーザ PC 内にある

既知非マルウェアのファイル名とハッシュ値の一覧を保持する。リストに含まれるファイルは以後 SA に送信されなくなる。以上で、サポートシステムでの処理は完了する。

4. 既知非マルウェア検知機能が、検体は非マルウェアと判定しなかった場合、SA は検体と環境情報をマルウェア解析機能に送り、検体の解析を依頼する。マルウェア解析機能は検体を解析し、検体の挙動に関する情報と検知結果を含む解析結果レポートを返答する。マルウェア解析機能が、検体はマルウェアではないと判定した場合、検体は既知非マルウェア判定機能と CA のローカルホワイトリストに登録され、サポートシステムでの処理は完了する。
5. マルウェア解析機能が、検体はマルウェアであると判断した場合、SA は検体と解析結果レポート、環境情報を駆除ツール生成機能に送信する。駆除ツール生成機能は検体と解析結果レポートを基に、駆除ツールのパターンファイルを生成して SA に返答する。
6. SA は駆除ツールのパターンファイルを CA に送信する。CA は駆除ツールエンジンを実行して、パターンファイルに記された駆除処理を行い、マルウェアを PC から駆除する。以上で、サポートシステムでの処理は完了する。

なお、実行ファイルをいっさい作成しないメモリ常駐型のマルウェアは、本システムの対象外とする。

4. シミュレーションモデル

本シミュレーションでは、Code Red や MSBlast など、多くのユーザ PC (以下、端末と表記) に感染するマルウェアが大量発生した場合に、マルウェア対策ユーザサポートシステムが、各端末に感染したマルウェアの検知・駆除を行うのにかかる時間を評価する。

4.1 マルウェアの感染モデル

シミュレーションでは、過去に出現したマルウェアの中でも、最大規模の感染を行った TCP ワームである Code Red の感染活動を模擬するマルウェアを、AAWP モデルに従って、モデル化した。

まず、Code Red の挙動や感染規模を基に、マルウェアの感染方法を表 1 のとおりに決定した。スキャンレートは既存研究 [9] のモデルで示された値を用いる。このモデルでは、時間経過によりマルウェアのスキャン活動が活発になった場合でも、ネットワーク帯域への影響が小さくス

キャンレートが減少しない環境を前提としている。

本シミュレーションでは、議論を簡単にするために、ユーザサポートシステムに参加していない端末がマルウェアに感染した場合は、検知・駆除などの対策はいっさい行われないものと仮定する。この感染モデルでは、時刻 $t+1$ 秒のときの

- マルウェアが攻撃可能な脆弱性を有する全端末のうち、まだマルウェアに感染していない端末数 $S(t+1)$
- マルウェアが攻撃可能な脆弱性を有する全端末のうち、マルウェアに感染している端末数 $I(t+1)$
- 脆弱性が修復され、マルウェアに感染しない端末数 $R(t+1)$
- $S(t+1)$ に該当する端末のうち、ユーザサポートシステムに参加している端末数 $S'(t+1)$
- $I(t+1)$ に該当する端末のうち、ユーザサポートシステムに参加している端末数 $I'(t+1)$
- $R(t+1)$ に該当する端末のうち、ユーザサポートシステムに参加している端末数 $R'(t+1)$ (仮定により、 $R'(t+1) = R(t+1)$)

は、時刻 t 秒から $t+1$ 秒の間にマルウェアが駆除されたが脆弱性を依然として有し、再感染する可能性がある端末数 $newS(t)$ (上記仮定により、すべてユーザサポートシステムに参加している端末)、時刻 t 秒から $t+1$ 秒の間にマルウェアが駆除され脆弱性が修復されて再感染しない状態に推移した端末数 $newR(t)$ (上記仮定により、すべてユーザサポートシステムに参加している端末)、アドレススキャンレート $scan$ を用いて

$$I(t+1) = I(t) + S(t) \left(1 - \left(1 - \frac{1}{2^{32}} \right)^{I(t) * scan} \right) - newS(t) - newR(t) \quad (1)$$

$$S(t+1) = S(t) * \left(1 - \frac{1}{2^{32}} \right)^{I(t) * scan} + newS(t) \quad (2)$$

$$R(t+1) = R(t) + newR(t) \quad (3)$$

$$I'(t+1) = I'(t) + S'(t) \left(1 - \left(1 - \frac{1}{2^{32}} \right)^{I(t) * scan} \right) - newS(t) - newR(t) \quad (4)$$

$$S'(t+1) = S'(t) * \left(1 - \frac{1}{2^{32}} \right)^{I(t) * scan} + newS(t) \quad (5)$$

$$R'(t+1) = R'(t) + newR(t) \quad (6)$$

と記述される。 $newS(t)$, $newR(t)$ を、4.2 節で述べるキューイングネットワークモデルを用いたシミュレーションにより算出する。そして、求められた値を式 (1)~(6) に代入

表 1 マルウェアの感染方法

Table 1 Propagation method of the simulated malware.

感染端末のスキャンレート	2.0/second [9]
スキャン方法	ランダムアドレススキャン [9]

することで、各時刻における、インターネット全体およびユーザサポートシステムに参加している端末における、脆弱端末数、感染端末数、復旧端末数を求める。なお、以後は、断りがない限り、脆弱端末数、感染端末数、復旧端末数は、ユーザサポートシステムに参加している端末を評価対象とする。

4.2 マルウェア対策ユーザサポートシステムの基本モデル

シミュレーションでは、マルウェア対策ユーザサポートシステムを、ユーザサポートセンタ、マルウェア解析機能、駆除ツール自動生成機能から構成される、キューイングネットワークモデルによってモデル化した。具体的には、これまでに提案されている、主なマルウェア動的解析・駆除ツール生成のアプローチを網羅した、以下の5種類のモデルを構築した。

- (1) SIS (Susceptible-Infected-Susceptible) モデル
- (2) SIR (Susceptible-Infected-Removed) モデル
- (3) SIR-MultiOS モデル
- (4) SIR-Similarity モデル
- (5) SIR-MultiAnalyzer モデル

(1) SIS モデルは、マルウェア駆除ツールがマルウェアを駆除した端末が、マルウェアに再感染する可能性がある場合のモデル (すなわち, $newR(t) = 0$) であり, (2) SIR モデルは、マルウェア駆除ルールがマルウェアを駆除した端末は、再感染しない場合のモデル (すなわち, $newS(t) = 0$) である. (3)~(5) は、SIR モデルを拡張したものである. (3) SIR-MultiOS モデルは、マルウェアの感染端末上での動作を高精度で再現するために [24], 異なる OS がインストールされた実行環境を保持し、感染端末に最も近い環境でマルウェアを解析する、マルウェア解析機能をモデル化する. (4) SIR-Similarity モデルは、検体の類似度 [16] に基づいて、今までに解析したことがない検体のみを解析するマルウェア解析機能をモデル化する. (5) SIR-MultiAnalyzer モデルは、複数のマルウェア解析機能の判定結果を基に、最終的なマルウェア判定を行う [5] システムをモデル化する.

