情報処理学会論文誌 Vol.57 No.3 1022–1039 (Mar. 2016)

不審活動の端末間伝搬に着目した標的型攻撃検知方式

川口 信隆[1,a)] 築地原 護[2] 井手口 恒太[1] 谷川 嘉伸[1] 冨村 英勤[1]

受付日 2015年6月22日, 採録日 2015年12月7日

概要：標的型攻撃は，情報窃取や資産破壊を目的に企業や国家機関などの特定組織のネットワークを執拗に狙う攻撃の総称である．近年の標的型攻撃の高度化にともない個々の端末やプロセスを分析する手法では攻撃の検知が難しくなりつつある．そこで，我々は攻撃に用いられる個々の要素を深く分析するのではなく，多くの標的型攻撃で発生する活動である拡散活動に着目する．拡散活動は，攻撃者が最終目的となる資産にたどり着くために複数の端末を渡り歩く活動である．我々は，複数端末で行われる様々な種類の不審活動を分析し，攻撃者の拡散経路をグラフ構造として抽出することで拡散活動を検知する方式を考案した．そして，ある組織の同一部署に属する 30 台の端末の 2 カ月間にわたる活動ログを用いて方式の評価実験を行った．その結果，提案方式は標的型攻撃を模擬した攻撃シナリオに対して検知率 97%を達成するとともに，既存方式と比べて誤検知頻度を 10 分の 1 まで削減できることが明らかになった．加えて，提案方式はプロセスやファイルシステムにいっさい痕跡を残さない高度な攻撃に対しても 70%以上の検知率を実現することを確認した．

キーワード：標的型攻撃，侵入検知，マルウェア

Detection of Advanced Persistent Threat Based on Cascade of Suspicious Activities over Multiple Internal Hosts

Nobutaka Kawaguchi[1,a)] Mamoru Tsuichihara[2] Kota Ideguchi[1] Yoshinobu Tanigawa[1] Hideyuki Tomimura[1]

Received: June 22, 2015, Accepted: December 7, 2015

Abstract: As Advanced Persistent Threats, which persistently target specific organization networks with the aim of stealing their information, destroying their assets, or disrupting their operations, have been more prevalent and sophisticated than ever before, it becomes further difficult to detect the attacks by solely analyzing each process or host. Instead of deeply analyzing each element that may be a part of an attack, we focus on an activity called lateral movement in which the attackers move from one host to another, looking for assets they want to access and manipulate. We designed and developed a scheme which detects the lateral movement by analyzing various types of suspicious activities over multiple hosts and extracting how an attacker moves in the network as a graph structure. Through evaluation experiments with activity logs from 30 hosts in the same organization's department network for two months and three types of attack scenarios, we found that the proposed scheme detects simulated targeted attacks at a rate of higher than 97%, while suppressing the false positives to about 10% of an existing work. In addition, we confirmed that the proposed scheme can detect sophisticated attacks, which do not leave any taints in processes and file systems and can evade the exiting work, at a rate of higher than 70%.

Keywords: advanced persistent threat, intrusion detection, malware

[1] 株式会社日立製作所
Hitachi Ltd., Yokohama, Kanagawa 244–0717, Japan
[2] 株式会社日立アドバンストシステムズ
Hitachi Advanced Systems Corporation, Yokohama, Kanagawa 244–0717, Japan
[a)] nobutaka.kawaguchi.ue@hitachi.com

Windows はマイクロソフト社の米国またはその他の国の登録商標である．Core i5/7 は Intel 社の米国またはその他の国の登録商標である．本稿中の製品および会社名は，各々の会社の登録商標である．

1. はじめに

標的型攻撃は，価値が高い資産を持っている特定の組織を執拗に狙う攻撃の総称である [1], [2], [3]．多くの重要インフラや企業がこの攻撃の被害を受けている [4]（たとえば，Stuxnet はイランにある核関連施設をダウンさせようとした [5]）．コンピュータワームのような従来の攻撃とは異なり [6], [7]，標的型攻撃の目的は，技術力の誇示や愉快犯ではなく，金銭の窃取や政治的動機，あるいはサイバー戦争・テロにある．

攻撃者の技術力が向上し，様々な攻撃回避技術（e.g. 環境依存型マルウェア，フットプリントの小さいツールや正規ツールの利用）[8], [9], [10], [11], [12], [13], [14] を用いるようになるに従い，攻撃を構成する要素（個々の被攻撃端末・プロセス，マルウェア，C&C 通信，エクスプロイト）を個々に分析するのみでは，検知が難しくなりつつある．

上記の問題を解決するために，我々は，標的型攻撃のなかで頻繁に行われる「拡散活動」に着目した検知方式を提案する．拡散活動では，攻撃者は，遠隔操作ツール（e.g. Psexec [11]）やシステムの脆弱性を悪用して，ある端末から別の端末に渡り歩く [2], [15]．必ずしもすべての標的型攻撃が拡散活動をともなうわけではないが，標的型攻撃と見られる過去のインシデントの多くで，拡散活動が行われていることが報告されている [12]．

本方式は攻撃に含まれる個々の要素に着目するのではなく，複数端末の様々な活動ログに対して時空間分析を行い拡散活動の検知を実現する．本方式は，攻撃者が高度なスキルを持っている場合であっても，攻撃にともない通常オペレーション時にはあまり観測されない「やや不審な活動」が被攻撃端末で発生することを仮定する．これは，端末の通常オペレーションと攻撃とでは，その目的が異なるためである．本方式は，攻撃に起因する可能性がある活動の発生頻度を基に端末の不審性を分析する．端末の不審性は，アノマリベース/ミスユーズベース，両方のアプローチで評価される．そして，拡散活動にともない不審性が高い端末が連鎖的に現れる現象を，被攻撃端末をノードとするグラフ構造として抽出する．そして，グラフがある基準を満たすとき，本方式は，標的型攻撃が発生していると判断してアラートをあげる．

本方式の特徴は，複数の端末で発生する，不審性がわずかでもある様々な種類の活動を収集・相関分析し，拡散活動を表すグラフ構造として抽出する点にある．これにより，個々の端末で行う活動の不審性が低い標的型攻撃の検知が可能になる．また，既存技術と比べて，誤検知の発生頻度を低く抑えることができる．

我々は，3 種類の攻撃シナリオと，ある組織の同一部署に属する 30 台の端末の 2 カ月間の活動ログを用いた評価実験を通じ，検知精度および，プロセスやファイルシステムに痕跡を残さない高度な攻撃に対する有効性の両面で，本方式は既存方式よりも優れていることを確認した．我々が知る限り，本方式は，拡散活動にともなう不審端末発生の連鎖に着目した初めてのアプローチである．

本稿は以下のとおり構成される．2 章では関連研究と本方式の位置付けについて説明する．3 章・4 章では本方式の詳細および実装について述べる．5 章では性能評価および実験の詳細について述べる．6 章では本方式の優位性および今後の研究課題について議論する．7 章を本稿のまとめとする．

