一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2016-250 (2017-03)

自律進化型ボットネットの感染抑制に関する考察

本行　航希† 　木村　共孝†† 　工藤　隆則††† 井上　文彰†††† 平田　孝志†

† 関西大学 システム理工学部 電気電子情報工学科 〒564-8680 大阪府吹田市山手町 3-3-35
†† 東京理科大学 工学部第一部 電気工学科 〒125-8585 東京都葛飾区新宿 6-3-1
††† 摂南大学 理工学部 電気電子工学科 〒572-8508 大阪府寝屋川市池田中町 17-8
†††† 大阪大学大学院 工学研究科 電子電気情報工学専攻 〒565-0871 大阪府吹田市山田丘 2-1
E-mail: †{k896955,hirata}@kansai-u.ac.jp, ††kimura@ee.kagu.tus.ac.jp, †††t-kudo@ele.setsunan.ac.jp,
††††y-inoue@post.comm.eng.osaka-u.ac.jp

あらまし　近年，機械学習の研究が盛んに行われている．機械学習は多くの分野で用いられるが，悪意ある攻撃者によってその手法が悪用された場合，従来とは比較にならないほどの甚大な被害がもたらされることが考えられる．そのような脅威として，自律進化型ボットネットの出現がこれまでに示唆されている．これは，ウイルスに感染したコンピュータの計算資源を利用した分散機械学習により，未発見の脆弱性を発見し自律進化するボットネットで，これまでの研究によりその高い感染力が示されている．本稿では，この自律進化型ボットネットの感染拡散対策として，マルコフ連鎖によって表現された複数のモデルを考え，それらの特性をシミュレーション実験によって明らかにする．
キーワード　ボットネット，ウイルス，機械学習，連続時間マルコフ連鎖

Considerations of suppressing the diffusion of self-evolving botnets

Kouki HONGYOU†, Tomotaka KIMURA††, Takanori KUDO†††,

Yoshiaki INOUE††††, and Kouji HIRATA†

† Faculty of Engineering Science, Kansai University, 3-3-35 Yamate-cho, Suita 564-8680, Japan
†† Faculty of Engineering, Tokyo University of Science, 6-3-1 Niijuku, Katsushika 125-8585, Japan
††† Faculty of Science and Engineering, Setsunan University,
17-8 Ikeda-nakamachi, Neyagawa 572-8508, Japan
†††† Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita 565-0871, Japan
E-mail: †{k896955,hirata}@kansai-u.ac.jp, ††kimura@ee.kagu.tus.ac.jp, †††t-kudo@ele.setsunan.ac.jp,
††††y-inoue@post.comm.eng.osaka-u.ac.jp

Abstract　Recently, machine learning has been extensively used and achieved significant results in many research areas. On the other hand, machine learning becomes a big threat when malicious attackers make use it for the wrong purpose. As such a threat, self-evolving botnets have been considered in the past. The self-evolving botnets autonomously predict vulnerability and evolve by performing machine learning with computing resources of zombie computers, which have high infectivity. In this paper, we consider several models of Markov chains to counter the spreading of the self-evolving botnets. Through simulation experiments, we show behaviors of these models.
Key words　Botnet, computer virus, machine learning, continuous-time Markov chain

1. はじめに

近年，コンピュータの性能上昇に伴い，機械学習の研究が盛んに行われている．機械学習は，検索エンジン，自然言語処理，音声認識や文字認識等の多様な分野に応用されている．加えて，静的コード解析及び機械学習によって，ソフトウェアのバグや脆弱性を発見するという研究も行われている [10], [12]．このような研究はソフトウェアの作成やその保護という観点から進められてきたものであるが，反対に悪意ある攻撃者がソフトウェアのバグや脆弱性を検出する手助けとなる危険性もある．また，機械学習の中には，深層学習 (Deep Learning) [3], [4] のように，ネットワーク内の複数の計算資源を用いて分散的に学習を行う

ものが存在し，これまで，多数の廉価なホストを用いて分散的に深層学習を行う手法が提案されている [3], [7], [11].