各モデルを採用した理由について以下に述べる。

SIS モデルは，駆除ツールとして，文献 [18] に示すマルウェア関連ファイル・レジストリの駆除，プロセス・サービスの停止を行うツールを想定している．現在のマルウェア対策ユーザサポートシステムの実装に最も近いモデルとして採用している．

SIR モデルは文献 [18] の駆除ツールの機能が拡張され、マルウェアが利用する脆弱性を修復したりネットワークから端末を切断したりするなど、再感染の防止が可能になった場合のシステムをモデル化している。このモデルは、文献 [18] の駆除ツールが、より実用的なものに発展した場合を想定して、採用している。

SIR-MultiOS モデルは、マルウェアの挙動が実行環境に

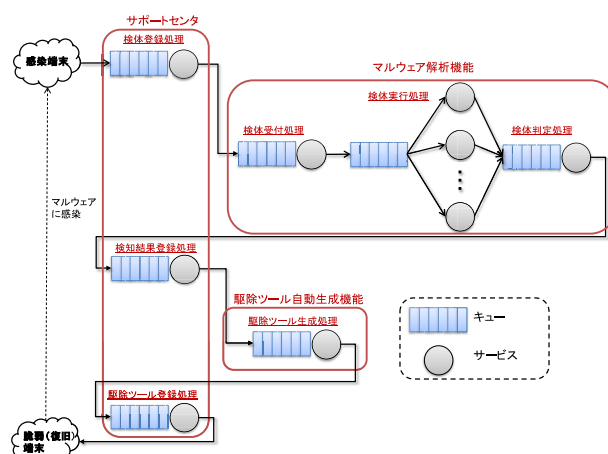


図 2 マルウェア対策ユーザサポートシステムのキューイングネットワークモデル

Fig. 2 Queuing network model of Anti-malware User Support System.

より大きく異なることが報告されている [24] 中で, マルウェア解析機能が感染端末と同じ OS を用いて解析を行うことを想定した場合の性能の変化を分析するために, 採用している。

SIR-Similarity モデルは、マルウェアの解析時間を短縮する手法が提案されている [15], [16] 中で、解析の短縮化がどの程度システムの性能に影響するのかを分析するために、採用している。

SIR-MultiAnalyzer モデルは、Cloud AV [5] のように、複数のマルウェア解析機能を用いて高精度な検知を実現する方式をマルウェア対策ユーザサポートシステムに適用した場合に、検知性能や検知・駆除時間にどのような影響が生じるのかを分析するために、採用している。

図 2 に、マルウェア対策ユーザサポートシステムの、キューイングネットワークの基本モデルを示す。上記 5 つのモデルは、この基本モデルを基にしている。

本モデルは、以下の7種類の処理から構成される。

(1) 検体登録処理

ユーザサポートセンタ内で実行される処理。端末から送信された検体をセンタに登録し、マルウェア解析機能に送信する。なお、既知非マルウェア判定機能も、この処理に含まれることとする。

(2) 検体受付処理

マルウェア解析機能で実行される処理、ユーザサポートセンタから取得した検体を検体実行処理のキューに登録する。なお、後述のとおり、検体実行処理は、複数の検体実行環境により並列に行われるが、断りがない限り検体実行処理のキューは1つとする。

(3) 検体実行処理

マルウェア解析機能内で実行される処理。検体を、検体実行環境内で実行して、挙動を記録する。挙動記録後は、実

行環境を元の状態に復旧する．このため，検体実行処理には，挙動記録処理と環境復旧処理の，2つの処理が含まれる．

なお，挙動記録処理完了後すぐに，検体は，後段の検体判定処理に回され，環境復旧処理はその後行われるものとする．検体実行処理には，通常，数分の時間を要し，解析機能の中で，最も負荷が大きい．このため，多くのマルウェア解析機能は，複数の実行環境を持ち，複数の検体の実行を並列で行う．

(4) 検体判定処理

マルウェア解析機能内で実行される処理．検体実行処理での検体記録を基に，マルウェア判定を行い，解析結果レポートを生成する．

(5) 検知結果登録処理

ユーザサポートセンタ内で実行される処理．検体の解析結果レポートをマルウェア解析機能から取得して，センタに登録する．検体がマルウェアである場合は，検体と解析結果レポートを駆除ツール自動生成機能に送信し，駆除ツール生成要求を行う．

(6) 駆除ツール生成処理

駆除ツール自動生成機能内で実行される処理．ユーザサポートセンタから取得した解析結果レポートと検体を基に，駆除ツールを生成する．

(7) 駆除ツール登録処理

駆除ツール自動生成機能内で実行される処理．駆除ツールを駆除ツール自動生成機能から取得し，センタに登録する．また，端末に対して駆除ツールを渡す．時刻 t 秒から $t+1$ 秒の間に駆除ツール登録処理を完了した検体数が， $newS(t)$ (SIS モデルの場合) または $newR(t)$ (SIR モデル，SIR-MultiOS モデル，SIR-Similarity モデル，SIR-MultiAnalyzer モデル) となる．

駆除ツール登録処理が完了した段階で，端末内のマルウェアは駆除されるものとする．なお，実際には，端末がマルウェアに感染してから，検体をサポートセンタに送信するまでに，数秒程度のインターバルが発生するが[14]，本シミュレーションではこの時間は，考慮しないものとする．(1)～(7)までの全処理にかかる時間を，「検知・駆除時間」と呼ぶ．

本シミュレーションでは，インターネットから直接攻撃可能な脆弱性を持つ端末が多数存在する組織ネットワークに，ユーザサポートセンタが導入されている状況を想定する．4.1 節に示したとおり，本稿では，最も代表的な感染活動形態であるランダムアドレススキャンを行う大規模感染型マルウェアを用いて，組織ネットワーク規模の端末群へのマルウェア対策手段としての，本システムの基本性能を評価する．

ランダムアドレススキャンはファイアウォールなどでフィルタリングされず，すべて攻撃先に到着することを想

表 2 シミュレーションの初期値

Table 2 Default parameters of the simulation.

検体登録処理		10 秒
検体受付処理		10 秒
検体実行処理	挙動記録処理	300 秒
	環境復旧処理	300 秒
検体判定処理		10 秒
検知結果登録処理		10 秒
駆除ツール生成処理		10 秒
駆除ツール登録処理		10 秒
システムに参加する脆弱端末数		1000 台
検体実行処理並列数		10 並列
マルウェア解析機能の検知率		100%
初期感染端末数		10 台

定している，このため，本シミュレーションで想定する環境は，実際のインターネット環境と比べて，防御側にとって不利なものとなる．また，マルウェアの中には，文献[11]に示されているような効率的なスキャンを行い，ランダムアドレススキャンを用いる場合と比べて数倍の速度で感染を拡大するものがある．このようなスキャンを行うマルウェアおよび，昨今増加している，Drive-by-Download 攻撃や，ソーシャルネットワークサービス・モバイルネットワークなどを介して感染活動を行うマルウェア[17], [20], [21]に対する性能評価は，今後の課題とする．

表 2 に，シミュレーションの初期値を示す．

各処理の所要時間は，文献[2], [14], [18]に示されている実測値を基に決定した．また，典型的な組織ネットワーク内の端末数が数百～数千である[12], [13]ことから，脆弱端末台数は1,000台とした．また，検体実行処理の並列数の初期値は，脆弱端末の台数の1%にあたる，10並列とした．また，検知率の初期値は100%とした．

Code Red に最終的に感染した感染端末数[9], [19]を基に，インターネット上の脆弱端末数を350,000台に設定した．初期感染端末の10台は，ユーザサポートシステムに参加していない脆弱端末349,000台(=350,000−1,000)の中から選択する．また，シミュレーションでは，マルウェア発生時に他の検体に対する処理がシステム内で行われていないことを想定する．

5. シミュレーションによる性能評価

図3に，5つのモデルにおける検知・駆除時間の平均値の比較を示す．各モデルは表2に示したデフォルトパラメータを用いている．SIR-MultiOS モデル，SIR-Similarity モデル，SIR-MultiAnalyzer モデルに固有のパラメータについては後述する．

図に示すとおり，SIS モデルにおける検知・駆除時間が54,000秒程度であるのに対し，SIR モデルでは11,000秒程度となっている．SIR モデルでは，マルウェアの再感染

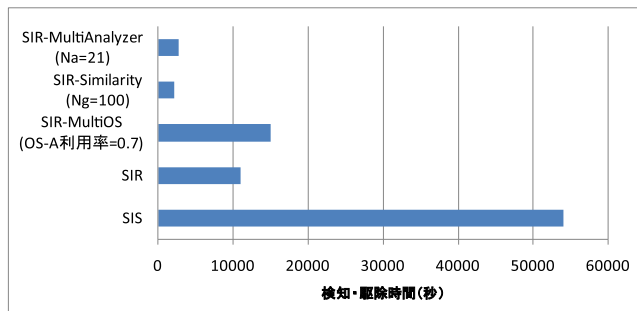


図 3 各モデルの検知・駆除時間の平均値の比較

Fig. 3 Comparison of detection-removal times for five models.

