L-001

人の行動を考慮したマルウェア感染シミュレータの作成

A development of a simulator for malware infections depend on human behavior

麻生 大貴† 甲斐 博† 森井 昌克‡
Aso Hiroki Kai Hiroshi Morii Masakatsu

1.はじめに

標的型メール攻撃をはじめとして，マルウェア感染による組織内部の情報システムへの脅威は深刻さを増している．マルウェアに感染したホストが存在するネットワーク全体において，どのような影響があるかについて明らかにするため，マルウェア感染がもたらすネットワークの状況の変化とその影響，および被害を測るためのモデルが提案されている[1][2]．金岡らの NSQ モデル[1]では，マルウェア感染の原因となる人の行動を考慮せず，機器間での通信のみを対象としているため，シミュレータが表現できる状態が限られるという課題がある．広岡らのモデル [2]は人の行動を考慮しているが，モデルの提案のみであり，シミュレータは存在しない．

本論文では，人の行動を考慮したマルウェアの感染拡大をシミュレートするシステムの作成を目標とした．シミュレートに用いるネットワークのモデルは，NSQ モデル[1]を用い，人の行動を加えたマルウェアのモデルは広岡らが提案したモデル [2]を参考にした．各種モデルは XML により表現され，シミュレータおよび可視化は Java で実装した．本研究では，実行例を示すことにより，我々の作成 したシステムの有効性を示す．

2.ネットワークモデル

ネットワークシステムは，サーバやファイアウォール，ルータなどの複数の機能を有する多様な機器から構成される．その設計や構築にはコストや冗長性，セキュリティなどの性質が求められるが，これらを満たすためには小規模なシステムでさえその構成は複雑なものになる．本研究では，ネットワークシステムのモデル化に金岡らが提案した NSQ モデルを用いる．

2.1 NSQ モデル

NSQ (Networked system Security Quantification) モデルはネットワークシステムを通信層（レイヤ）毎に分解し，それぞれの通信層に依存するネットワーク機能をノードとして表現する．複数の通信層にまたがってノードを接続したものをモジュールと定義する．例えば一般的なネットワーク機器はモジュールとして表現することができる．ノードが同一レイヤ内でどのノードからアクセスされるか，上位層のノードがどの下位層のノードに依存して存在しているかをノード間の接続（リンク）として表す．ノード，モジュール，リンクにより，ネットワークシステム内のアクセス制御と依存性が表現可能となる．

金岡らが提案した NSQ モデルの実装について，公開された仕様はない．本論では NSQ モデルを用いてネットワークモデルを構築するために XML を用いて表現した（表 1）．

表 1 NSQ モデルを表現するための XML 要素

親要素	子要素
<node>	<id>, <layer>, <module>
<link>	<id>, <layer>, <module>, <startnode>, <endnode>
<module>	<id>, <name>, <port>, <connect>

親要素<node>，<link>，<module>はそれぞれ NSQ モデルでのノード，リンク，モジュールを表す．子要素の<id> は認識子である．<node>，<link> が持つ子要素<layer>，<module> はリンクが所属するレイヤ，モジュールを示す．<startnode>，<endnode>はリンクの始点と終点を示す．<module> 内の<name>はモジュール名，<port>は解放されたポート番号，<connect>は隣接して接続しているモジュールの id を表す．

2.2 マルウェアと人の行動のモデル化

ネットワークにおけるマルウェアの感染拡大の影響は，人間の行動に影響されない場合と影響される場合がある．人間の行動に影響されないマルウェアとして，例えば，W32.Blaster.Worm は，TCP ポート 69 番を利用して侵入し感染する．そして TCP ポート 135 番に接続しバッファをオーバーランさせる．その結果リモートシステムが TCP ポート 4444 番からローカルシステム権限で shell を得られるようになる．しかし，W32.Sobig.F は，電子メールの添付ファイルとしてマシンに侵入して感染し，大量のメールを発信する．25 番ポートや 110 番ポートが開いているモジュール上で，ユーザが添付ファイルをクリックすることで実行され発症する．この場合の発症条件として，「添付ファイルをクリックする」という人間の行動がある．

マルウェアの表現や人の行動の表現のため，次の表の XML をモデルに加えて実装している．

表 2 マルウェアと人の行動を表現する XML 要素

親要素	子要素
<malware>	<id>, <name>, <port>, <human>
<human>	<id>, <name>, <module>

親要素<malware>，<human>はそれぞれ人の行動，マルウェアを表す．子要素の<id>は認識子で，<name>は名前である．<malware>が持つ子要素<port>，<human> はそれぞれ感染拡大の際に用いられるポート番号，人の行動を表し，人の行動を表す<human> が持つ子要素<module>はどのモジュールでの行動かを示す．

† 愛媛大学 Ehime University
‡ 神戸大学 Kobe University

3.実装と動作確認

3.1 シミュレータの実装

これまでに示してきた，ネットワークやマルウェアなどモデルを XML で表記し，シミュレータに読み込ませてマルウェアの感染拡大状況を視覚的に確認できるように表示させる．そのために，Java を用いてネットワークシステムを記述した XML ファイルを読み込み，仮想的なネットワークを構成する．さらに，マルウェアの行動に従って，感染拡大の様子を描画する．

