Vol.2012-GN-83 No.1
2012/3/21

キューイングネットワークモデルを用いたマルウェア対策ユーザサポートシステムの性能評価

川口 信隆[†] 余田 貴幸[†] 山口 演己[†]
笠木 敏彦[††] 星澤 裕二[†††]

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．これに伴い，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，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シミュレーションを用いて評価した．

Evaluation of Anti-Malware User Support System using Queuing Network Model

NOBUTAKA KAWAGUCHI[†] TAKAYUKI YODA[†]
HIROKI YAMAGUCHI[†] TOSHIHIKO KASAGI[††]
YUJI HOSHIZAWA[†††]

With the increasing number of new malware species, traditional malware detection approaches relying on signature files are being 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 remove 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.

[†]株式会社日立製作所
[††]KDDI 株式会社
[†††]株式会社セキュアブレイン

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．これに伴い，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て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」[14]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], 本論文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．

以下，第 2 章では本論文の関連研究を，第 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，第 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．第 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．第 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 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できていない．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，高精度な感染シミュレーションモデルである Analytical Active Worm Propagation (AAWP)モデル[9]を用いた．

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 [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．

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，駆除ツール生成機能）に分割する．

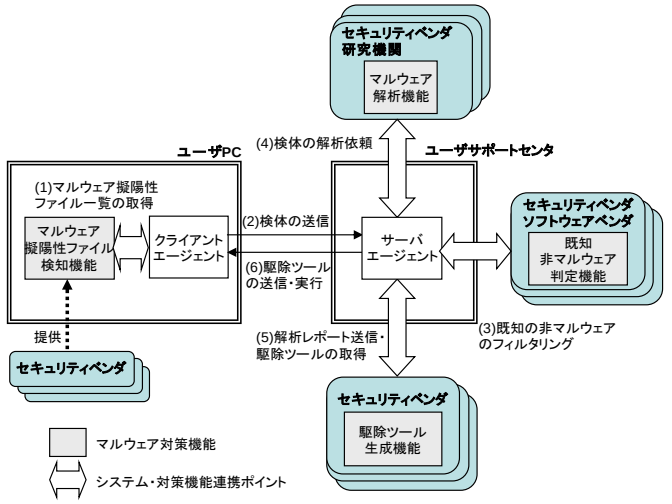

図 1 マルウェア対策ユーザサポートシステムの概要
Figure 1 Overview of Anti-Malware User Support System

マルウェア擬陽性ファイル検知機能は，ユーザ PC 内を探索して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．例えば，マルウェアは静的解析を防ぐためパッカー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ー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 AuthentiCode 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等から構成される．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，検体の振る舞いがマルウェアに固有なものであるか，或いは，非マルウェア（正規アプリケーション）の振る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[17]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．駆除エンジンは予めユーザ PC 上にインストールされている．

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．

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 MS Blast 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．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 の通りに決定した．

表 1 マルウェアの感染方法
Table1 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]

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 3 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-Similarity 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有る場合のモデル（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，再感染を行わない場合のモデル（すなわち，newS(t) = 0）である．(3) 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無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．

図 2 に，マルウェア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．上記のモデルは，この基本モデルを基に導出される．

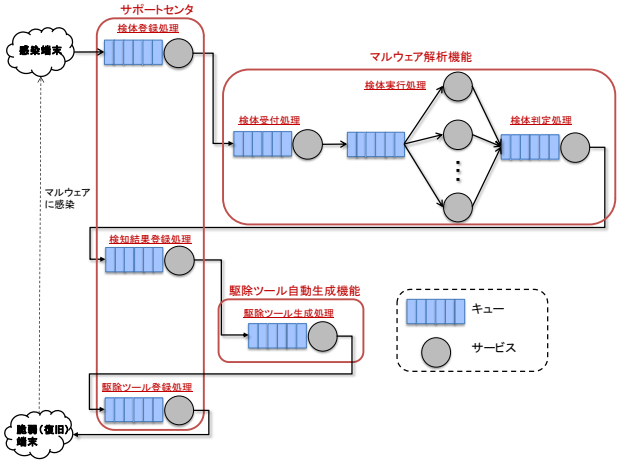

図 2 マルウェア対策ユーザサポートシステムのキューイングネットワークモデル
Figure 2 Queuing Network Model of Anti-Malware User Support System

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．尚，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．尚，後述の通り，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，断りが無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．

尚，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して行う．

(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．

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．尚，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．

ユーザサポートシステムのサーバ側設備であ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するなど，様々な形態で導入されることを想定している．

本シミュレーションでは特に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 に示した通り，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．表 2 にシミュレーションの初期値を示す．

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

各処理の所要時間は [2][14][17]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 1000 台とした．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．検知率の初期値は 100%とした．

CodeRed に最終的に感染した感染端末数[9]を基にインターネット上の脆弱端末数を 350000 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 349000 台(=350000－1000)の中から選択した．通常マルウェアのサイズは高々数百 KBytes であることから，検体のネットワーク伝送に関わる遅延は本シミュレーションでは考慮しない．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