また，コンピュータウイルスは日々進化しており，変形性ウイルスのように，ホストに感染する度に自分自身のコードを書き換えることで難読化を行い，アンチウイルスソフトに検出されにくくするものや，既知のウイルスのコードを組み合わせることで新たなウイルスを生み出す手法が提案されている [1], [2], [8]．さらに，ウイルスに感染したホスト (ゾンビコンピュータ) 群がネットワークを形成し，攻撃者によって遠隔操作されることで，DDoS 攻撃等を行うようなボットネットが非常に大きな脅威となっている [9]．ボットネットは数十万から数百万台のゾンビコンピュータで構成されるような大規模なものも存在し，攻撃者がボットネットのゾンビコンピュータ資源を利用して，上述のような分散機械学習を悪用する状況が十分に予想される．

このような背景のもと, [6] において，ゾンビコンピュータ資源を利用した静的コード解析及び分散機械学習によりソフトウェアの脆弱性を自動的に検出し，それに合わせてコードを変異させ自律進化するような新種のボットネットの出現が示唆されている．本稿ではこのようなボットネットを自律進化型ボットネットと呼ぶ．自律進化型ボットネットは，自律的に変異を繰り返しながら進化するため，通常のウイルスに比べ感染力が非常に高いと考えられる. [6] では，自律進化型ボットネットの感染モデルが提案されており，マルコフ解析及びシミュレーション実験によりその脅威が示されている．

本稿では，この自律進化型ボットネットの感染拡散対策として，マルコフ連鎖によって表現された複数のモデルを考える．具体的には，ウイルスに対して免疫を持つホストが他のホストの保護を行うキルシグナル (Kill Signal: KS) モデル [5]，ボランティアのホスト群の計算資源を用いた分散コンピューティングにより未知の脆弱性を自律進化型ボットネットよりも先に発見し，それを塞ぐボランティアモデル，及びそれらを複合したモデルの検討を行う．本稿では，これらのモデルの特性をシミュレーション実験によって明らかにする．

2. 自律進化型ボットネット [6]

2.1 自律進化型ボットネットの感染モデル

[6] では，自律進化型ボットネットの脅威を明らかにするために，その挙動を表すウイルス感染モデルを提案している．このモデルではネットワーク内に存在する各ホストの状態を，図 1 に示される SIRS (Susceptible-Infected-Recovered-Susceptible) モデルにより表す．“S” はホストに何らかの脆弱性が存在する状態 (晒し状態) を，“I” はホストがウイルスに感染している状態 (感染状態) を，“R” はホストに既知の脆弱性が全く存在しない状態 (保護状態) を意味する．なお，保護状態は既知の脆弱性から保護されている状態であり，未知の脆弱性からは保護されていない．図 1 に示すように，晒し状態にあるホストは，ウイルスに感染する可能性があり，ボットネットによる攻撃により感染状態に移行する．また，晒し状態もしくは感染状態にあるホストは，OS のアップデートやウイルスの駆除等で脆弱性

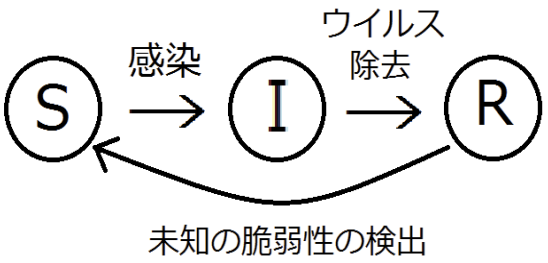

図 1　SIRS モデル

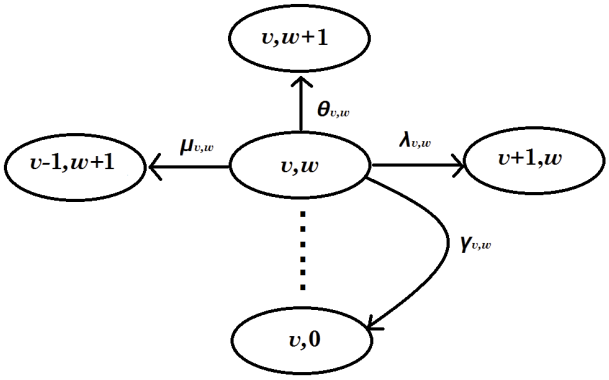

図 2　連続時間マルコフ連鎖上の (v, w) からの状態遷移

が取り除かれることで，保護状態に移行する．さらに，自律進化型ボットネットが，既知の脆弱性情報を用いた分散機械学習により新たな脆弱性を発見した場合，保護状態にあるホスト全てが晒し状態に遷移する．以下に [6] におけるウイルス感染モデルの動作をまとめる．