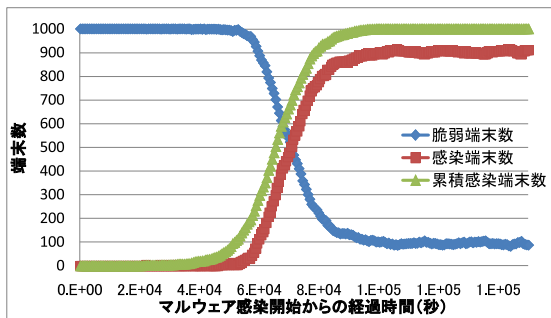


図 4 SIS モデルにおけるマルウェア感染の推移

Fig. 4 Malware propagation on SIS model.

が起きないため、SIS モデルに比べて、サポートセンタに送信される検体数が少なくなるためである。

SIR-MultiOS モデルは、SIR モデルよりも検知・駆除時間が長くなる傾向にある。これは、5.3 節で述べたとおり、感染端末で利用されている OS の種類の比率と、マルウェア実行環境で利用されている OS の種類の比率が異なる場合、実行環境の使用率に偏りが生じるためである。SIR-Similarity モデル、SIR-MultiAnalyzer モデルの検知・駆除時間は、SIR モデルの数分の一程度となる。SIR-Similarity モデルでは、受信した検体とすでに解析済みの検体の類似度を比較し、今まで解析していない検体のみ解析するため、SIR モデルと比べて、解析待ち時間が短くなる。SIR-MultiAnalyzer モデルは、複数のマルウェア解析機能を用いるため、各マルウェア解析機能の使用率が低くなり、SIR モデルと比べて解析待ち時間が短くなる。各モデルの詳細については、5.1 節以降で詳述する。

5.1 SIS (Susceptible-Infected-Susceptible) モデル

SIS モデルでは、駆除ツールによってマルウェアを駆除した端末は、再度マルウェアに感染する。このため、端末の状態は、Susceptible (脆弱性があるが感染はしていない状態) と Infected (感染している状態) の間を推移する。

図 4 に、SIS モデルにおけるマルウェア感染開始からの脆弱端末数、感染端末数、累積感染端末数の推移を示す。

マルウェア発生から約 100,000 秒 (約 28 時間) 後に、累積感染端末数 (過去にマルウェアに感染したことがある端

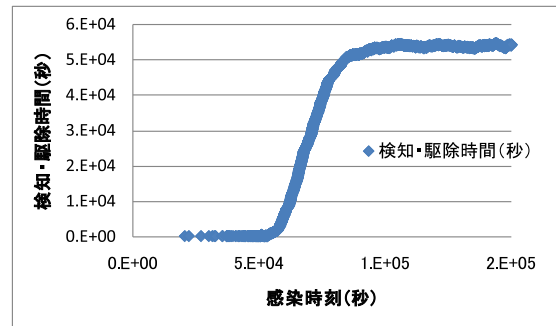


図 5 SIS モデルにおける検知・駆除時間の推移

Fig. 5 Detection and removal times on SIS model.

末数) は 1,000 台に達する。一方、脆弱端末数 (ある時刻に、マルウェアに感染していない端末数) は 100 台前後に収束する。

感染端末数が一定数に収束しているとき、単位時間あたりにマルウェアに感染する端末数と単位時間あたりにマルウェアが駆除される端末数は均衡する。時刻 t における、インターネット上の感染端末数を $I(t)$ 、ユーザサポートシステムに参加している脆弱端末数 (脆弱性はあるが、マルウェアには感染していない端末数) を $S'(t)$ とすると、単位時間あたりにマルウェアに感染する台数は、 $I(t) \times scan \times \frac{S'(t)}{2^{32}}$ に近似できる。ここで、十分な時間が経過した後は、インターネット上の脆弱端末は、駆除ツールによって一時的に駆除された端末を除いて、すべてマルウェアに感染していると考えてよいので、 $I(t) \sim 350,000$ となる。

一方、マルウェア解析機能内の実行環境数は 10 台であり、また、各実行環境の処理時間は、復旧時間も含めて 600 秒であるため、最大で、60 秒に 1 台の割合で、検体実行が完了する。ユーザサポートシステム内の他の処理は、すべて 10 秒で完了するため、検体実行処理が、システム全体のボトルネックとなる。このため、均衡状態では

$$350,000 \times 2 \times \frac{S'(t)}{2^{32}} = \frac{1}{60} \quad (7)$$

が成立する、これを解くと、 $S'(t) = 102$ となり、図 4 の値と一致する。

次に、図 5 に、端末がマルウェアに感染した時刻ごとの、検知・駆除時間の推移を示す。マルウェアの感染開始から検知・駆除時間は単調増加していく。感染活動開始から 10,000 秒経過以後にマルウェアに感染した端末からマルウェアを駆除するには、54,000 秒 (約 15 時間) 程度かかる。

次に、図 6 にマルウェア解析機能内の検体実行処理の並列数を変化させた場合の、均衡状態時の平均脆弱端末数を示す。検体実行処理の並列数が 60 を超えると、脆弱端末数は約 600 台に収束する。これは、並列数が 60 以上になると、10 秒に 1 台以上の速度で検体実行処理が完了するため、1 検体あたりの処理に 10 秒かかる他処理が、マルウェア駆除速度に影響することになるためである。

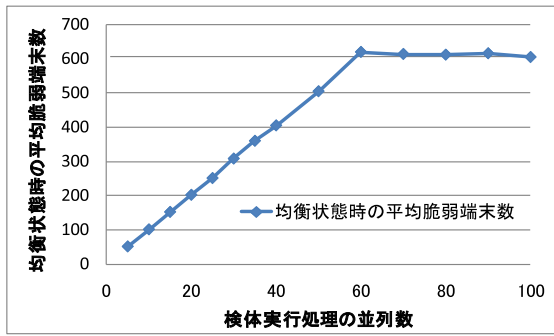


図 6 SIS モデルにおける平均脆弱端末数

Fig. 6 Convergence number of susceptible hosts on SIS model.

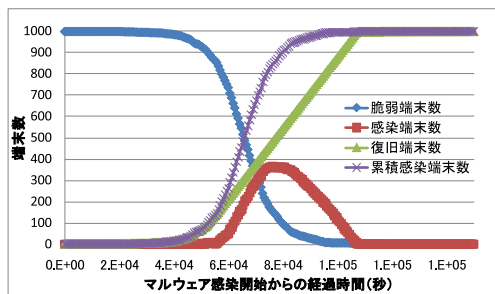


図 7 SIR モデルにおけるマルウェア感染の推移

Fig. 7 Malware propagation on SIR model.