2. 関連研究

既存研究の多くは，標的型攻撃を構成する単一要素（個々の被攻撃端末・プロセス，マルウェア，C&C 通信，エクスプロイト）を基に攻撃検知を行っている．

（1）マルウェアの挙動

動的解析・静的解析を基にマルウェアを検知・分類する技術に関しては，これまでに多数の研究発表がなされている [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]．これらの研究では，ルールや振舞いモデルに従って，マルウェアの検知あるいはファミリの分類を行う．また文献 [26] では，エンタープライズネットワークに対する低速アドレススキャンを検知する方法が提案されている．前述のとおり，攻撃の高度化にともない，正規ツールやフットプリントが小さいマルウェアが利用され，発見しやすい既知脆弱性が用いられなくなるようにつれ，これらの方式で攻撃を効果的に検知するのは難しくなりつつある [8], [9], [10], [11], [12], [13], [14]．たとえば文献 [22] では，プロセス・ファイル・ソケット・レジストリ間のデータフローをグラフ化してマルウェアによく見られる挙動を検知するが，挙動の定義から外れる動作を行うものは検知できない．また，正規プロセスの 2%弱を誤検知するため，実運用では大量の誤アラートが発生する可能性がある．また，文献 [27] では，ボットのネットワーク活動の挙動モデルを構築し，モデルへの適応度が高い端末をボット感染端末として検知する．提案されている挙動モデルは，SPAM や DDoS 攻撃などをともなうボットに特化しているため，標的型攻撃検知に応用することは難しい．

（2）アクセス先ドメイン

マルウェアが C&C 通信などでアクセスするサーバが存在するドメインの特徴量は，標的型攻撃の検知に利用できる可能性がある [28], [29], [30], [31], [32]．たとえば，マルウェアの中にはドメイン生成アルゴリズム（DGA）を用いてアクセス先ドメイン名を定期的に生成するものがある．これらのドメイン名は通常のドメインとは異なる場合が多い．また，これらのドメインは生存期間が比較的短い．このため，機械学習アルゴリズムなどを活用して個々の URL やドメインの不審性を評価し攻撃を検知すること

が可能である．しかしこれらのアプローチでは分類モデル作成のために大規模な監視・分析が必要である [29]．また，DropBox などの有名ドメイン上にサーバが設置された場合の検知が難しい [10]．

（3）プロセスホワイトリスト

プロセスホワイトリスト機構は，あるホストで実行可能なプログラムの種類を制限する [33]．このアプローチはマルウェアが含まれる実行ファイルの起動を防ぐことができるが，ユーザやシステムが頻繁に新規アプリケーションのインストールを行うような環境では，多数の誤検知が発生するという問題がある．また，このアプローチではプロセスやファイルシステムに痕跡を残さない攻撃に対応できない．

（4）マスカレード検知

マスカレード検知システムは，ユーザのコマンド履歴やキーストローク・マウス操作履歴を分析することで，正規ユーザになりすました攻撃者（マスカレーダ）を発見する [34], [35]．標的型攻撃の検知には，ユーザレベルの活動に加え，システムやアプリケーションの活動も考慮する必要がある．たとえば，攻撃ツールのダウンロードは，ユーザが手動で行う場合もあればアプリケーションやシステムが自動的に行う場合もある．加えて，この方式では各被攻撃端末で行われた活動量が小さい場合の検知が難しいという課題がある．

（5）エンタープライズネットワークの分析

文献 [36], [37], [38] では，エンタープライズネットワーク中で行われた活動を分析し，どのように攻撃検知に活用可能かを議論している．これらの研究の主な注目対象はネットワーク活動であり，各端末が行う端末内活動（プロセス起動やファイル作成）については論じられていない．また，文献 [39] では，標的型攻撃を模擬するためのエンタープライズネットワークモデルを考案している．このモデルは，標的型攻撃対策を検討するうえで有用であるが，拡散活動のステップは含まれていない．

（6）複数の被害端末の関連付け

近年，エンタープライズネットワーク内の複数端末のログを収集し，ネットワークワームやボットネットを検知する研究が行われている．文献 [40], [41], [42], [43], [44], [45] では，ワームの拡散活動をツリー，チェインまたはグラフ構造として表現して検知を行う．これらのアプローチは多数の端末に感染するワーム検知には有効であるが，被攻撃端末数が比較的少数と考えらえる標的型攻撃を検知するのは難しい．また，文献 [6], [7], [46] は他端末からのインバウンド通信を受けた端末が発信元端末に類似した不審挙動を示すという特徴を基にワームの検知を行う．また，文献 [47] では複数端末の活動の類似度を基にボットネットを検知する．これらの研究では，複数の端末での挙動が異なる攻撃を検知することが難しい．また，文献 [48] では複数端末が示す異常活動の時系列における関連性を基に攻撃検知を行う．このアプローチでは，複数端末に対して同時に攻撃が行われるケースを想定しており，攻撃の進行にともない複数端末が次々と攻略されていくような標的型攻撃は対象外となる．文献 [49] では，C&C サーバに対するトラヒックと拡散活動に起因する通信を関連付けることで標的型攻撃を検知する方式が提案されている．このアプローチは，拡散活動・C&C 通信に用いられる特定のプロトコル（Microsoft SMB）の特徴に強く依存しているため，既知の攻撃に対しては有効であるものの，異なるタイプの通信が用いられる攻撃を検知するのは難しい．また，踏み台攻撃を検知する方式 [50], [51] では，C&C サーバとの通信フローと拡散活動に起因する通信フローを関連付けることは難しい．これは，攻撃では個々のフローに異なる通信プロトコルを用いることが可能であり，また，チャフの挿入などにより分析を妨げることが比較的容易であるためである．

上述のとおり，攻撃を構成する個々の要素（被攻撃端末・プロセス，C&C 通信など）に基づく検知方式は，意図的あるいは偶発的に当該要素を含まない攻撃には有効ではない．また，複数の要素を関連付けて検知を行う方式も標的型攻撃の本質的な特徴の 1 つである拡散活動を十分にはとらえていない．我々のアプローチは，端末活動の抽象化モデルを基に複数端末をまたがる拡散活動を検知することで，既存方式を補完する．具体的には，提案方式は以下の点で既存研究とは異なる．

- 本方式は，複数端末で見られる数種類の不審性を関連付けることで拡散活動を検出する．このため，本方式は，個々の端末での活動の不審性がわずかな場合であっても検知が可能であり，誤検知の発生も抑えることができる．また，監視対象の活動の一部を行わない攻撃も検知することができる．
- 端末の不審性は異常検知の観点と，ネットワーク活動・端末内活動に関する抽象度が高いモデルから求められる．このため，本方式は特定のアプリケーションプロトコルや振舞いに依存せず，未知の脆弱性を狙うような攻撃を検知できる．我々は，活動の主体を区別しない．我々の監視対象には，ユーザプロセス・アプリケーションプロセス・システムプロセスがすべて含まれる．これは，攻撃者は上述の主体のうちいずれかまたはすべてを用いて攻撃を行うためである．
- 提案方式の検知性能を評価するために，我々は拡散活動を組み入れることで，既存のものよりも現実性が高い 3 種類の標的型攻撃のシナリオを設計した．

3. 標的型攻撃検知方式

3.1 コンセプト

提案する標的型攻撃検知方式の目的は，標的型攻撃により組織ネットワークに侵入したマルウェア/攻撃者による

拡散活動を複数のネットワーク内端末の活動の不審性を基に早期検知することで攻撃被害を最小化することにある．本方式のアプローチは，標的型攻撃に関する以下の事実・仮説に基づいている．

- 事実：標的型攻撃では，攻撃者にとっては，最終目的を達成する前の検知を回避することが重要である．このため，攻撃者は，Windows 標準コマンドの活用，活動ごとに異なる攻撃ツールの使い分けなどを通じて，アンチウイルスソフトなどによる検知を避ける傾向にある．このため，個々の端末やプロセスのみを分析して検知を行うのは困難になりつつある [8], [9], [10], [11], [12], [13], [14].
- 仮説 1：その一方で，端末の通常オペレーションと攻撃活動とでは目的が異なるため，攻撃にともない，「やや不審な活動（以下，不審活動）」が端末内で発生すると考えられる．例としては，通常オペレーション（攻撃が発生していない状態での端末動作）では使用頻度が低い Windows コマンドの実行，これまでに起動実績のないプロセスの起動，通常アクセスしない端末との通信などがある．
- 仮説 2：これらの不審活動は，通常オペレーションでも発生しうる活動であるため，この点のみに着目して検知を行おうとすると，誤検知が大量に発生する恐れがある．しかし，攻撃者が拡散活動により複数の端末を渡り歩く場合，不審活動を行う端末（以下，不審端末）が連鎖的に発生することになる．この現象は，通常オペレーション時には発生しにくく，高精度な検知に活用可能である．