1)　感染ホスト同士でボットネットを形成することで，分散機械学習を行い，自動的に新たな脆弱性を発見する．感染ホストが多いほど，脆弱性発見率は高くなる．このとき，全ての保護状態のホストが晒し状態へ遷移する．

2)　晒し状態にあるホストは，一定確率で保護状態へ遷移する．

3)　感染ホストは一定確率でウイルスが取り除かれ，保護状態に遷移する．このとき，脆弱性は区別されず一度に全て取り除かれる．

4)　晒し状態にあるホストは，一定の確率でウイルスに感染し，ボットネットに組み込まれる．一方，保護状態にあるホストは感染しない．

2.2 連続時間マルコフ連鎖による表現

上記モデルにおいて各種イベントの発生がポアソン過程に従うと仮定し，それらの挙動を表す連続時間マルコフ連鎖を考える．ネットワーク内のホスト数を N 台，時刻 t において感染状態にあるホスト数を $V(t)$，保護状態にあるホスト数を $W(t)$ とする．このとき，晒し状態にあるホスト数は $N - V(t) - W(t)$ となり，マルコフ連鎖のシステムの状態を $(V(t), W(t))$ で表すことができる．図 2 に，自律進化型ボットネットに対するマルコフ連鎖の状態遷移図を示す．システムの状態が $(V(t), W(t)) = (v, w)$ にあるときの各状態への遷移率は以下の通りである (ただし，$v + w \leqq N$ である).

a) 晒し状態にあるホストがウイルスに感染し感染状態に移行すると，状態は $(v+1, w)$ に遷移する．このときの遷移率 $\lambda_{v,w}$ を以下のように与える．

$$\lambda_{v,w} = \alpha v(N - v - w) \tag{1}$$

ただし，α はホスト一台あたりのウイルスの感染率である．ウイルスは感染状態にあるホストから晒し状態にあるホストへ感染するため，$\lambda_{v,w}$ はそれらの組合せ数に従い決定される．

b) 感染状態にあるホストのウイルスが取り除かれ保護状態に移行すると，状態は $(v-1, w+1)$ に遷移する．このときの遷移率 $\mu_{v,w}$ を以下のように与える．

$$\mu_{v,w} = \delta_i v \tag{2}$$

ただし，δ_i はホスト一台あたりのウイルスの除去率である．

c) 晒し状態にあるホストの脆弱性が取り除かれ保護状態に移行すると，状態は $(v, w+1)$ に遷移する．このときの遷移率 $\theta_{v,w}$ を以下のように与える．

$$\theta_{v,w} = \delta_s(N - v - w) \tag{3}$$

ただし，δ_s はホスト一台あたりの保護率である．

d) 自律進化型ボットネットが新しい脆弱性を発見したときに，状態は $(v, 0)$ へ遷移する．このときの遷移率 $\gamma_{v,w}$ を以下のように与える．

$$\gamma_{v,w} = \eta(v+1) \tag{4}$$

ただし，η はホスト 1 台当たりの計算資源による新たな脆弱性発見率である．式 (4) に示されるように，脆弱性の発見率 $\gamma_{v,w}$ は感染ホスト数 v に比例し増加しており，これは，計算資源の増加による分散機械学習の能力の増加を意味する．

ここで，マルコフ連鎖の状態 (v, w) を

$$\{(0,0),(0,1),\ldots,(0,N) \mid (1,0),(1,1),\ldots,(1,N-1) \mid \ldots \\ \mid \ldots,(N-2,2) \mid (N-1,0),(N-1,1) \mid (N,0)\}$$

と並べ，$p_{v,w}(t) = \Pr(V(t) = v, W(t) = w)$ とする．このとき，$\boldsymbol{p}(t) = \{p_{v,w}(t)\}$ を時刻 t に各状態にある確率を表すベクトルとすると，

$$\frac{d\boldsymbol{p}(t)}{dt} = \boldsymbol{p}(t)\boldsymbol{Q}$$

となる．$\boldsymbol{Q}$ はマルコフ連鎖の推移速度行列であり，以下のように与えられる．

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{B_0} & \boldsymbol{C_0} & & & & \boldsymbol{0} \\ \boldsymbol{A_1} & \boldsymbol{B_1} & \boldsymbol{C_1} & & & \\ & \boldsymbol{A_2} & \boldsymbol{B_2} & \boldsymbol{C_2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \boldsymbol{A_{N-1}} & \boldsymbol{B_{N-1}} & \boldsymbol{C_{N-1}} \\ \boldsymbol{0} & & & & \boldsymbol{A_N} & \boldsymbol{B_N} \end{bmatrix}$$