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 3 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．

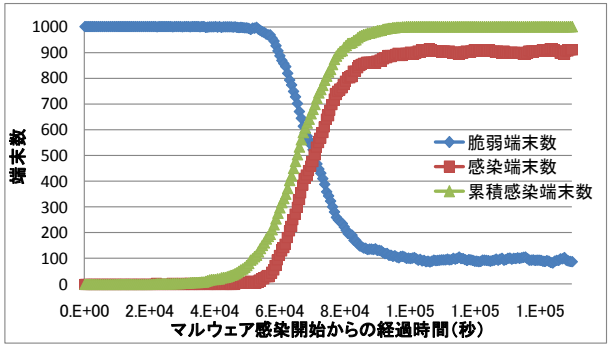

図 3 SIS モデルにおけるマルウェア感染の推移
Figure 3 Malware Propagation on SIS Model

マルウェア発生から約 100,000 秒(＝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末数）は 1000 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時，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，単位時間当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ほぼ全てマルウェアに感染していると考えていいので，I(t)~350000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，全て 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

$$350000 \times 2 \times \frac{S'(t)}{2^{32}} = \frac{1}{60} \quad \text{式(1)}$$

が成立する，これを解くと，$S'(t) = 102$となり，図 3 の値と一致する．

次に図 4 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(=15 時間)程度かかる．

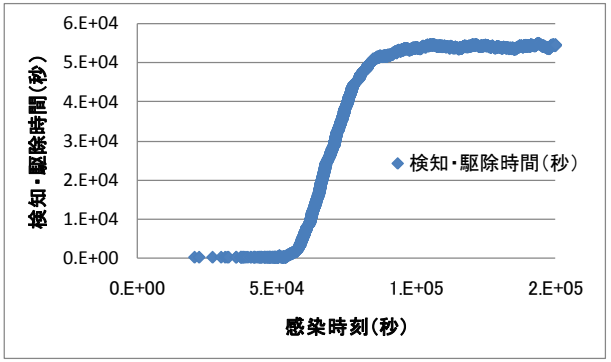

図 4 SIS モデルにおける検知・駆除時間の推移
Figure 4 Detection and Removal Times on SIS Model

5.2 SIR(Susceptible-Infected-Removed)モデル

SIR モデルは SIS モデルと異なり，駆除ツールによってマルウェアを駆除した端末は，再度マルウェアに再感染しないことを想定する．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

の何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 5 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 秒～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感染速度を上回るため感染端末数は減少する．

次に，図 6 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(＝3 時間)程度となる．並列数の増加に伴い検知・駆除時間は急速に減少し，並列数が 30 を超えたあたりで 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，ほぼ全ての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，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よりも早く，マルウェア対策を実施することができる．

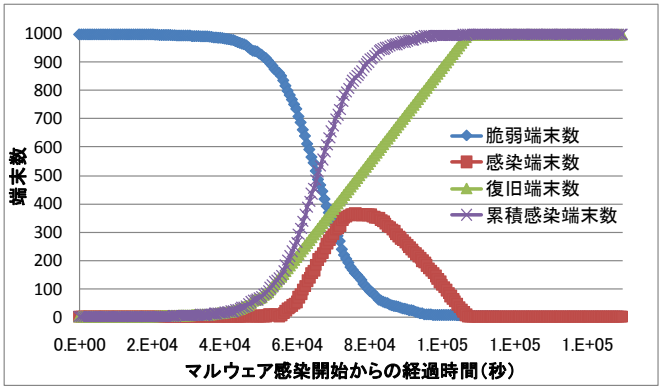

図 5 SIR モデルにおけるマルウェア感染の推移
Figure 5 Malware Propagation on SIR Model

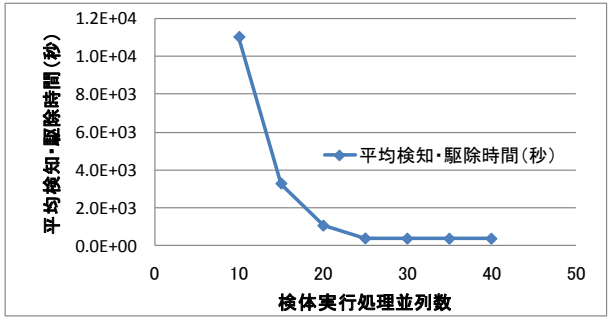

図 6 SIR モデルにおける検体実行処理並列数の影響
Figure 6 Effect of The Number of sample execution processes on SIR Model

5.3 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，処理効率を向上できる．

ある検体が，既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無いため演算負荷が小さく高速に判定できる利点がある．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と比較する[16]．そして類似度が高い挙動情報を持つ検体があった場合，検体は既に解析済みであると判断する．この手法は実際の挙動を基に判定を行うため，パッカーや暗号化エンジンを用いてファイル毎のデータ構造を複雑に変えるポリモフィック・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 7 に SIR-Similarity モデルにおける検体実行処理の流れ[16]を示す．