5.2 SIR (Susceptible-Infected-Removed) モデル

SIR モデルでは、SIS モデルとは異なり、駆除ツールによってマルウェアを駆除した端末は、マルウェアに再感染しない。再感染の防止は、

- 駆除ツールに、脆弱性にセキュリティパッチを当てる機能がある、
- 駆除ツール実行後に、端末をネットワークから切断する、

のいずれかにより実現できる。どちらの場合も、シミュレーションの結果は同じになる。

SIR モデルでは、端末の状態は、Susceptible（脆弱性があるが感染はしていない状態）から、Infected（感染している状態）を経て、Removed（マルウェアが駆除され、再感染しない状態）に推移する。

図 7 に SIR モデルにおける、脆弱端末数、感染端末数、復旧端末数、累積感染端末数の推移を示す。脆弱端末数、累積感染端末数の推移は、SIS モデルの場合とほぼ同じである。累積感染端末数が、感染開始から 40,000 秒あたりから増加するのに対して、感染端末数は、感染開始から 50,000 秒を超えるまで 10 未満である。これは、この段階では、ユーザサポートシステムの駆除速度がマルウェア感染速度を上回るため、感染した端末はすぐに復旧されるためである。経過時間が 50,000 秒を超えたあたりから、マルウェアの感染速度は高速化し、感染端末数は増加する。72,000～85,000 秒（20～24 時間）あたりが感染端末数のピークであり、330～350 台の間で推移する。それ以降は、駆除速度が

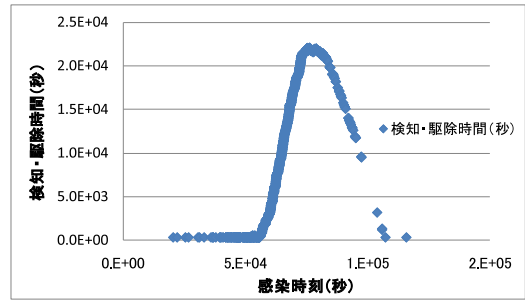


図 8 SIR モデルにおける検知・駆除時間の推移

Fig. 8 Detection and removal times on SIS model.

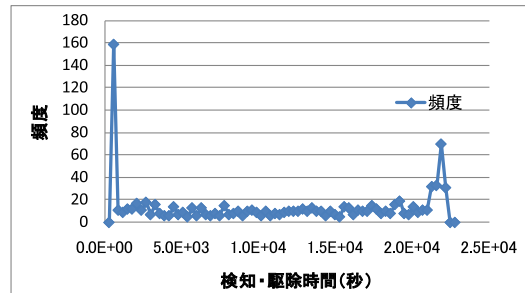


図 9 SIR モデルにおける検知・駆除時間の分布

Fig. 9 Distribution of detection and removal times on SIR model.

感染速度を上回るため、感染端末数は減少する。

次に、図 8 に、端末がマルウェアに感染した時刻ごとの検知・駆除端末時間の推移を示す。感染端末数と同様に、検知・駆除時間のピークは、感染開始から 72,000～85,000 秒の間であり、感染から駆除までに 20,000 秒（約 6 時間）程度かかることになる。以降は、新規感染端末数の減少にともない、検知・駆除時間は下降する。

また、図 9 に検知・駆除時間の分布を示す。分布は 2 つのピークを持つ。1 つは、システム内で処理待ちの検体数が少ない場合である。150 程度の端末に対しては、キュー内での待ち時間がほぼない状態で、検知・駆除が行われている。もう 1 つのピークは、マルウェアの感染速度とシステムの検知・駆除速度が均衡する場合にあり、検知・駆除時間が 20,000 秒となる端末数が 150 程度ある。このように検知・駆除時間の分布を詳細に分析するにはキュー内の待ち時間を考慮する必要がある。キュー内待ち時間を考慮した解析は従来の確率モデルでは困難であり、提案するキューイングネットワークモデルを利用する利点の 1 つである。

次に、図 10 に検体実行処理並列数を変化させた場合の、平均検知・駆除時間の推移を示す。検体実行処理並列数が初期値（＝10）の場合、平均検知・駆除時間は 11,000 秒（約 3 時間）程度となる。並列数の増加にともない、検知・駆除時間は急速に減少し、並列数が 30 を超えると、360 秒に収束し、キュー内での待ち時間はほぼ 0 となる。

以上より、SIR モデルでは、1 日以内に、ほぼすべての

表 3 SIR モデルにおける挙動記録処理時間の分布の影響

Table 3 Effect of distribution of behavior monitoring time on SIR model.

	定数	指数分布	一様分布
平均 (秒)	11011	11687	10945
標準偏差	7891	8283	8201

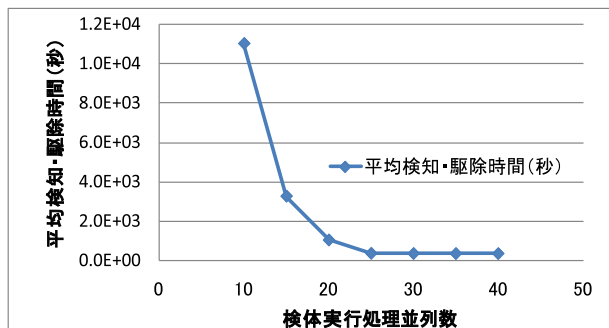


図 10 SIR モデルにおける検体実行処理並列数の影響

Fig. 10 Effect of the number of sample execution processes on SIR model.

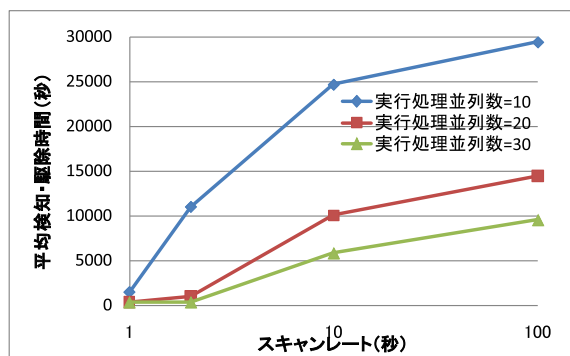


図 11 SIR モデルにおけるスキャンレートと平均検知・駆除時間の関係

Fig. 11 Effect of scanning rate on detection and removal times on SIR model.

脆弱端末に感染する大規模感染型マルウェアに対して、端末が感染してから平均 3 時間、最大 6 時間で、検知・駆除を行うことができる。このため、シミュレーションによる理論値と実環境における実測値との違いはあるものの、本システムは、一般的なアンチウイルスソフトのシグニチャが対応する*1よりも早く、マルウェア対策を実施することができる。

図 11 に、スキャンレートと平均検知・駆除時間の関係を示す。図が示すとおり、平均検知・駆除時間は、スキャンレートに比例して上昇する。これは、スキャンレートが高くなるほど、多数の脆弱端末が感染開始から短時間で感染するため、多数の検体が一挙にサポートセンタに送られ、

*1 文献 [5] によると、主要 10 種類のアンチウイルスソフトにおいて、マルウェアが出現してから 1 日後にシグニチャが配布される確率は 35～70%、1 週間後に配布される確率は、40～80%程度である。マルウェアの特性や流行の度合い、セキュリティベンダと顧客との契約形態によっては、発生から数分から数時間程度で、シグニチャが配布される場合もある。

キュー内で解析待ちになるためである。

図に示すように、実行処理並列数が 10 のとき、平均検知・駆除時間は、30,000 秒程度に収束する。これは、最も時間がかかる検体実行処理が 600 秒、脆弱端末数が 1,000 端末、センタ内に解析待ち検体がない場合の検知・駆除時間が 360 秒であるため、スキャンレートが高くなり全脆弱端末がほぼ同時に感染するような場合、平均検知・駆除時間の最大値(収束値)の概算値が、 $\frac{1}{1,000} \sum_{i=1}^{1,000} \frac{(i-1)}{10} * 600 + 360 = 30,330$ 秒になるためである。平均検知・駆除時間が長くなるほど、端末がマルウェアに感染している時間が長くなるため、被害が大きくなる。また、平均検知・駆除時間の最大値は、検体実行処理時間に比例し、実行処理並列数に反比例する。よって、スキャンレートが高いマルウェアによる被害を抑制するには、検体実行処理時間の短縮または実行処理並列数の増加が必要になる。