ここで，$\boldsymbol{A_v}$，$\boldsymbol{B_v}$，$\boldsymbol{C_v}$ $(v = 0, 1, \ldots, N)$ は以下の式で与えられる．

$$\boldsymbol{A_v} = \begin{bmatrix} 0 & \mu_{v,0} & & & \boldsymbol{0} \\ & & \mu_{v,1} & & \\ & & \ddots & \ddots & \\ \boldsymbol{0} & & & 0 & \mu_{v,N-v} \end{bmatrix}$$

$$\boldsymbol{B_v} = \begin{bmatrix} \Lambda_{v,0} & \theta_{v,0} & & & \boldsymbol{0} \\ \gamma_{v,1} & \Lambda_{v,1} & \theta_{v,1} & & \\ \gamma_{v,2} & & \Lambda_{v,2} & \ddots & \\ \vdots & & & \ddots & \theta_{v,N-1-v} \\ \gamma_{v,N-v} & \boldsymbol{0} & & & \Lambda_{v,N-v} \end{bmatrix}$$

$$\boldsymbol{C_v} = \begin{bmatrix} \lambda_{v,0} & & & \boldsymbol{0} \\ & \lambda_{v,1} & & \\ & & \ddots & \\ & & & \lambda_{v,N-1-v} \\ \boldsymbol{0} & & & 0 \end{bmatrix}$$

ただし，$\Lambda_{v,k} = -(\gamma_{v,k} + \theta_{v,k} + \lambda_{v,k} + \mu_{v,k})$ $(k = 0, 1, \ldots, N-v)$ とし，$\gamma_{v,0} = 0$，$\theta_{N,0} = 0$，$\lambda_{N,0} = 0$，$\mu_{0,k} = 0$ である．また，$\boldsymbol{A_v}$ は $(N+1-v) \times (N+2-v)$ 行列，$\boldsymbol{B_v}$ は $(N+1-v) \times (N+1-v)$ 行列，$\boldsymbol{C_v}$ は $(N+1-v) \times (N-v)$ 行列である．

3. 自律進化型ボットネットの感染拡散対策

本稿では自律進化型ボットネットの感染拡散対策として，KS モデル及びボランティアモデルの導入を考える．KS モデルは，既知の脆弱性を取り除く，また，感染ホストのウイルスを取り除くことで，自律進化型ボットネットの感染拡散を抑制することを狙うものである．一方，ボランティアモデルは，未知の脆弱性を塞ぐことで自律進化型ボットネットに対抗する．以下にそれぞれのモデルの詳細を説明する．

3.1 キルシグナルモデル

KS モデルは [5] において提案されたモデルで，SIRS モデルの一種である．これは，キルシグナルと呼ばれる警告を導入したモデルである．SIRS モデルと同様に，ウイルスに感染したホストからウイルスが除去された場合，そのホストは除去されたウイルスと同タイプのウイルスに感染することはなく，免疫を保持することとなる．KS モデルではさらに，ウイルスが除去されたホスト (つまり保護状態にあるホスト) が，脆弱性を持つ他のホストに対して警告信号であるキルシグナルを送信する．キルシグナルを受信したホストは，キルシグナルにより脆弱性の存在を知り，その脆弱性を塞ぐことができる．KS モデルではこの動作を表現するために，式 (3) を

$$\theta_{v,w} = \delta_s(N - v - w) + \beta_s w(N - v - w) \tag{5}$$

と置き換える．ただし，β_s は保護状態にある一台のホストから晒し状態へのホストに対するキルシグナルの発生率である．これは晒し状態にあるホストが自ら脆弱性を除去する，もしくは，保護状態にあるホストからのキルシグナルによって，脆弱性を

除去することにより，状態が (v, w) から $(v, w+1)$ へ遷移することを示している．

また，KS モデルでは，キルシグナルによってウイルスに感染しているホストのウイルスを除去し，保護状態に移行させることも想定している．この場合，状態は (v, w) から $(v-1, w+1)$ へ遷移することになり，式 (2) を

$$\mu_{v,w} = \delta_i v + \beta_i v w \tag{6}$$

とすることで表現できる．ただし，β_i は保護状態にある一台のホストから感染状態にあるホストに対するキルシグナルの発生率である．

3.2 ボランティアモデル

自律進化型ボットネットはゾンビコンピュータの計算資源を利用して未知の脆弱性を発見し，それによりホストへ攻撃を行う．そのような攻撃に対して，保護状態にあるホストが自身のみで未知の脆弱性を発見することは難しい．そこでボランティアモデルでは，ウイルスに感染していないホストの計算資源を利用することで，未知の脆弱性を自律進化型ボットネットよりも先に発見し，その脆弱性を塞ぐことを狙う．