本方式は，個々の端末の活動を個別に分析するのではなく，不審活動・不審端末の発生タイミングの相関性を基に，複数の端末をまたがった拡散活動の痕跡を「不審活動グラフ」として検出する．これにより，個々の活動・端末を監視するだけでは発見が困難な，不審性が低い高度な標的型攻撃の検知が可能になる．

3.2 ユースケース

本方式のユースケースを**図 1** に示す．本方式は，ネットワーク内に設置された拡散活動検知装置と Web プロクシ，各端末にインストールされている端末監視機能（TM）から構成される．

ユースケースではまず，インターネットから組織ネットワークへの侵入に成功した攻撃者が，端末 A → 端末 B → 端末 C → 端末 D の順で不正アクセスを行い，活動を広げていく．その際には，サーバからの機密情報の窃取，サービスに対する脆弱性攻撃，Psexec などの遠隔実行ツールを用いた遠隔端末操作などが行われる．

これに対して本方式では，まず，各端末内にインストールされた端末監視機能が，端末の活動（プロセス起動やファイルアクセス，同一ネットワーク内の他端末との通信など）に関するログを，逐一拡散活動検知装置に送信する．同様に，ネットワークに設置された Web プロクシは，端末とインターネット上の Web サーバ間の通信ログを，逐一拡散活動検知装置に送信する．

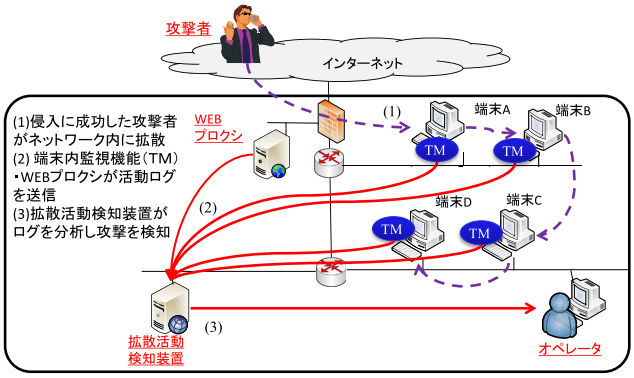


図 1 本方式のユースケース
Fig. 1 Use case of proposed scheme.

一方，拡散活動検知装置は，端末監視機能・Web プロクシから受信したログをリアルタイムに分析して不審活動を特定する．次に，不審活動の不審度がある基準を満たす端末を不審端末として特定する．さらに，複数の不審端末から構成される不審活動グラフを検知する．不審活動グラフがある基準を満たす場合，ネットワーク内で拡散活動が発生していると判断し，アラートをオペレータなどに送信する．

なお，端末の活動（端末活動）のうち，どの活動が不審活動であり，どの活動が正常活動であるかの判定は，拡散活動検知装置が行う．このため，端末監視機能は端末内で発生した活動の情報を，不審・正常の判断を行わずにそのまま拡散活動検知装置に送信する．同様に，Web プロクシは，すべての Web 通信の情報を端末監視機能に送信する．

3.3 検知手順概要

図 2 に本方式の検知手順の概要を，**図 3** に拡散活動検知装置のアーキテクチャをそれぞれ示す．拡散活動検知装置は，各端末の不審性を「不審活動度」として評価する．不審活動度は，時間経過にともない，不審活動の内容・頻度に応じて増減する．各端末は不審活動度に応じた「状態」をとる．不審活動度が閾値未満である端末は「正常状態」にあり，閾値以上である端末は「不審状態」にあるとする．本方式では，攻撃者が侵入先端末で悪意ある活動を行うに従い，端末の不審活動度は徐々に上昇し，ある時刻に正常状態から不審状態に転換する．同様に，攻撃者の活動が完了すると不審活動度が閾値を下回り，端末の状態は正常状態に再転換する．

拡散活動検知装置は，検知された不審端末を基に不審活動グラフを構築する．不審活動グラフは，不審端末をノード，不審端末間で発生する TCP/IP 通信（内部通信）を有

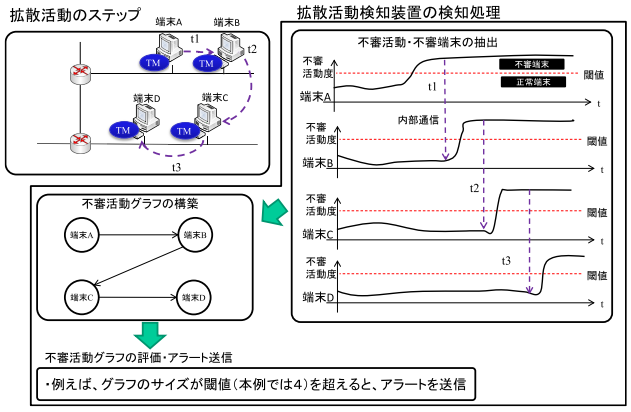


図 2 検知手順概要
Fig. 2 Overview of detection step.

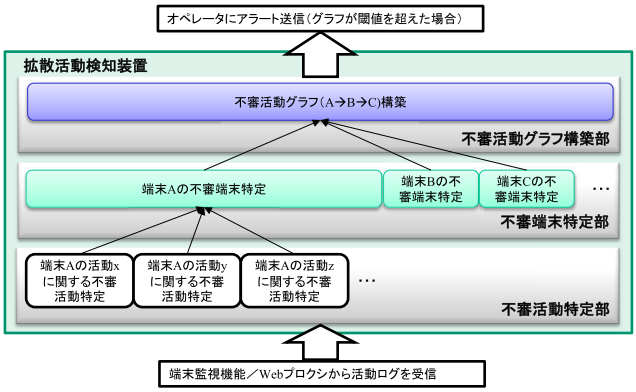


図 3 拡散活動検知装置アーキテクチャ
Fig. 3 Architecture of lateral movement detector.

向辺とする有向グラフである．本方式では，拡散活動にともない，正常状態であった端末が不審端末に次々に転換していく現象をとらえる．このため，概略を示すと，2 つの端末（端末 A および端末 B）は，

- 不審状態にある端末 A から，正常状態である端末 B に対して内部通信が発生する，
- 内部通信発生後一定内に，端末 B の状態が不審状態に転換する，

場合に限り，A→B というグラフ構造をとる．

最後に，不審活動グラフの形状（たとえばサイズ）を評価し，評価値が閾値を超えた場合にアラートを送信する．

以降では，本方式が想定する脅威，不審活動および不審端末特定方法，不審活動グラフ構築方法の詳細について述べる．

3.4 想定脅威

本方式では以下に示す標的型攻撃を想定脅威とする．

1. 攻撃者は組織ネットワークからの機密情報窃取を最終目的とし，マルウェアや脆弱性攻撃など何らかの方法によりネットワーク内端末の制御を奪い，C&C サーバに接続させる．
2. 攻撃者は C&C サーバを通じて攻撃に用いるツールを端末にダウンロードする．また，端末を操作し，ipconfig や tasklist といった Windows 標準コマンドなどを用い端末のプロパティ情報（プロセスリストや IP アドレスなど）を取得する．さらに，アドレススキャンなど何らかの手段で拡散先端末を決定する．
3. 攻撃者は，Psexec などの遠隔管理ツールまたは脆弱性攻撃（MS08-067 の悪用など）などにより，拡散先端末に内部通信を介して侵入して制御を奪う．なお，遠隔管理ツールを用いてログインする場合，攻撃者は何らかの手段により拡散先端末の ID/パスワードを入手済みであることを前提とする．
4. 攻撃者は，所望の機密情報が保存されている端末にたどり着くまで，2〜3 の動作を繰り返す．
5. 攻撃者は，所望の機密情報が保存されている端末を発見すると機密情報を C&C サーバにアップロードする．
6. 攻撃者は，最後に，拡散した端末から痕跡を削除して攻撃を完了する．