3.2 利用したネットワークモデル例

今回は感染拡大状況を予測しやすいように，簡易的なネットワークモデルを用いてシミュレートを行う．図 1 は 2 つのサーバ間にファイアウォールを挟んだだけのモデルを表現したものである．

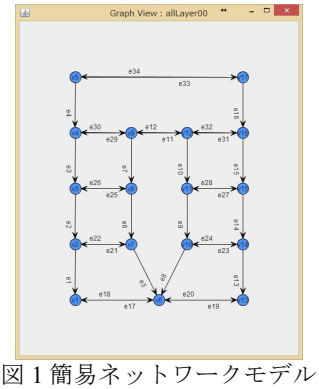

図 1 簡易ネットワークモデル

3.3 動作確認

シミュレータの動作を確認するために，2 種類のマルウェアを導入した際の動作を確認する．一つは W32.Blaster.Worm，もう一つは W32.Sobig.F を用いる．条件を次の表 2 および表 3 に示す．

表 2 W32.Blaster.Worm での動作条件

条件	TCP ポート	人の行動
A	69,135 が閉じられている	なし
B	69,135 が閉じられていない	なし

表 3 W32.Sobig.F での動作条件

条件	TCP ポート	人の行動
A	全て解放	なし
B	全て解放	メールの添付ファイルを開く

3.3 動作結果と考察

上で示した条件を元にシミュレータを動作させた結果を以下の図 2，図 3 に示す．初期感染ノードはいずれの場合も v4 とし，発症し感染した場合，ノードの色を変える（色が変わった個所を矢印で示すが，実際には色が変わるだけである）．

W32.Blaster.Worm の動作結果（図 2）は人の行動を考慮しないためマルウェアに感染してすぐに発症する．図 2 において条件 A と条件 B の画像を比較すると，感染の範囲が異なる．ファイアウォールにて感染を防げることが確認できた．

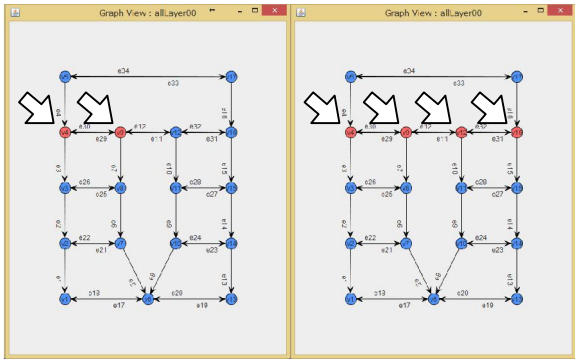

図 2 W32.Blaster.Worm の感染範囲
(左：条件 A，右：条件 B)

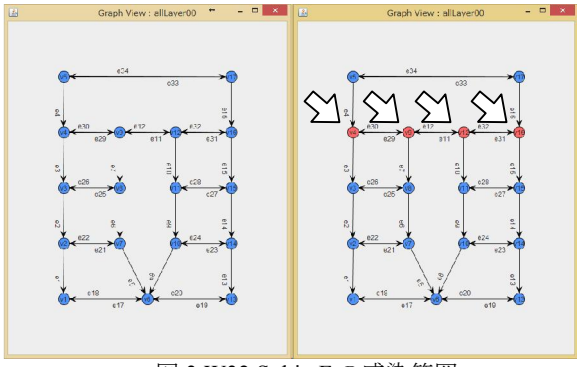

図 3 W32.Sobig.F の感染範囲
(左：条件 A，右：条件 B)

W32.Sobig.F の動作結果（図 3）では，条件 A の場合には人の行動がないのでどのノードにも感染していないが，条件 B の場合にはノード v4 で発症し感染拡大を行う．これにより人の行動によるマルウェアの感染をシステム上で確認することできた．

4.おわりに

本論文ではマルウェア感染したネットワーク上で人の行動がネットワーク上でどのような影響を及ぼすかを確認するため，NSQ モデルを基に，マルウェアと人の行動のモデルを追加したシミュレータを提案した．シミュレートした例題は小さいものだったが，マルウェアの感染拡大が容易に視認できるシステムが構築できた．シミュレートに用いることのできるモデルの種類を増やし，応用を拡げることが今後の課題である．

参考文献

[1] 金岡晃，原田敏樹，加藤雅彦，勝野恭治，岡本栄司，“安全なネットワークシステム設計のためのマルチレイヤネットワークモデルの提案と応用”，情報処理学会論文誌，Vol. 51，No. 9，pp. 1726-1735（2010）．
[2] 広岡俊彦，市川幸宏，白石善明，森井昌克，中尾康二，“ウイルスの挙動を解析するための実ネットワークを使った仮想感染ネットワークの設計”，コンピュータセキュリティシンポジウム 2004 論文集，pp. 433-438，(2004)．
[3] Symantec Security Response，http://www.symantec.com/security_response/（2015 年 6 月 1 日確認）