ボランティアモデルでは，感染していないホストがグリッドコンピューティングのようなボランティアグループを形成することを想定する．ネットワーク管理者等は，このボランティアグループの計算資源を利用することが可能であり，それにより，自律進化型ボットネットと同様に分散機械学習を行い未知の脆弱性を発見する．ネットワーク管理者は発見した脆弱性を各ホストに知らせることでその脆弱性を塞ぎ，自律進化型ボットネットの拡散防止を狙う．

本稿では簡単化のため，晒し状態，保護状態に関わらずネットワークに存在する全ての未感染ホストがボランティアグループに属し，また，発見された脆弱性は全ての未感染ホストが共有するものとする．このボランティアモデルの導入により，式 (4) が

$$\gamma_{v,w} = \frac{\eta(v+1)}{\sigma(N-v)} \tag{7}$$

と表され，状態 (v, w) から $(v, 0)$ への遷移が妨げられることとなる．ただし，σ はボランティアグループに属するホスト 1 台当たりの計算資源による新たな脆弱性発見率である．

4. 評　　価

4.1 想 定 環 境

本稿では，シミュレーション実験により，標準の自律進化型ボットネット感染モデル，KS モデル，ボランティアモデル，および KS モデルとボランティアモデルの複合モデルの 4 種類のモデルの感染拡散の違いについて考察する．ここでは，全ホスト数 N を 1000 台とする．そのうち一台のホストが感染しており，それ以外の全てのホストに脆弱性がある場合に，ウイルスがどのように拡散，消滅するかを調べる．つまり，システムの初期状態を $(V(0), W(0)) = (1, 0)$ とする．ここで，初期状態 $(V(0), W(0)) = (v_0, w_0)$ から状態 $(0, j)$ $(j = 0, 1, 2, \ldots, N)$ (つまり，ウイルスが全て消滅した状態) への初到達時間 T_{v_0,w_0}

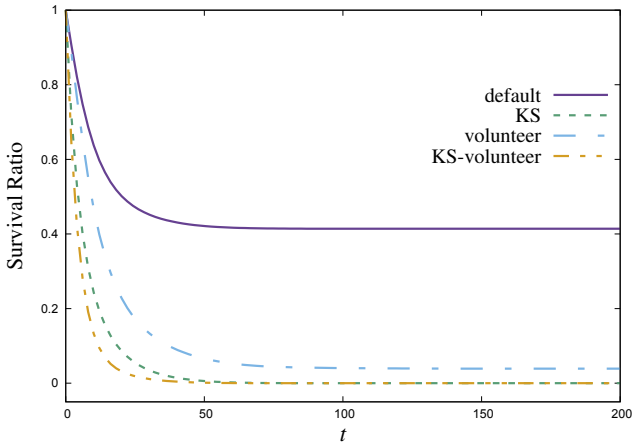

図 3　ウイルス生存率 ($\delta_s = 1.0$，$\sigma = 0.005$)

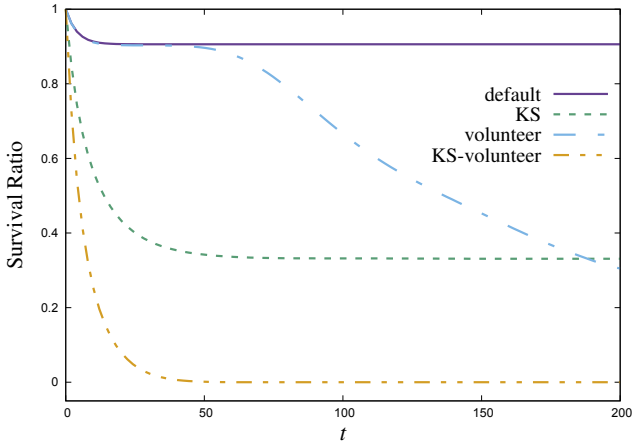

図 4　ウイルス生存率 ($\delta_s = 0.01$，$\sigma = 0.3$)