本検知方式では，拡散活動をともなわない標的型攻撃は検知の対象外とする．また，拡散活動に際して拡散元から拡散先への内部通信をともなわない攻撃も対象外とする．80%以上の標的型攻撃において，拡散活動と思われる活動が行われているという報告もあり [12]，本検知方式により多くの標的型攻撃に対応可能と考える．

3.5 不審活動特定

標的型攻撃にともない発生する場合があると考えられる不審活動を，以下の 2 種類に分類する．

- アノマリ型：攻撃が発生していない一定の通常オペレーション期間で観測されなかったあるいは観測頻度が低い活動（例：普段実行しないプロセスの起動）
- ミスユーズ型：通常オペレーション中でも発生しうるが，経験的に，攻撃中に発生する頻度が高いことが分かっている活動（例：実行ファイルの作成）

ここで，アノマリ型不審活動の特定には通常状態を定義した通常プロファイルが必要である．このため，本方式は，通常プロファイル作成のための事前学習を行う学習フェイズ，学習結果を用いて検知を行う検知フェイズという 2 つのフェイズを持つ．このため，本方式は機械学習型の検知手法の一種といえる．

我々は，前述の想定脅威の各攻撃ステップにおいて，アノマリ型/ミスユーズ型のいずれか，または両方の攻撃が実施されうると想定する．

どういった活動を不審活動としてとらえるかは，標的型攻撃やマルウェアに関する知見（文献 [12], [14], [52] など）を基に決める必要がある．今回我々は，アノマリ型不審活動として 4 種類，ミスユーズ型不審活動として 2 種類の合計 6 種類の不審活動を定義し監視対象とした．以下に詳細を示す．なお，関連研究にあげた検知方式を不審活動の一

種として本方式で扱うことも可能である．

3.5.1 アノマリ型不審活動および特定方法

アノマリ型不審活動については，以下の 4 種類の活動を監視対象項目とする．

（1）プロセス起動

端末ごとに，学習フェイズ中に一定回数以上発生したプロセスのリストをプロセスに関する通常プロファイルとして構築する．通常プロファイルの各要素は（端末，起動プロセス名）のタプルである．検知フェイズ中で通常プロファイルに含まれないプロセスが起動すると，不審活動として特定される．

標的型攻撃発生の際は，攻撃ツールや普段使わない標準コマンドの実行にともない学習フェイズ中で観測されなかったあるいは観測頻度が低いプロセスの起動が発生する場合があると考えられる．プロセス起動のイベントは端末監視機能により取得される．

プロセス起動に関する不審活動は（端末，起動プロセス名，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生したプロセス起動に関する不審活動の集合は $S_{Proc}(x, t1, t2)$ と表記する．

（2）ポートオープン

端末ごとに，学習フェイズ中に一定回数以上 TCP/UDP のリスニングポートを開いたプロセスのリストをポートオープンに関する通常プロファイルとして構築する．通常プロファイルの各要素は（端末，ポートオープンしたプロセス名）のタプルである．検知フェイズ中で通常プロファイルに含まれないプロセスがポートを開こうとすると，不審活動として特定される．

標的型攻撃発生の際は，いわゆるバックドア作成にともない学習フェイズ中で観測されなかったあるいは観測頻度が低いプロセスがポートオープンを行う場合があると考えられる．ポートオープンのイベントは端末監視機能により取得される．

ポートオープンに関する不審活動は（端末，ポートオープンしたプロセス名，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生したポートオープンに関する不審活動の集合は $S_{Port}(x, t1, t2)$ と表記する．

（3）内部通信

端末ごとに，学習フェイズ中に一定回数以上 TCP/IP 通信を行った同一組織ネットワーク内の端末のリストを，内部通信に関する通常プロファイルとして構築する．通常プロファイルの各要素は（端末，送信先端末）のタプルである．検知フェイズ中で通常プロファイルに含まれない通信先端末に対して通信が発生すると，不審活動として特定される．

標的型攻撃発生の際は，拡散活動にともない学習フェイズ中で観測されなかったあるいは観測頻度が低い通信先との通信が発生する場合があると考えられる．内部通信のイベントは端末監視機能により取得される．

内部通信に関する不審活動は（端末，送信先端末，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生した内部通信に関する不審活動の集合は $S_{Comm}(x, t1, t2)$ と表記する．

（4）Web 通信

端末ごとに，学習フェイズ中に一定回数以上，CONNECT メソッドで一定サイズ以上のデータ送信をともなう通信を行った Web サーバのドメインを，Web 通信に関する通常プロファイルとして構築する．通常プロファイルの各要素は（端末，通信先ドメイン）のタプルである．検知フェイズ中で，通常プロファイルに含まれないドメインに対して CONNECT メソッドで一定サイズ以上のデータ送信が発生すると，不審活動として特定される．

標的型攻撃発生の際は，C&C サーバへの情報漏えいにともない，学習フェイズ中で観測されなかったあるいは観測頻度が低いドメインに対して一定サイズ以上のデータ送信が発生する場合がありうる．

Web 通信のメソッドを CONNECT に限定するのは，標的型攻撃では，端末と C&C サーバ間で SSL トンネルを張りセッションを維持してコマンドの送受信・情報窃取を行うケースが多いためである．

我々は，組織ネットワーク内の端末が外部ネットワークの Web サーバと通信する際は Web プロクシを経由することを想定する．このため，Web 通信のイベントは Web プロクシにより取得される．

Web 通信に関する不審活動は（端末，通信先ドメイン，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生した Web 通信に関する不審活動の集合は $S_{Web}(x, t1, t2)$ と表記する．

なお，後述の評価実験において，POST/GET/CONNECT などすべてのメソッドを対象にし，データ送信サイズを考慮せず通信先ドメインのみを基に不審活動を特定した場合の検知性能について言及する．

3.5.2 ミスユーズ型不審活動および特定方法

ミスユーズ型不審活動については，以下の 2 種類の活動を監視対象項目とする．

（1）実行ファイル作成

端末のローカルドライブに対する実行ファイルの作成は不審活動として特定される．標的型攻撃発生の際は，攻撃ツールのダウンロードが行われるため，実行ファイル作成が発生する場合があると考えられる．

一定期間内に複数の実行ファイルが作成された場合は，1 つの不審活動としてカウントされる．これは，通常オペレーション時に，複数の実行ファイルから構成される新規アプリケーションがインストールされる際に，多数の不審活動として特定されないようにするためである．

実行ファイル作成のイベントは端末監視機能により取得

される．

実行ファイル作成に関する不審活動は（端末，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生した実行ファイル作成に関する不審活動の集合は $S_{File}(x, t1, t2)$ と表記する．

（2）ICMP Echo Request

端末が ICMP Echo Request を数回送信し，受信した ICMP Echo Reply 数が送信数の $1/N$（$N > 1$）以下になるとき，不審活動としてカウントされる．標的型攻撃発生の際は拡散先端末を探索するアドレススキャンにより Reply をともなわない ICMP Echo Request が発生する場合があると考える．