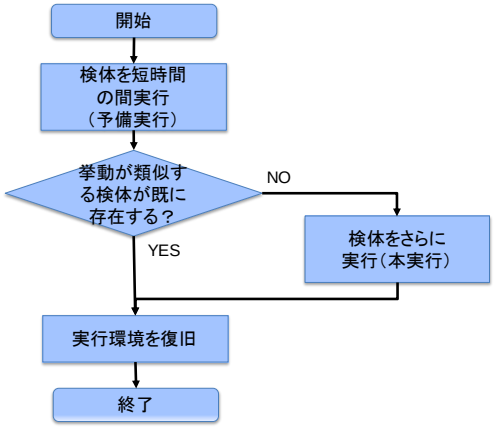

図 7 SIR-Similarity モデルにおける検体実行処理の流れ
Figure 7 Flow of Sample Execution Process on SIR-Similarity Model

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，類似する検体が既にある場合は，実行を終了し，実行環境を復旧する．一方，類似する検体が無い場合は，通常の処理（挙動記録処理）を行う．

次に，本シミュレーションにおける，検体実行処理のパラメータを表 3 に示す．

表 3 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 3 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

予備実行処理時間は[16]を基に，60 秒に決定した．検体が既に解析済みの場合，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～GroupN_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，検体ファイルは全て 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$k(\geq 1)$番目に受け付けた検体が，既に解析済みである確率$P_{similarity}(k)$は

$$P_{similarity}(k) = 1 - \left(1 - \frac{1}{N_g}\right)^{k-1} \qquad \text{式(2)}$$

となる．

図 8 に，N_gを 10 から 1000 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言える．[16]によると，N_gは，既に解析した検体と，新たに解析した検体の挙動の違いを，どこまで許容するかにより決まる．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 [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，[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%以下に減少できることを意味している．

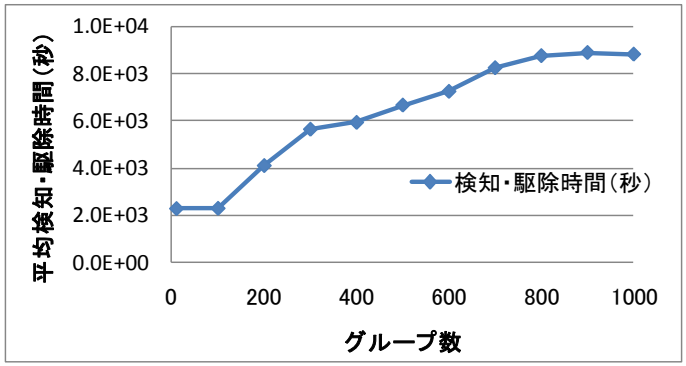

図 8 グループ数の平均検知・駆除時間への影響
Figure 8 Effect of the Number of Groups on Detection and Removal Times

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，即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-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．

今後は様々な条件下でシミュレーションを行うと共にマルウェア対策ユーザサポートシステムの実測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．

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] C. Willems, et al.,“Toward Automated Dynamic Malware Analysis Using CWSandbox,” IEEE Security and Privacy Magazine, Vol.5, Issue 2, 2007.
[2] D. Inoue, 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] A.Lanzi, et al., “AccessMiner: Using System-Centric Models for Malware Protection”, Proc. of 17th ACM Conference on Computer and Communications Security, 2010.
[4] National Software Reference Library, http://nsrl.nist.gov/.
[5] J. Oberheide, et al., “CloudAV: N-Version Antivirus in the Network Cloud”, In Proc. of the 17th Usenix Security Symposium, July, 2008.
[6] Norman Solutions. Normand sandbox whitepaper, http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf.
[7] Virustotal, http://www.virustotal.com
[8] C.C.Zou, et al., “On the performance of internet worm scanning strategies”, International Journal on Performance Evaluation, 63(7):700-723, 2006.
[9] Z.Chen, et al., “Modeling the spread of active worms”, Proc. of IEEE INFOCOM 2003, 2003.
[10] C.Onwubiko, et.al, “An improved worm mitigation model for evaluating the spread of aggressive network worms”, Proc. of the IEEE International Conference on Computer as a Tool 2005, pp.1710-1713, 2005.
[11] C.C.Zou, et al., “Worm propagation modeling and analysis under dynamic quarantine defense”, Proc. of the 2003 ACM Workshop on Rapid Malcode, pp.51-60, 2003.
[12] 稲場，他，“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 年.
[13] N.Kawaguchi, et al., “Hit-list Worm Detection Using Distributed Sliding Window”, IPSJ Journal, Vol.52, No.4, pp.1717-1726, 2011.
[14] 川口，他，“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 年.
[15] G.Wicherski, “peHash: A Novel Approach to Fast Malware Clustering,” Proc. of USENIX LEET’09 2009.
[16] U.Bayer, “Improving the Efficiency of Dynamic Malware Analysis”, Proc. of ACM Symposium on Applied Computing, 2010.
[17] 川口，他，“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”，電子情報処理学会第 14 回 ICSS 研究会予稿集，2011 年.