を，

$$T_{v_0,w_0} = \inf\{t \geqq 0; V(t) = 0 \mid V(0) = v_0, W(0) = w_0\}$$

と定義する．式 (1) より，$V(t) = 0$ の状態はマルコフ連鎖における吸収状態であるため，初到達時間 T_{v_0,w_0} の分布関数は，

$$\Pr(T_{v_0,w_0} \leqq t) = \Pr(V(t) = 0 \mid V(0) = v_0, W(0) = w_0) \tag{8}$$

で与えられる．

4.2 ウイルス感染挙動

以下では，式 (1)，(2) 及び (4) における各種パラメータをそれぞれ，$\alpha = 0.001$，$\delta_i = 0.1$ 及び $\eta = 0.05$ とする．また，式 (3) 及び (5) において，$\delta_s = \beta_s$ とする．さらに，式 (6) において $\beta_i = 0$ として，キルシグナルによるウイルス除去は考慮しないこととする．

図 3 に，$\delta_s = 1.0$，$\sigma = 0.005$ の場合の，時間経過 t に対するウイルス生存率の変化を示す．同様に，図 4 に，$\delta_s = 0.01$，$\sigma = 0.3$ の場合の，時間経過 t に対するウイルス生存率の変化を示す．ここでウイルス生存率とは，総サンプル数に対する，時刻 t においてまだ感染ホストが 1 台以上存在しているサンプル数の割合を意味する．なお，本稿では総サンプル数を 1000 とする．上述のとおり，$V(t) = 0$ の状態 (つまり感染ホスト数

が 0 の状態) はマルコフ連鎖における吸収状態であるため，ウイルス生存率は経過時間 t に対して単調減少であり，時間の経過に伴って感染ホストが存在する可能性が低くなる．またこれは，式 (8) で表される $\Pr(T_{1,0} \leqq t)$ の補分布 $\Pr(T_{1,0} > t)$ に対応する．図において，“default” は自律進化型ボットネットのみの感染モデル，“KS” は KS モデル，“volunteer” はボランティアモデル，“KS-volunteer” は KS モデルとボランティアモデルの複合モデルの結果を表す．

図 3 より，自律進化型ボットネットのみのモデルでは，ウイルス生存率が 0.5 あたりまで下がった後，その値がほぼ横ばいになることがわかる．これは，一度自律進化型ボットネットの感染が拡散してしまうと，ウイルスの完全な除去がほぼ不可能であることを意味する．また，図 3 と 4 を比較すると，図 4 のほうが，自律進化型ボットネットのみのモデルのウイルス生存率が高いことがわかる．これは，式 (3) におけるパラメータである保護率 δ_s が影響しており，この値が高いほど感染を抑制しやすい．

また，図 3 より，保護率 δ_s が高い場合においては，感染抑制を用いたモデルではどの手法においても，多くの場合は早期にウイルスが死滅することがわかる．しかし，ボランティアモデルでは一割程度のサンプルにおいて，ウイルスが蔓延している．これは，式 (7) におけるパラメータ σ を，自律進化型ボットネットの脆弱性発見率 η に対して比較的低い値に設定しているためであると考えられる．また，KS モデル及び混合モデルではほぼすべてのサンプルにおいて早期にウイルスを死滅させており，複合モデルのほうがその影響力は大きい．一方，図 4 に示すように，保護率 δ_s が低い場合は，KS モデルは四割程度のサンプルにおいてウイルスが蔓延していることがわかる．ボランティアモデルを用いた場合，しばらく高い値でウイルス生存率が推移するが，一定時間経過後に生存率が低下していく．これは σ の値が大きい場合，例えウイルスが蔓延しても，徐々にその影響を薄めることが可能であることを示している．また，両者を同時に用いることで，保護率が小さい場合でもウイルスを効果的に死滅させることができることがわかる．

図 5 及び 6 にそれぞれ，$\delta_s = 1.0$，$\sigma = 0.005$ 及び $\delta_s = 0.01$，$\sigma = 0.3$ の場合の時刻 t における平均感染ホスト数 $E[V(t)]$ を示す．図 5 に示すように，自律進化型ボットネットのみのモデル及びボランティアモデルでは早期に感染ホスト数が増加している．しかし，ボランティアモデルでは比較的感染ホスト数が抑えられており，その後の感染ホスト数の増加も見られない．一方，KS モデルおよび複合モデルでは短時間で平均感染ホスト数が 1 を下回っており，感染抑制が効果的に行われていることがわかる．また，図 6 においては，保護率 δ_s が小さいため，KS モデルにおいて感染抑制が十分に行われておらず，平均感染ホスト数が大幅に増加している．一方，ボランティアモデルでは早期に感染ホスト数が急激に増加しているが，その後減少している．そのため，ボランティアモデルは保護率が低い環境において有効であるといえる．またこの場合においても，複合モデルは有効に動作している．