ICMP Echo Request のイベントは端末監視機能により取得される．

ICMP Echo Request に関する不審活動は（端末，発生時刻）のタプルで識別される．端末 x で時刻 $t1$ から $t2$ に発生した ICMP Echo Request に関する不審活動の集合は $S_{ICMP}(x, t1, t2)$ と表記することとする．

3.6 不審端末特定

拡散活動検知装置は，端末監視機能・Web プロクシから取得した不審活動を基に，各端末の過去 *window* 時間内における不審活動度を求める．

まず，監視対象として $A1, \ldots, Ai, \ldots, An$ という n 種類の不審活動が定められたときに，拡散活動を検知するという最終目的に基づいて，個々の端末の不審活動度を算出する方法について述べる．時刻 $t1$ から $t2$ に端末 x で発生した不審活動 Ai の集合を $Si(x, t1, t2)$ をとするとき，時刻 t における端末 x の不審活動度 $\mathrm{SUS}(x, t)$ を，以下の式 (1) で定義することとする．

$$\mathrm{SUS}(x, t) = \Sigma_{i=1}^{n} W_i * F_i(S_i(x, t - window, t)) \qquad (1)$$

Fi は集合 Si を評価する関数であり，活動 Ai の種類により異なる．Wi は活動 Ai の不審さに基づく重み付けである．

不審活動度は，複数の不審活動に関わる情報を基に算出されるものである．このため，複数の属性に基づいて評価値を算出する基本的な方法である，重み付き線形和を用いて式 (1) を立式した．線形和よりもふさわしい不審活動度の算出方法については，今後の検討課題とする．

本稿で検討した 6 種類の不審活動に基づく不審活動度は以下の式 (2) で求められる．

$$\begin{aligned}\mathrm{SUS(x,t)} = {} & W_{Proc} * |\mathrm{Uniq}(S_{Proc}(x, t - window, t))| \\ & + W_{Port} * |\mathrm{Uniq}(S_{Port}(x, t - window, t))| \\ & + W_{WEB} * |\mathrm{Uniq}(S_{WEB}(x, t - window, t))| \\ & + W_{Comm} * |\mathrm{Uniq}(S_{Comm}(x, t - window, t))| \\ & + W_{File} * |S_{File}(x, t - window, t)| \\ & + W_{ICMP} * \mathrm{Min}(1, |S_{ICMP}(x, t - window, t)|)\end{aligned} \qquad (2)$$

ここで，$|.|$ 関数は集合内の要素数を返す関数である．Uniq 関数は発生時刻以外の属性が異なる要素の集合を返す関数である．このため，たとえばプロセス起動に関する不審活動では，同一プロセスを window 時間内に何度も起動しても不審活動度は上昇しない．Min 関数は引数のうち値が小さいものを返す関数である．このため，ICMP Echo Request に関する不審活動は，「あり」または「なし」の 2 値で評価される．

上式が示すように，不審活動度は，重み付けされた各不審活動の発生頻度の合計値として求められる．このため，攻撃者が 6 種類の不審活動をすべては行わない場合，あるいは監視が困難な場合であっても，不審活動度を算出できる．攻撃者が各端末内でどのような活動を行うかを事前に予想することは難しい．また，攻撃者のスキルによっては不審活動の一部は観測が難しくなる可能性がある．たとえば，攻撃が，起動中の他プロセスを奪取する場合や OS カーネルに手を加える場合は，プロセス起動やポートオープン，実行ファイル作成を正しく監視できない可能性がある．一方で，ネットワークに関する活動である，ICMP Echo Request，内部通信，Web 通信はネットワーク上からの監視が可能であるため監視見逃しが少なくなると考えられる．

このため，ネットワークに関する不審活動度のみを用いて攻撃検知を行いたい場合，もう 1 つの不審活動度 SUS_{Net} を用いる．SUS_{Net} は以下の数式 (3) で求める．

$$\begin{aligned}\mathrm{SUS}_{Net}(x,t) = {} & W_{WEB} * |\mathrm{Uniq}(S_{WEB}(x, t - window, t))| \\ & + W_{Comm} * |\mathrm{Uniq}(S_{Comm}(x, t - window, t))| \\ & + W_{ICMP} * \mathrm{Min}(1, |S_{ICMP}(x, t - window, t)|)\end{aligned} \qquad (3)$$

最後に，不審活動度が閾値 TH_{SUS} に達した場合，当該端末は不審端末と判断される．ある端末が不審端末であるかどうかはリアルタイムに判断される．このため，ある時刻に不審端末に転換した端末において，その後の不審活動の発生がないと不審活動度は減少し，閾値を下回ると正常端末に再転換する．

3.7 不審活動グラフ構築

拡散活動検知装置は不審端末をノード，不審端末間の内部通信を有向エッジとする不審活動グラフを構築する．本節では，グラフ構築アルゴリズムを示す．

まず，グラフ構築アルゴリズムに用いられる記号を**表 1** のとおり定義する．拡散活動検知装置は，監視対象のネットワークで観測される複数の不審活動グラフを管理する．個々の不審活動グラフには，ノードおよびエッジの集合，

表 1 グラフ構築アルゴリズムで用いられる記号
Table 1 Notations for graph building algorithm.

記号	説明
G	これまでに構築された不審活動グラフの集合
g=<nodes, edges>	G に含まれる 1 つのグラフ．nodes はノードの集合，edges はエッジの集合
n=<host, gene_time, last_activity_time>	nodes に含まれるノード．host はノードが表現する端末の識別子，gene_time はノードの生成時刻であり，host が不審端末になった時刻に対応する．last_activity_time は，ノードが生成されてから最後に host が不審活動を行った時刻である．last_activity_time の値は，ノード生成後の host の不審活動の発生間隔が gap1 以下で有る限り，時間経過と共に更新される．
e=<src, dst, gene_time>	edges に含まれる有向エッジ．src,dst はエッジの起点・終点となるノード，gene_time はエッジの生成時刻
C	拡散活動検知装置がこれまでに収集した内部通信の集合
c=<src, dst, gene_time>	C に含まれる 1 つの内部通信．src,dst は通信の送信元端末・送信先端末，gene_time は内部通信の発生時刻
activity=<host, suspicious>	拡散活動検知装置が受信した端末活動情報．host は活動を行った端末の識別子，suspicious は，活動が不審活動かどうかのフラグ
current	現在時刻
gap1	現在時刻と last_activity_tme の時間差の上限を規定するパラメータ
gap2	現在時刻と通信発生時刻の時間差の上限を規定するパラメータ
buildGraph(G, C, activity)	拡散活動検知装置が新規の端末活動を受信する度に，不審活動グラフを基に拡散活動の発生を判定する関数
evalGraph(g)	グラフ g が拡散活動に起因するものかどうかを評価する関数

nodes と edges が含まれる．

個々のノードには，ノードの基となる端末の識別子（host），ノードの生成時刻（gene_time），ノードが生成されてから host が最後に不審活動を実施した時刻（last_activity_time）の情報が含まれる．last_activity_time は，host の不審活動の発生間隔が gap1 以下である限り，時間経過とともに更新される．

個々のエッジには，エッジの起点・終点となるノードの識別子，エッジの生成時刻の情報が含まれる．エッジは，不審端末間を結ぶ内部通信を基に生成される．ここでの内部通信は，不審通信だけでなく，正常プロファイルに含まれる正常通信も含まれる．これは，攻撃者が頻繁に通信を行う端末間を拡散した場合の検知をできるようにするためである．

アルゴリズム 1 に，不審活動グラフの構築アルゴリズムの詳細を示す．本アルゴリズムは，新しい端末活動の情報を端末監視機能または Web プロクシから受信するたびに実行される．引数は，これまでに発生したグラフの集合 G，これまでに拡散活動検知装置が受信した内部通信の集合 C，新しい端末活動を表現する activity の 3 つである．