次に、ユーザサポートシステム内の処理で最も時間がかかる挙動記録処理の処理時間が定数 (300 秒) ではなく、指数分布・一様分布に従う場合の、検知・駆除時間の平均値および標準偏差を表 3 に示す。一様分布の場合、処理時間は 0～600 秒の間のいずれかの値を同確率でとる。結果が示すように、分布の違いが検知・駆除時間の平均値・標準偏差に及ぼす影響は小さい、原因としては、マルウェアの感染速度には、時刻により大きな偏りがあるため、挙動記録処理の偏りによる影響が相対的に小さくなるためと考えられる。

5.3 SIR-MultiOS モデル

マルウェア解析機能の中には、複数種類の実行環境を持つものがある。これらの解析機能では、検体を送信した端末と最も構成に近い実行環境を選択し、マルウェアを解析する。端末の環境と実行環境が類似しているほど、マルウェアの挙動も類似するため [24]、解析結果レポートを基に生成される駆除ツールの質が高くなる。

SIR-MultiOS モデルは、OS が異なる複数種類の実行環境を持つマルウェア解析機能をモデル化する。SIR モデルとの違いは、検体取得処理と検体実行処理を接続するキューが、検体実行処理の OS の種類ごとに存在する点である。検体受付処理は、端末から送信された環境情報を基に、検体を追加するキューを選択する。

本シミュレーションでは、マルウェア解析機能上には、OS A または OS B 上でマルウェアを実行する、2 種類の検体実行処理が存在することを仮定する。各検体実行処理

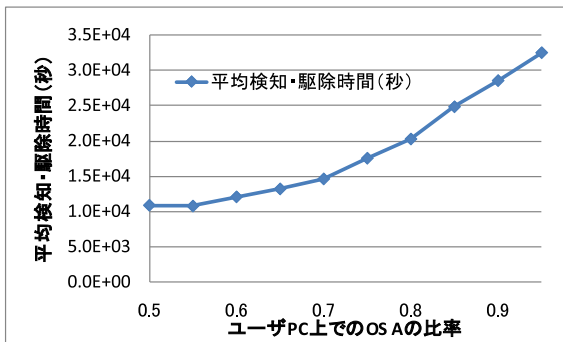


図 12 OS A の比率の影響

Fig. 12 Effect of the ratio of OS A.

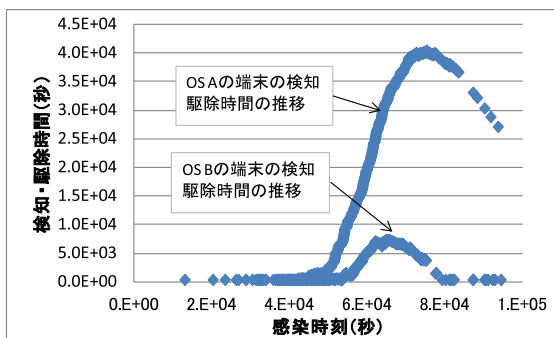


図 13 OS A の比率が 0.7 の場合の検知・駆除時間の推移

Fig. 13 Detection and removal times with the ratio of OS A = 0.7.

の並列数は等しく、それぞれ 5 とする。

図 12 に、OS A をインストールしている端末の比率を、0.5～0.95 の範囲で変化させたときの、平均検知・駆除時間の推移を示す。OS A の比率が 0.5、すなわち、OS A を利用している端末数と OS B を利用している端末数が等しい場合、平均検知・駆除時間は、SIR モデルの場合と同等の、11,000 秒程度となる。しかし、OS A を利用している端末の比率が高くなるにつれて、平均検知・駆除時間は長くなる。これは、処理負荷が OS A を扱う実行環境に偏り、OS B を扱う実行環境を十分に生かせないためである。

例として、図 13 に OS A の比率が 0.7 の場合の感染時刻と検知・駆除時間の推移を示す。平均検知・駆除時間は 15,000 秒であるが、検知・駆除時間の分布は、OS A と OS B で、大きく異なる。OS A の端末に対する検知・駆除時間は、ピーク時に 40,000 秒を超える。一方で、OS B の端末に対する検知・駆除時間のピークは 7,000 秒程度であり、OS A の場合の 1/6 程度である。また、感染活動を開始してから 80,000 秒を超えると、OS A の検体実行処理のキューには大量の検体があるのに対して、OS B の検体実行処理のキューはほとんど空になる。

以上から、ユーザサポートシステムに参加する端末における OS の比率と、実行環境における OS の比率が 20% でも異なると、検知・駆除時間が大きく増大することが分かる。このため、マルウェア解析機能が複数種類の OS を扱

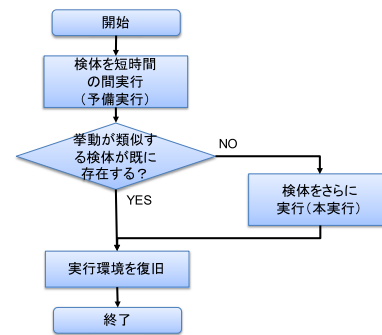


図 14 SIR-Similarity モデルにおける検体実行処理の流れ

Fig. 14 Flow of sample execution process on SIR-Similarity model.

う場合は、端末における OS の比率を反映させることが重要といえる。

5.4 SIR-Similarity モデル

マルウェア解析機能の中には、受け付けた検体を解析する前に、過去に解析した検体の中に同一のものが存在しないかを検査し、新しい検体である場合に限り、解析を行うものがある。重複した検体の解析を行わないことで、処理効率を向上させることができる。

ある検体が、すでに解析した検体と同一であるか否かを判定する手法には、静的類似度判定手法 [15] と動的類似度判定手法 [16] の 2 種類がある。

静的類似度判定手法は、検体のファイル構造を基に判定を行う。実際に検体を実行する必要がないため、演算負荷が小さく、高速に判定できるという利点がある。具体的な方法としては、検体のハッシュ値を計算し、過去に解析した検体に重複したものがあるか確認する [15]。しかし、マルウェアは、ファイルに無意味なデータを大量に追記することで、この手法を回避することが可能である。

動的類似度判定手法は、検体を短時間、実行環境内で実行し、その結果得られた挙動情報と、過去に解析した検体の挙動情報を比較する [16]。そして、類似度が高い挙動の検体があった場合、検体はすでに解析済みであり、これ以上の挙動解析は不要と判断する。この手法では、実際の挙動を基に判定を行うため、パッカや暗号化エンジンを用いて、ファイルごとのデータ構造を複雑に変える、ポリモルフィック・マルウェアに対応できる。

SIR-Similarity モデルでは、動的類似度判定手法を用いて、同一検体の判定を行うマルウェア解析機能をモデル化する。図 14 に、SIR-Similarity モデルにおける、検体実行処理の流れ [16] を示す。

検体を短時間 (予備実行) した後、挙動情報を基に、挙動が類似する検体がすでに存在するか否かを判定する。判定の結果、類似する検体がある場合は、実行を終了し、実行環境を復旧する。一方、類似する検体がない場合は、

表 4 SIR-Similarity モデルにおける検体実行処理のシミュレーションパラメータ

Table 4 Simulation parameters of sample execution process on SIR-Similarity model.

予備実行処理及び類似度判定処理	60 秒
本実行処理（挙動記録処理）	300 秒
実行環境復旧処理	300 秒

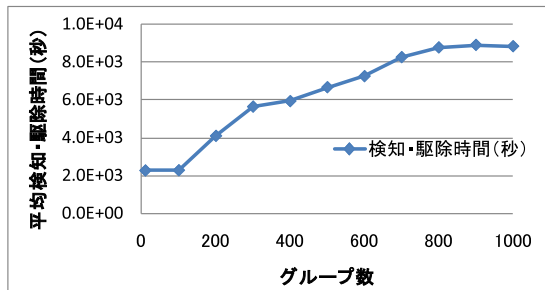


図 15 グループ数の平均検知・駆除時間への影響

Fig. 15 Effect of the number of groups on detection and removal times.