図 7 及び 8 にそれぞれ，$\delta_s = 1.0$，$\sigma = 0.005$ 及び $\delta_s = 0.01$，$\sigma = 0.3$ の場合の，時刻 t において感染ホストが存在するサンプルに対する平均感染ホスト数 $E[V(t) \mid T_{1,0} > t]$ を示す．これらの図において KS モデルや複合モデルの結果のグラフが途中で途切れているものは，全てのサンプルにおいてウイルスが死滅したことを表す．図 7 より，自律進化型ボットネットのみのモデル及びボランティアモデルでは，ほぼ全てのホストが感染していることがわかる．なお，図 3 に示すように，これらのモデルのウイルス生存率は一定時間経過後横ばいになっている．つまりこれらの結果は，ウイルスが運よく死滅することもあるが，一度ウイルスが拡散してしまうと，ほぼ全てのホストが感染してしまうことを意味する．また，図 8 においては，保護率 δ_s が小さいため，KS モデルにおいても急激に感染ホスト数が増加している．一方ボランティアモデルは，感染数が急激に増加した後も，ウイルスが全ホストに感染することを防いでおり，ボットネットの感染を早期に抑制できなかった場合にも有効であるといえる．複合モデルを用いた場合では，図 7 及び 8 のいずれの状況においてもウイルスが早期に死滅しており，二つの感染抑制を組み合わせることは有効である．

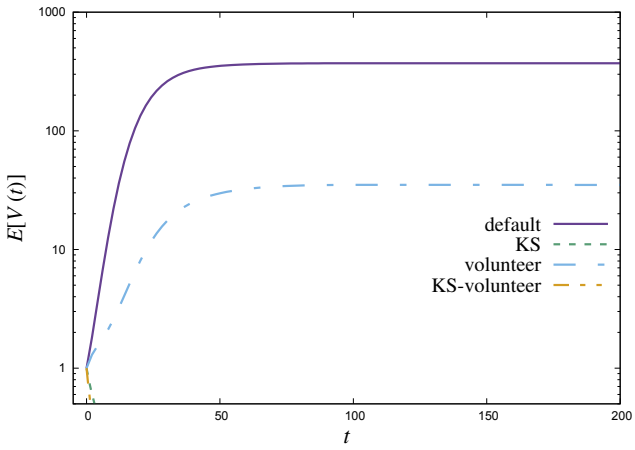

図 5　平均感染ホスト数 $E[V(t)]$ ($\delta_s = 1.0$，$\sigma = 0.005$)

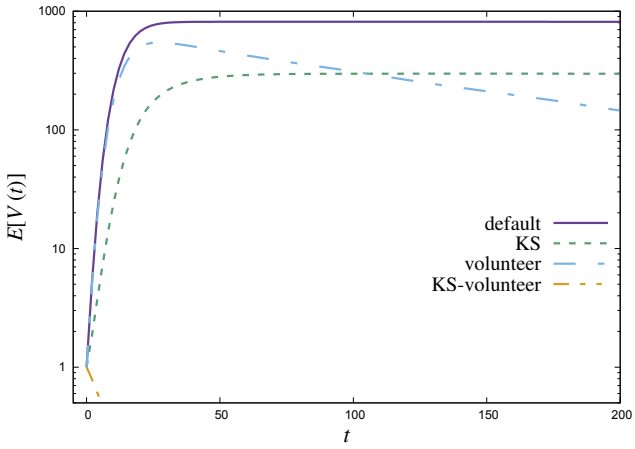

図 6　平均感染ホスト数 $E[V(t)]$ ($\delta_s = 0.01$，$\sigma = 0.3$)