アルゴリズムの 3–4 行目は，端末活動が不審活動であるかどうかを判断し，正常活動の場合は処理を終了することを示している．端末活動が，プロセス起動・ポートオープン・内部通信・Web 通信の場合は正常プロファイルに基づくアノマリの定義に沿って判定が行われ，実行ファイル作成および ICMP Echo Request の場合はミスユーズの定義に沿って判定が行われる．

5–6 行目は，前節で述べた基準に従い端末の不審活動度を求め，端末が不審端末でない場合は処理を終了する．

10–14 行目では，集合 G に含まれる各グラフ g の中に，不審活動を実施した端末 h を基とする既存ノード n が存在し，ノード n の last_activity_time と現在時刻との差分が gap1 未満の場合，last_activity_time を現在時刻に更新する．時刻の差分が gap1 より以上の場合は，ノード n は，現在時刻において拡散活動に関わっている可能性が低いと判断し，何も処理を行わない．

17–25 行目は，端末 h を含むノードが存在しないグラフ g に対してのみ実行される．17–19 行目では，グラフ g 内に，端末 h に対して侵入を行った可能性がある端末に基づくノード n が存在するかどうかを判定する．17 行目では，ノード n の端末から端末 h に対して確立された内部通信 c を集合 C から抽出する．18 行目では，内部通信 c の発生時刻が，ノード n の生成時刻と last_activity_time + gap1 の間にあるかどうかを判定する．次に，19 行目では，内部通信 c が発生した時刻には端末 h はまだ不審端末ではなく，かつ通信受信後 gap2 時間以内に不審端末に転換したかど

アルゴリズム 1 グラフ構築アルゴリズム
Algorithm 1 Graph building algorithm.

```
1    buildGraph(G, C, activity){
2        h=activity.host
3        if(activity.suspicious==false)  //不審活動でない場合は処理を終了
4            return
5        if(SUS(h, current)<THSUS) //不審活動を行った端末が不審端末でない場合は処理を終了
6            return
7
8        update=false
9        for g in G {
10           if(∃n∈g.nodes(n.host==h)){ //①グラフにhを含むノードがある場合，last_activity_timeを更新
11               if(current-n.last_activity_time≦gap1){
12                   n.last_activity_time=current
13                   update=true
14               }
15           }else{
16               for n in g.nodes { //②l.17-19の条件が満たされる場合，新規ノードが生成され既存グラフに結合
17                   if(∃c∈C( c.src==n.host and c.dst==h and
18                   n.gene_time≦c.gene_time ≦ n.last_activity_time+gap1 and
19                   SUS(h, c.gene_time)<THSUS and current-c.gene_time≦gap2)){
20                       v = <h, current, current>
21                       g.nodes ← g.nodes + v
22                       e = <n, v, c.gene_time>
23                       g.edges ← g.edges + e
24                       update=true
25                       evalGraph(g) //新規ノードが追加されたグラフが不審活動に起因するものか判定
26                   }
27               }
28           }
29       }
30
31       if(!update){ //①・②が何れも起きなかった場合，新規ノード・新規グラフを生成してGに追加
32           v = <h, current, current>
33           g = <v, null>
34           G ← G + g
35           evalGraph(g)
36       }
37   }
```

うかを判定する．

17–19 行目の条件をすべて満たす場合，端末 h はノード n の端末から内部通信 c によって侵入され，それから gap2 時間内に不審端末に転換した可能性が高いと判断される．この場合，20–23 行目において，端末 h に基づく新規ノード v と，内部通信 c に基づく新規エッジ e が生成され，グラフ g に追加される．この場合，ノード n はノード v の親ノードであり，同様に，ノード v はノード n の子ノードであると，表現する．

なお，17–19 行目を満たす内部通信が複数存在する場合は，最も発生時刻が古いものが c として選択される．

25 行目では，更新されたグラフ g が拡散活動によるものかどうかの判定が行われる．evalGraph は，単一のグラフ内に含まれるノード数（グラフスコアと呼称）を基に拡散活動を検知する関数である．グラフスコアが閾値 TH_{GRAPH} 以上になった場合に，拡散活動が発生していると判断し，検知アラートを発する．

32–35 行目は，それまでの処理で既存ノードの last_activity_time の更新または新規ノードの生成・グラフへの追加がいっさい行われなかった場合の処理である．この場合，端末 h は，集合 G 内の既存グラフには反映されていない，新しい拡散活動の起点である可能性があると判断される．そして，端末 h に基づく新規ノード n が生成される．そして，ノード n だけを要素に持つグラフ g が生成され，集合 G に追加される．

本アルゴリズムでは，個々のノードはたかだか 1 つの親ノードを持つ．これは，本アルゴリズムが構築する不審活動グラフは，各端末がどの端末から “初めて” 侵入された

のか，すなわち「侵入側の端末と侵入される側の端末」の関係性を表現するものであるからである．個々の拡散活動において，ある端末に，“初めて” 侵入した端末の数はたかだか 1 つである．

このため，ある端末が複数の不審端末から内部通信を同時刻に受信した後に不審端末に転換した場合，個々の内部通信に対応して複数のノードが生成される．たとえば，同時刻に，端末 x1，x2 が端末 x3 に内部通信を送信し，その後 gap2 時間内に x3 が不審端末に転換した場合，アルゴリズム 16–24 行目では，x3 に基づく 2 つのノード n3a(x3)，n3b(x3) が生成され，n1(x1)→n3a(x3) および n2(x2)→n3b(x3) という 2 つの不審活動グラフが構築される．なおこの場合，アルゴリズムの 25 行目に示すとおり，2 つのグラフのグラフスコアは各々別々に評価される．

4. 実装

端末監視機能は C# で実装した．各不審活動の監視には，Windows が提供する標準 API を用いた．プロセスの正確な識別には起動元実行ファイルのハッシュ値などを用いる必要があるが，今回は実装の簡便さの点から，プロセス名が同じであるプロセスは同一のものとして扱った．同様に，実行ファイルの正確な特定にはファイルコンテンツを検査する必要があるが，今回は拡張子が “.exe” であるファイルを実行ファイルとして扱った．

Web プロクシは Squid 2.7 を基に作成した．拡散活動検知装置は C# および Microsoft SQL Server で作成した．

プロセス起動，ポートオープン，内部通信，Web 通信に関しては，学習フェイズ中で 1 回以上発生した活動を正常活動としてプロファイル化した．実行ファイル作成に関しては，60 秒の期間内に複数の実行ファイルが作成された場合は，1 つの不審活動として扱うこととした．

Web 通信では，プロファイルに含まれずに送信データサイズが 10 KB を超える CONNECT 通信を不審活動として扱うこととした．SSL 通信にはネゴシエーションフェイズなど様々なオーバヘッドが含まれるため，数 KB のデータを送信しただけで，送信データサイズは 10 KB を超える．

また，ICMP Echo Request に関しては，端末が ICMP Echo Request を 3 回以上送信し，対応する ICMP Echo Reply 数が Request 数の 1/3 以下であるとき，不審活動と見なすこととした．

5. 評価実験

5.1 実験概要

提案方式の評価実験について述べる．実験では，ある組織内ネットワークに設置された同一事業部に属する PC 30 台に対して，拡散活動をともなう標的型攻撃が仕掛けられることを想定し，提案方式の検知性能を評価した．

提案方式の評価には，通常プロファイルの作成を行う学習フェイズと，検知精度を評価する検知フェイズを実施する必要がある．我々は，PC 30 台の挙動を 2 カ月間監視・記録し，前半の 1 カ月間をプロファイル作成に用い，後半の 1 カ月間を誤検知頻度の評価に用いた．実験期間中 PC は様々な業務のために頻繁に利用されていた．また検知率評価のためにクローズドなネットワーク環境を構築し，3 種類の標的型攻撃を模擬した．

表 2 通常プロファイル件数
Table 2 Number of normal profiles.