通常の処理（挙動記録処理）を行う．本シミュレーションの検体実行処理のパラメータを表 4 に示す．

予備実行処理時間は文献 [16] を基に，60 秒に決定した．検体がすでに解析済みの場合，検体実行処理は 360 秒（＝ 60 + 300）で完了する．一方，過去にない検体を解析する場合，検体実行処理は 660 秒（＝ 60 + 300 + 300）かかる．

次に，本シミュレーションで仮定する，動的類似度判定手法の精度について述べる．判定処理によって，あるマルウェアから派生する検体ファイルは，Group1～Group N_g まで， N_g 個のグループに分類されるものとする． $N_g = 1$ なら，検体ファイルはすべて 1 つのグループに分類される． $N_g = 2$ なら，検体ファイルは 2 つのグループに分類される．

マルウェア解析機能が k (≥ 1) 番目に受け付けた検体が，すでに解析済みである確率 $P_{\text{similarity}}(k)$ は，

$$P_{\text{similarity}}(k) = 1 - \left(1 - \frac{1}{N_g}\right)^{k-1} \quad (8)$$

となる．

図 15 に， N_g を 10 から 1,000 まで変化させた場合の平均検知・駆除時間の推移を示す． N_g が増えるほど，新たに本実行する必要がある検体数が増えるため，平均検知・駆除時間は増える傾向にある．しかし，グループ数が 100 までの範囲では，平均検知・駆除時間は，SIR モデルの場合の約 20% の，2,200 秒程度で推移している．このため，類似度判定手法の性能が，同一マルウェアの検体ファイルを 100 以下のグループに分類できるなら，ユーザサポートシステム全体として十分な性能を示すことができるといえる．文献 [16] によると，すでに解析した検体と新たに解析

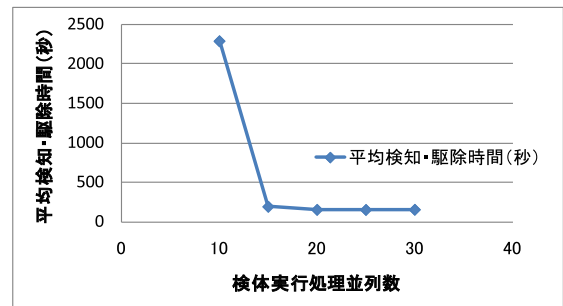


図 16 SIR-Similarity モデルにおける検体実行処理並列数の影響

Fig. 16 Effect of the number of execution processes on SIR-Similarity model.

した検体の挙動の違いを，どこまで許容するかによって， N_g は異なる．許容範囲が大きいほど， N_g は小さくなる．インターネット上の大量のマルウェアを収集・解析することを目的とする文献 [16] のシステムでは， $N_g \leq 2$ 以下となるパラメータを選択しているが，検体の挙動を駆除ツールの駆除手順に反映する本システムでは，許容される新検体と既検体の挙動の違いは，文献 [16] よりも小さい．このため，駆除ツールの質の点では， N_g は大きいほど好ましいが， $N_g = 100$ までなら $N_g = 2$ の場合と同等の検知・駆除時間を達成できる．また，この結果は，マルウェアがメタモルフィック性を持ち，実行ファイルによって挙動が変化する場合であっても，変化のバリエーションが 100 程度以下であるなら，検知・駆除時間を，SIR モデルの 20% 以下に減少させることができることを意味している．

次に，図 16 に， $N_g = 100$ のときに，検体実行処理並列数を変化させた場合の平均検知・駆除時間の推移を示す．並列数が 20 になると，待ち時間は 0 となり，平均検知・駆除時間は，SIR モデルの場合（＝ 360 秒）の 40% 程度である，150 秒に収束する．また，SIR モデルのとき，360 秒の平均検知・駆除時間を達成するには，検体実行処理並列数は 30 必要であるが，SIR-Similarity モデルでは，半数以下の並列数で，同等の検知・駆除時間を達成できていることが分かる．

5.5 SIR-MultiAnalyzer モデル

これまでのシミュレーションモデルでは，マルウェア解析機能の検知率を 100% と仮定してきた．しかし，実行環境の構成や検体の挙動記録方法や判定アルゴリズムによっては，マルウェアを見逃したり，反対に非マルウェアを誤検知したりする可能性がある．

解決策として，検知方法が異なる複数種類のマルウェア解析機能に検体を送信し，多数決により最終的な判定を行う，という方法が考えられる [5]．たとえば，検知率が 0.9，誤検知率が 0.1 のマルウェア解析機能が 3 種類ある場合，2 つ以上のマルウェア解析機能が，検体をマルウェアと判定したときに，ユーザサポートセンタが検体はマルウェアで

STEP.1	検体受付処理は、 $N_a(=2k+1)$ 個のマルウェア解析システムから $x=k+1$ 個を選択し、検体を送信する。STEP.2に移動。
STEP.2	検知結果取得処理は、 x 個の判定結果を取得する。STEP.3に移動。
STEP.3	これまでに取得した判定結果のうち、マルウェアと判定したものを N_d 、非マルウェアと判定したものを N_b とする。 $N_d \leq k$ かつ $N_b \leq k$ であるなら、STEP.4に移動。 $N_d > k$ であるなら、STEP.5に移動。 $N_b > k$ であるなら、STEP.6に移動。
STEP.4	未だ検体を送信していない、 $x=k+1-\text{MAX}(N_d, N_b)$ 個の解析システムに検体を送信する。STEP.2に移動。
STEP.5	検体はマルウェアであると判断し、駆除ツール生成処理に進む。
STEP.6	検体は非マルウェアであると判断し、処理を完了する。

図 17 SIR-MultiAnalyzer モデルにおける検体送信方法

Fig. 17 Transmission of samples on SIR-MultiAnalyzer model.

あると最終的に判断することで、検知率は 0.972 まで上昇し、誤検知率は 0.028 まで低下する。

SIR-MultiAnalyzer モデルでは、ユーザサポートセンタは、 $N_a = 2k + 1$ ($k = 0, 1, 2, \dots$) 個のマルウェア解析機能と接続し、図 17 に示すように、検体の送信および、マルウェア判定を行う。

本モデルでは、マルウェアは高度なポリモルフィック性、メタモルフィック性を備えており、同じマルウェア解析機能で、同一のマルウェアの検体ファイルを複数回解析した場合でも、検知結果は、検知率 P に従い、確率論的に決定されるものと仮定する。

このため、本モデルでは、ユーザサポートセンタにおけるマルウェア検知率 P_c は、

$$P_c = \sum_{i=k+1}^{2k+1} i \cdot C_k P^{k+1} (1-P)^{i-k-1} \quad (9)$$

となり、判定結果が出るまでに、検体を送信される解析機能の数 N_s の期待値 $E(N_s)$ は、

$$E(N_s) = \sum_{i=k+1}^{2k+1} i \times i \cdot C_k (P^{k+1} (1-P)^{i-k-1} + P^{i-k-1} (1-P)^{k+1}) \quad (10)$$

となる。

たとえば、3 種類のマルウェア解析機能が存在し、 $P = 0.9$ であるとき、 $P_c = 0.972$ となり、検知率は、1 台で解析する場合よりも 7.2% 向上する。また、 $E(N_s) = 2.18$ となり、ユーザサポートセンタにおいて最終的な判定結果が出るまでに、82% の確率で 2 種類の解析機能に、18% の確率で 3 種類すべての解析機能に、検体を送信される。

シミュレーション結果を以下に示す。なお、最新のマルウェアに対する、マルウェア動的解析システムの検知率が 90% 強である [3], [17] ことから、 $P = 0.9$ と仮定した。

図 18 に、 N_a を 1 から 21 まで変化させた場合の平均検知・駆除時間および、 P_c の理論値の推移を示す。マルウェア解析機能数の増大にともない、平均検知・駆除時間は低下する。理由は、 N_a の増加率と比べて、実際に検体を送信される解析機能数の期待値 $E(N_s)$ の増加率が低

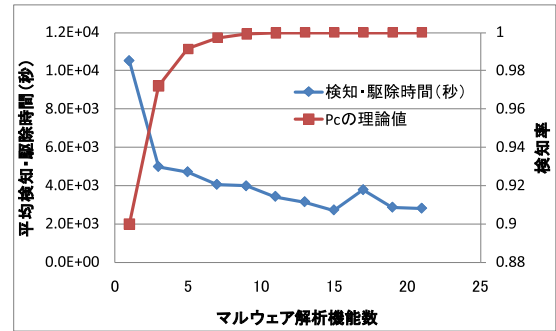


図 18 SIR-MultiAnalyzer モデルにおけるマルウェア解析機能数の影響

Fig. 18 Effect of number of malware analysis systems on SIR-MultiAnalyzer model.