5. ま と め

本稿では，この自律進化型ボットネットの感染拡散対策として，ウイルスに対して免疫を持つホストが他のホストの保護を

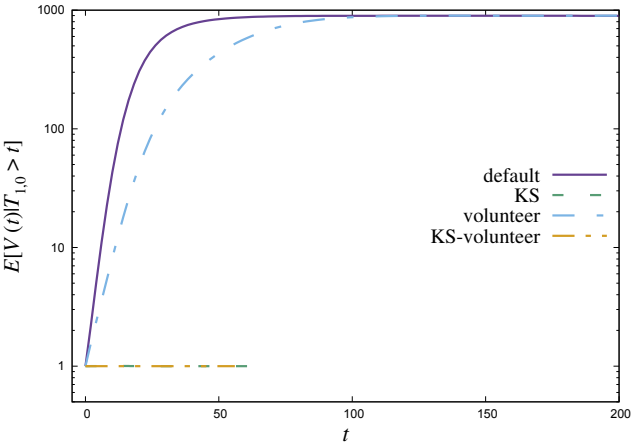

図 7　平均感染ホスト数 $E[V(t) \mid T_{1,0} > t]$ ($\delta_s = 1.0$，$\sigma = 0.005$)

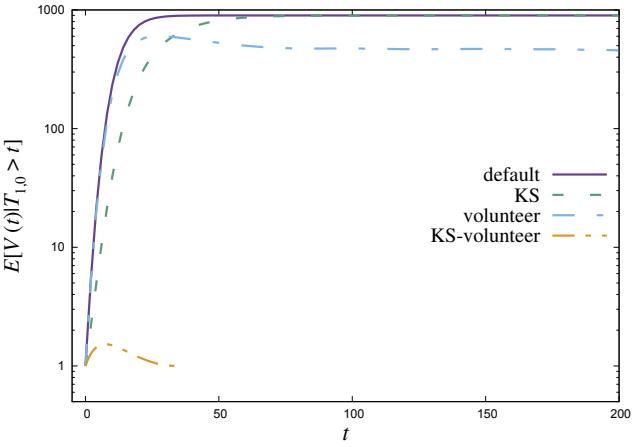

図 8　平均感染ホスト数 $E[V(t) \mid T_{1,0} > t]$ ($\delta_s = 0.01$，$\sigma = 0.3$)

行う KS モデル，ボランティアのホスト群の計算資源を用いた分散コンピューティングにより未知の脆弱性を自律進化型ボットネットよりも先に発見し，それを塞ぐボランティアモデル，及びそれらを複合したモデルの検討を行なった．本稿では，これらのモデルの特性をシミュレーション実験を通じて明らかにした．

文　献

[1] J. Borello and L. Me, “Code obfuscation techniques for metamorphic viruses,” *Journal in Computer Virology*, vol. 4, no. 3, pp. 211–220, 2008.

[2] A. Cani, M. Gaudesi, E. Sanchez, G. Squillero, and A. Tonda, “Towards automated malware creation: code generation and code integration,” in *Proc. Symposium on Applied Computing*, Gyeongju, Korea, Mar. 2014.

[3] J. Dean et al., “Large scale distributed deep networks,” in *Proc. Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2012.

[4] G. E. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] J. O. Kephart and S. R. White, “Measuring and modeling computer virus prevalence,” in *Proc. the 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, May. 1993, pp. 2–15.

[6] T. Kudo, T. Kimura, Y. Inoue, H. Aman, and K. Hirata, “Behavior analysis of self-evolving botnets,” in *Proc. the 2016 International Conference on Computer, Information, and Telecommunication Systems (CITS 2016)*, Kunming, China, Jul. 2016.

[7] E. Meeds, R. Hendriks, S. Faraby, M. Bruntink, and M. Welling, “MLitB: machine learning in the browser,” arXiv:1412.2432.

[8] S. Noreen, S. Murtaza, M. Z. Shafiq, and M. Farooq, “Evolvable Malware,” in *Proc. Genetic and Evolutionary Computation Conference*, Montreal, Canada, Jul. 2009.

[9] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A multifaceted approach to understanding the botnet phenomenon,” in *Proc. ACM SIGCOMM conference on Internet measurement*, Rio de janeiro, Brazil, Oct. 2006.

[10] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, “Predicting vulnerable software components via text mining,” *IEEE Transactions on Software Engineering*, vol. 40, no. 10, pp. 993–1006, 2014.

[11] M. Wang, H. Zhou, M. Guo, and Z. Zhang, “A scalable and topology configurable protocol for distributed parameter synchronization,” in *Proc. Asia-Pacific Workshop on Systems*, Beijing, China, Jun. 2014.

[12] F. Yamaguchi, F. Lindner, and K. Rieck, “Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning,” in *Proc. USENIX conference on Offensive Technologies*, San Francisco, CA, Aug. 2011.