活動	通常プロファイル件数	
	30 台累計	1 台平均
プロセス起動	4983	166.1
ポートオープン	491	16.4
内部通信	205	6.8
Web 通信	4545	151.5

検知率評価をクローズドネットワークで実施したのは，組織のセキュリティポリシの理由で攻撃の模擬を業務に使用しているネットワークで行うのは好ましくないと判断したためである．また，既存手法と比較することで本方式の優位性を検証した．本評価は実験期間中に組織ネットワーク内では標的型攻撃は発生していないことを前提とする．

以下に，各実験の詳細および比較対象手法の概要，および実験結果について示す．

5.1.1 通常プロファイル作成実験

通常プロファイル作成実験は，2014 年 12 月 10 日から 2015 年 1 月 10 日までの 1 カ月間，同一組織ネットワーク内で稼働する PC 30 台（以下，評価用端末）を対象に実施した．評価用端末は Windows Vista/7 がインストールされている汎用 PC であり，多数のユーザからアクセスされる専用サーバは含まれない．ただし，一部の端末はファイル共有や実験，Windows が提供するサービスの影響で他端末からアクセスされる．

多くの端末内で行われるオペレーションは書類作成などの一般業務・WEB 閲覧・ソフトウェア開発などである．業務のために，SSH などのサーバが動作している端末もある．各ユーザは，自身が使用する端末の管理者権限を有している．

評価用端末 30 台の性能は，CPU Intel Corei5/Corei7 シリーズ，メモリ 4 GB/8 GB，搭載 OS Windows Vista/7（32 bit/64 bit）である．拡散活動検知装置および Web プロクシの性能は，CPU Intel Corei7-4460 3.2 GHz，メモリ 8 GB，搭載 OS Windows 7（64 bit）である．実験に用いた組織ネットワークの帯域速度は 1 Gbps である．

実験の結果導出された 4 種類の活動の通常プロファイルの件数は**表 2** のとおりである．

内部通信は，通信先端末が評価用端末である通信のみを

表 3 デフォルトパラメータ
Table 3 Default parameters.

パラメータ	デフォルト値
window	3600 秒
TH_{SUS}	4
gap1	window の 1/2 の値
gap2	window と同値
TH_{GRAPH}	2

分析対象とした．30 台の評価用端末間で行われる内部通信の送受信端末の組合せ数は 870（$= 30 \times 29$）通りである．このため，1 カ月の間に可能な組合せの 24%が実際に発生したことになる．通信プロトコルとしては TCP/5357, TCP/2869, TCP/137 が大半を占めた．これらの TCP ポートはそれぞれ，Windows Vista/7 を実行している他端末の探索，UPNP サービス，NetBIOS 名前解決に用いられる．

5.1.2 誤検知頻度評価実験

誤検知頻度評価実験では，2015 年 1 月 11 日から 2015 年 2 月 10 日までの 1 カ月間に発生したアラート数を，誤検知頻度として測定した．実験期間中に組織ネットワーク内では標的型攻撃は発生していないことを前提とするため，実験期間中に発生したアラートはすべて誤検知と判断する．

提案検知手法のパラメータのデフォルト値は**表 3** のとおりである．

不審端末の分析期間である window のデフォルト値は，速効型の標的型攻撃の持続時間（攻撃開始から目的達成までにかかる時間）が最長で数時間以内であるという知見 [12], [53] を基に設定した．

gap1，gap2 は window 値に依存するものとした．

不審端末の判定閾値である TH_{SUS} のデフォルト値は，攻撃者は，端末侵入後にプロセス起動などの活動を数回以上行うという知見 [54] を基に設定した．

不審活動グラフの判定閾値である TH_{GRAPH} のデフォルト値は，拡散活動により起因するグラフスコアの最小値を設定した．また，$SUS(x, t)$ における各活動の重み付けはすべて 1 とした．

なお，数日・数週間など，window 時間より長い期間をかけて行う攻撃であっても，各端末で，各 window 期間に TH_{SUS} 件以上の不審活動をともなうものであれば，原理的に検知が可能である．

各不審活動特定に関するパラメータについては，4 章で述べたデフォルト値をそのまま用いた．これらのデフォルト値は経験的に，それぞれの不審活動をとらえるうえでおおむね妥当と考えられる値である．

実行ファイル作成に関するパラメータ（1 つの不審活動として扱われる実行ファイル作成時刻の時間差の幅，デフォルト 60 秒）を大きくすると，1 つのアプリケーションのインストールを複数回の不審活動としてカウントする可能性が低くなるため，誤検知頻度は減少する．その一方で，攻撃者が，異なる用途に用いるマルウェアや攻撃ツールを，別々のタイミングでダウンロードした場合にも 1 つの不審活動としてカウントされるため，端末が不審端末に転換する時間が遅くなり，検知性能が低下する．反対に，パラメータを小さくすると，検知性能は向上するが，誤検知頻度も増加する．

Web 通信に関するパラメータ（不審活動として扱われる送信データサイズ，デフォルト 10 KB）を大きくすると，大きなファイルをアップロードしたときのみ不審活動の判定対象となるため，誤検知頻度は減少する．一方で，大規模な情報漏えいを行わない攻撃を検知できなくなるため，検知性能は低下する．反対に，パラメータを小さくすると，検知性能は向上するが，誤検知頻度も増加する．

ICMP Echo Request に関するパラメータ（アドレススキャンと見なすのに必要な，ICMP Echo Request 数の下限値，および対応する Response 数の割合の上限値，デフォルトはそれぞれ 3 回，1/3）に関しては，以下のとおりとなる．まず，Request 数を増やすあるいは Response 割合を減らすと，通常オペレーション中に，ダウンしている端末に対して Ping を送った場合などに不審活動としてカウントされる可能性が低くなるため，誤検知頻度は減少する．一方で，攻撃者のアドレススキャンの発見が遅くなるあるいは発見できなくなるため，検知性能は低下する．反対に，Request 数を減らすあるいは Response 割合を増やすと，検知性能は向上するが，誤検知頻度も増加する．

5.1.3 検知率評価実験

検知率評価実験では，端末（PC1，PC2，PC3）を対象とした標的型攻撃をクローズドネットワークにおいて模擬した．クローズドネットワークは，組織ネットワークに相当するサブネットおよび C&C サーバなどが存在し攻撃を仕掛けるサブネットから構成される．そして，誤検知頻度評価実験で得られた不審活動ログを基に，当該攻撃が 30 台の評価用端末の一部に対して行われた場合を想定して検知率を評価した．

検知率評価では通常オペレーション時に発生する不審活動の検知率への影響を測るために，以下の 2 種類の検知率を評価した．

- ベースライン検知率：攻撃を受ける端末内で，攻撃に関係しない活動がいっさい行われていない場合を想定したときの検知率
- 実質検知率：端末内で，攻撃に関係しない通常オペレーションに関する活動が行われている場合を想定したときの検知率．誤検知頻度評価実験で用いたログを基に測定する．

実験では，文献 [2], [3], [54] に基づき，以下に示す 3 種類のシナリオを実施した．

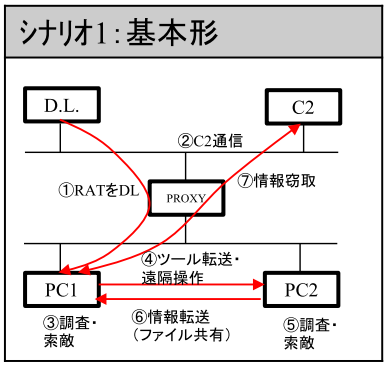


図 4 攻撃シナリオ 1
Fig. 4 Attack scenario 1.