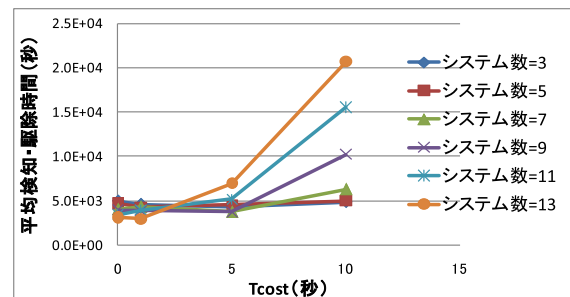


図 19 T_{cost} の平均検知・駆除時間への影響

Fig. 19 Effect of T_{cost} on detection and removal times.

いため、解析機能での待ち時間が短縮されるためである。たとえば、 $N_a = 1$ である場合、 $E(N_s)$ は 1 であるため、 $\frac{E(N_s)}{N_a} = 1$ となる。 $N_a = 3$ の場合、 $E(N_s) = 2.18$ となり、 $\frac{E(N_s)}{N_a} = 0.72$ となり、0.28 減少する。マルウェア解析機能数をさらに増やした場合、 $\frac{E(N_s)}{N_a}$ は、約 0.55 に収束する。すなわち、実際に検体を送信されるマルウェア解析機能数は、全マルウェア解析機能の半分程度となる。

$N_a = 9$ の場合、 P_c は 99.9% 以上となり、マルウェアに感染したにもかかわらず検知・駆除処理が行われない端末数は 1 台未満となる。このため、本シミュレーションでは、9 台以上のマルウェア解析機能と接続するのが好ましいといえる。

ただし、上記の平均検知・駆除時間の結果は、検体を送信するマルウェア解析機能数が増えた場合に、サポートセンタ側のコストが増大しないと仮定した場合である。図 19 に、検体の送信・検知結果レポートを受信する解析機能数が 1 台増えるたびに、検体登録処理および検知結果取得処理の処理時間が T_{cost} 秒増加すると仮定した場合の、平均検知・駆除時間の推移を示す。 T_{cost} が増加するにつれ、平均検知・駆除時間は増大する。また、マルウェア解析機能が多いほど、増加幅は大きくなる。

$T_{\text{cost}} = 10$ 秒、すなわち、検体登録処理・検知結果取得処理時間が、検体を送信するマルウェア解析機能数の倍数となる場合、 $N_a = 9$ のときの平均検知・駆除時間は約 10,000

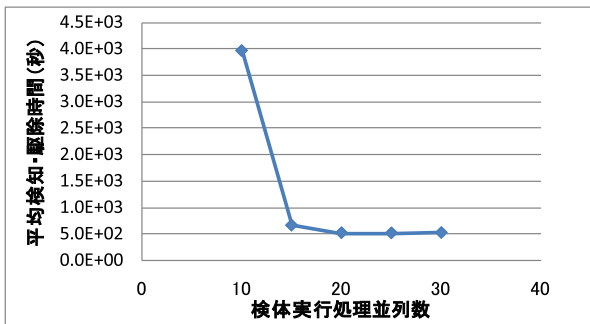


図 20 SIR-MultiAnalyzer モデルにおける検体実行処理並列数の影響

Fig. 20 Effect of the number of sample execution processes on SIR-MultiAnalyzer model.

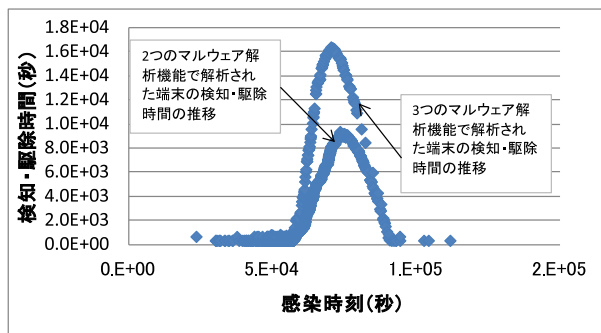


図 21 SIR-MultiAnalyzer モデルにおける検知・駆除時間の推移

Fig. 21 Detection and removal times on SIR-MultiAnalyzer model.

秒で, $N_a = 1$ の場合と同程度となる. 以上より, 平均検知・駆除時間を増やさずに, 未検知端末数を 1 未満とするには, $N_a = 9$ が適当であることが分かる.

次に, 図 20 に, $N_a = 9$ のときに, 検体実行処理並列数を変化させた場合の, 平均検知・駆除時間の推移を示す. 検体実行処理並列数が 20 以上になると, 平均検知・駆除時間は 510 秒に収束する. SIR モデルと異なり 360 秒に収束しないのは, 検体を複数のマルウェア解析機能に送信するという処理が, 複数回発生する場合があるためである (図 17, STEP 4). このため, SIR-MultiAnalyzer モデルでも, サポートセンタ側の検体送信, 解析結果レポート受信処理のコストを無視できるなら, 検体実行処理並列数が十分大きいとき (30 以上) は, 図 17 の STEP 1 で, すべてのマルウェア解析機能に検体を送信する方が, 検知・駆除時間を短縮できる.

次に, 図 21 に, $N_a = 3$ の場合の検知・駆除時間の推移を示す. 分布は 2 つのグループに分かれる. ピークが低い分布は, サポートセンタが最初に検体を送信した 2 つのマルウェア解析機能の判定結果が一致した場合の分布である. ピークが高い分布は, 最終的に 3 つすべてのマルウェア解析機能に検体が送信された場合を示している.

6. まとめと今後の課題

本稿では, マルウェア対策ユーザサポートシステムをキューイングネットワークモデルでモデル化し, 大規模なマルウェア感染が発生した場合のユーザサポートシステムの性能をコンピュータシミュレーションにより評価した. 著者らが知る限り, 本研究は, 大規模感染型マルウェアに対する, マルウェア動的解析システムを用いた検知・駆除対策の性能評価を行った, 初めての試みである.

我々は, システムの性能評価を通じて, 以下のような知見を得た.

- SIS モデルでは, 収束後の脆弱端末数は, 検体実行処理の並列数に比例し, マルウェアのスキャン速度とインターネット上の感染端末数に反比例する.
- SIR モデルでは, マルウェアに感染してから, 平均 3 時間, 最大 6 時間で, 検知・駆除を行うことができる. このため, 一般的なアンチウイルスソフトが対応するより早く, マルウェア対策を実施できる. また, 検体実行処理の並列数が 30, すなわち, システムに参加する端末数の 3% 以上の場合, キュー内での待ち時間を 0 にできる.
- SIR-MultiOS モデルでは, 端末にインストールされている OS の比率とマルウェア解析機能の検体実行処理が扱う OS の比率の違いが, 検知・駆除時間に大きく影響する.
- SIR-Similarity モデルでは, 動的類似度判定手法が検体ファイルを 100 以下のグループに分類可能であれば, 平均検知・駆除時間を SIR モデルの 20% 程度に減らすことができる. また, SIR モデルと比べて, 半分以下の検体実行処理の並列数で, キュー内での待ち時間を 0 にできる.
- SIR-MultiAnalyzer モデルでは, 検体登録処理・検知結果取得処理の処理時間がマルウェア解析機能数に比例する場合, 9 つ程度のマルウェア解析機能に接続した場合に, 検知・駆除時間を増大させずに, 平均誤検知端末数 1 台未満を達成する.

シミュレーションにより得られた大規模感染型マルウェアが発生した場合の検知・駆除時間は, 実際にマルウェア対策ユーザサポートシステムを導入する際に, どの程度の規模・機能を有するシステムを構築すれば, 許容可能な時間内に対策の実施が完了できるのかを見積もるうえで有用である. たとえば, マルウェアに感染した場合の大きな脅威の 1 つは, バックドアを通じて攻撃者に PC を遠隔操作されて機密情報が漏洩することであるため, 感染から対策完了までにかかる時間を見積もり, 攻撃者の手が及ぶ前に駆除を完了するシステムを構築することは非常に重要である. 具体的には, 1,000 人規模のユーザを擁する SIR モデルのシステムで, 平均検知・駆除時間を 1,000 秒以内に抑

えたい場合は、図 10 より、検体実行処理並列数が 20 以上必要なことが分かる。

今後はシステムを実装・実証実験を行い、その結果とシミュレーション値との比較検討を行い、より精度が高いシミュレーションモデルを構築していく。特に、昨今増加している、Drive-by-Download 攻撃や SNS、モバイルネットワークを通じて感染するマルウェアに対する、本システムの有効性を検証していく。

また、組織ネットワークにユーザサポートシステムが設置されている場合、ある感染端末に配布された駆除ツールを同じマルウェアに感染したネットワーク内の他端末に再配布することで、駆除処理を高速化する方法が考えられる。この方法についても検討を進め、シミュレーションモデルを構築していく。

謝辞 本研究成果の一部は、独立行政法人情報通信研究機構の委託研究「マルウェア対策ユーザサポートシステムの研究開発」によるものです。

参考文献

- [1] Willems, C. et al.: Toward Automated Dynamic Malware Analysis Using CWSandbox, *IEEE Security and Privacy Magazine*, Vol.5, No.2 (2007).
- [2] Inoue, D. et al.: Automated Malware Analysis System and its Sandbox for Revealing Malware's Internal and External Activities, *IEICE Trans. Information and Systems*, Vol.E92-D, No.5 (2009).
- [3] Lanzi, A. et al.: AccessMiner: Using System-Centric Models for Malware Protection, *Proc. 17th ACM Conference on Computer and Communications Security* (2010).
- [4] National Software Reference Library, available from <http://nsl.nist.gov/>.
- [5] Oberheide, J. et al.: CloudAV: N-Version Antivirus in the Network Cloud, *Proc. 17th Usenix Security Symposium* (July 2008).
- [6] Norman Solutions: Norman sandbox whitepaper, available from http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf.
- [7] Virustotal, available from <http://www.virustotal.com/>.
- [8] Zou, C.C. et al.: On the performance of Internet worm scanning strategies, *International Journal on Performance Evaluation*, Vol.63, No.7, pp.700-723 (2006).
- [9] Chen, Z. et al.: Modeling the spread of active worms, *Proc. IEEE INFOCOM 2003* (2003).
- [10] Onwubiko, C. et al.: An improved worm mitigation model for evaluating the spread of aggressive network worms, *Proc. IEEE International Conference on Computer as a Tool 2005*, pp.1710-1713 (2005).
- [11] Zou, C.C. et al.: Worm propagation modeling and analysis under dynamic quarantine defense, *Proc. 2003 ACM Workshop on Rapid Malcode*, pp.51-60 (2003).
- [12] 稲場ほか：ダミーアドレスからのコネクショントレースバックによるワーム早期抑制手法, 情報処理学会論文誌, Vol.50, No.7, pp.1735-1744 (2009).
- [13] Kawaguchi, N. et al.: Hit-list Worm Detection Using Distributed Sliding Window, *IPSJ Journal*, Vol.52, No.4, pp.1717-1726 (2011).
- [14] 川口ほか：マルウェア対策ユーザサポートシステムを用いた CCC DATASet 2010 の解析, マルウェア対策研究人材育成ワークショップ 2010 予稿集 (2010).
- [15] Wicherski, G.: peHash: A Novel Approach to Fast Malware Clustering, *Proc. USENIX LEET'09* (2009).
- [16] Bayer, U.: Improving the Efficiency of Dynamic Malware Analysis, *Proc. ACM Symposium on Applied Computing* (2010).
- [17] Rieck, K. et al.: Cujo: Efficient Detection and Prevention of Drive-by-Download Attacks, *Proc. ACSAC'10* (2010).
- [18] 川口ほか：マルウェア解析システムを用いたマルウェア自動駆除手法の検討, 電子情報通信学会第 14 回 ICSS 研究会予稿集 (2011).
- [19] Zou, C.C. et al.: Code red worm propagation modeling and analysis, *Proc. 9th ACM Conference on Computer and Communications Security* (2002).
- [20] Yan, G. et al.: Malware Propagation in Online Social Networks: Nature, Dynamics, and Defense Implications, *Proc. ASIA CCS* (2011).
- [21] Zhu, Z. et al.: A Social Network Based patching Scheme for Worm Containment in Cellular Networks, *Proc. IEEE Infocom 2009* (2009).
- [22] 林 幸雄ほか：ネットワーク成長によるメール型ウイルスの再流行と重点的なハブの免疫化の効果, 情報処理学会論文誌, Vol.44, No.SIG14, pp.9-19 (2003).
- [23] 岡本 剛ほか：電子メールにより拡散するコンピュータウイルスの拡散モデルの解析, 信学論 D-I, Vol.J84-D-I, No.5, pp.474-482 (2001).
- [24] Martignoni, L. et al.: A Framework for Behavior-based Malware Analysis in the Cloud, *Proc. 5th Int'l Conference on Information Systems Security* (2009).



川口 信隆 (正会員)

2008 年 3 月慶應義塾大学大学院理工学研究科後期博士課程修了。博士(工学)。2008 年 4 月株式会社日立製作所に入社。ネットワークセキュリティの研究開発に従事。2008 年 IPSJ 論文船井若手奨励賞受賞。2011 年より情報処理学会グループウェアとネットワークサービス研究会運営委員。IEEE, ACM 各会員。

商品名称等に関する表示：

Windows は Microsoft Corporation の米国及びその他の国における登録商標または商標です。

本稿に記載されている会社名、製品名は、それぞれの会社の登録商標もしくは商標です。



余田 貴幸

1999年3月早稲田大学理工学部電子・情報通信学科卒業。2001年3月早稲田大学大学院修士課程修了(工学)。2001年4月日立製作所入社情報・通信プラットフォームグループ通信事業部に配属。2002年4月日立製作所公

共システム事業部にてセキュリティ関連システムのシステムエンジニアとして従事。



山口 演己

1998年(株)日立製作所入社。情報システム部でセキュリティ・ネットワーク等の運用管理を行った後、セキュリティ・トレーサビリティ事業部にてセキュリティ製品の開発やセキュリティシステムの構築に従事。



笠木 敏彦

1985年国際電信電話株式会社入社。無線・伝送システムの建設、保全業務に従事後、ISPにおけるネットワーク、セキュリティ対策およびサーバホスティングの業務を担当。現KDDI株式会社情報システム本部共通業務シ

ステム部所属。