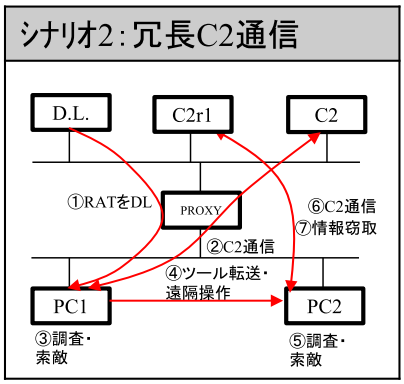


図 5 攻撃シナリオ 2
Fig. 5 Attack scenario 2.

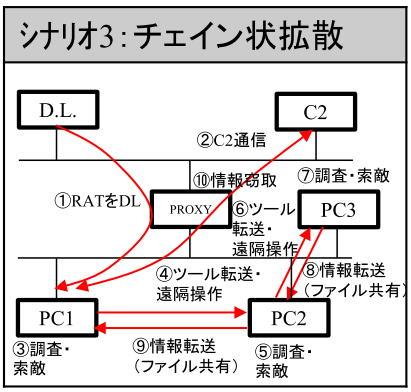


図 6 攻撃シナリオ 3
Fig. 6 Attack scenario 3.

シナリオ 1 では，PC1 を乗っ取った攻撃者が PC2 に拡散し，PC2 内にある機密ファイルを，PC1 を経由して C&C サーバ（C2）に送信する．

シナリオ 2 はシナリオ 1 と同様だが，PC1 を介さず，PC2 から機密ファイルを，直接，異なる C&C サーバ（C2r1）に送信する．このシナリオでは，攻撃者が，攻撃に関与する端末やサーバを分散させ，個々の端末・サーバに着目した検知方法を回避しようとするというケースを想定している．

シナリオ 3 では，攻撃者は PC1，PC2 を介して PC3 にある機密ファイルを窃取する．このシナリオでは，機密ファイルが存在する PC3 に対して PC1 から直接アクセスできない場合に，PC2 を経由するというケースを想定している．

表 4 シナリオ 1 で発生する活動件数
Table 4 Number of activities in scenario 1.

	プロセス起動	ポートオープン	内部通信	Web 通信	実行ファイル作成	ICMP Echo Request
Step.1	1	0	0	0	0	0
Step.2	1	1	0	1	1	0
Step.3	6	0	0	0	0	1
Step.4	2	0	1	0	1	0
Step.5	5	0	0	0	0	0
Step.6	4	0	1	0	0	0
Step.7	5	0	0	0	0	0
PC1 累計	14	1	1	1	1	1
PC2 累計	10	0	1	0	1	0

シナリオ 1 は 7 ステップから構成される．各ステップでの実施内容および発生する活動件数を**表 4** に示す．

1. PC1 が不正なショートカットファイルを実行し，RAT ツールをダウンロード用サーバからダウンロード・実行する．
2. RAT ツールが C2 サーバに SSL で接続し，遠隔操作が開始される．攻撃者は攻撃に使うツール（Paexec [55]，SDelete [56]，ファイル圧縮ツール，デスクトップ操作のための高機能型 RAT，アドレススキャナ[*1]）をダウンロードする．また，トンネリングに用いるポートを開く．
3. PC1 に侵入した攻撃者は，Windows 標準コマンド（netstat，arp，tasklist，ipconfig，systeminfo）を実行し，PC1 の端末情報を調査する．この端末情報の送信量は累計で 10 KB を超えることを想定する．また，アドレススキャナを実行し，PC1 の近傍にいる PC2 を発見する．なお，シナリオ 3 では，PC2 は PC3 のアドレスを知っていることを前提とし，アドレススキャンを行わない．
4. 攻撃者は Paexec [55] を実行して，PC1 から PC2 に拡散する．また必要な攻撃ツールをコピーする．

[*1] アドレススキャナは，本実験のために独自開発したプログラムである．起動すると，自身の近傍にある IP アドレスに対して ICMP Echo Request を順々に送信して，アクセス可能な端末を発見するという処理を 1 秒おきに行う．送信先アドレスは，自身のアドレスが <a.b.c.d> とすると，<a.b.c.d+1>，<a.b.c.d+2> のように，第 4 オクテットを 1 ずつインクリメントして決める．実装言語は C# である．

5. 攻撃者は PC2 の端末情報を調査する．
6. 攻撃者は PC2 内のファイルを圧縮し，Windows ファイル共有機能を介して PC1 にコピーする．
7. 攻撃者は圧縮ファイルを C2 サーバにアップロードする．また PC1，PC2 に残されたファイルを削除し，攻撃を完了する．

本方式は，PC1→PC2 のグラフを抽出するためには，Step4 完了までに PC1 を不審端末として判定する必要がある．攻撃に用いた RAT ツール，アドレススキャナは，実験のために作成した独自なものであり，一般的なアンチウイルスソフトで検知されないことを確認している．

本シナリオの実行に際しては，以下の事項を前提条件とする．

- 攻撃者は PC2，PC3 のアカウント情報を何らかの手段で知っている．
- PC2，PC3 は Paexec（Psexec の互換ツール）による遠隔操作が可能である．クライアント型 Windows 端末を Paexec で遠隔操作するには，事前のレジストリ設定が必要である．実験中にこの設定がされている評価対象端末は半数以下であったが，各端末で本設定が有効であり，拡散活動が可能であることを仮定して検知性能を見積もった．前述のとおり，本方式は遠隔管理ツールや脆弱性攻撃など何らかの手段を用いて拡散活動を行う攻撃を想定脅威としている．また，本方式は特定の遠隔操作方法やプロトコルには依存しない．このため，本シナリオでは実施が容易な Paexec を用いて攻撃を模擬したが，他の遠隔管理ツールや脆弱性攻撃を用いた場合も検知結果の一般性は失われないと考える．
- D.L. サーバ，C2 サーバは，学習フェイズにおいてアクセスがなかったドメイン上に設置されたものである．
- 攻撃開始から完了までにかかる時間は 1 時間程度である．

攻撃に用いられる Windows 標準コマンドや拡散先との通信が不審活動と判断されるかは，各端末の通常プロファイルに依存する．このため，同一の攻撃であっても，端末によって，不審端末に転換するタイミングは異なる．

シナリオ 1・2 では 2 台の PC，シナリオ 3 では 3 台の PC が攻撃に含まれる．また，評価用端末は 30 台である．このため，シナリオ 1・2 に対しては 870（$= 30 \times 29$）通り，シナリオ 3 に対しては 24360（$= 30 \times 29 \times 28$）通りの PC の組合せに対する攻撃を実施し，検知率を求めた．

5.1.4 比較対象手法

提案方式を既存の代表的な標的型攻撃検知のアプローチと比較したときの検知精度の優位性を評価するために，文献 [33] での提案を基にしたプロセスホワイトリスト手法（PWL）を比較対象手法として用いる．PWL は，定義ファイルに依存したアンチウイルスソフトなどでは検知できない未知マルウェアや正規ツールを悪用した標的型攻撃に対処するアプローチの代表例である．PWL は，ネットワーク管理者があらかじめ作成したホワイトリストにないプロセスが起動すると標的型攻撃の発生を検知する．一方でこの方式には，新規プログラムが導入されたり更新が行われる場合に誤検知が発生しやすいこと，後述のようにプロセスに痕跡が残らない攻撃への検知が難しいという問題がある．

本実験で用いる PWL では，学習フェイズで 30 台の端末内で起動したプロセスを，端末の区別なくホワイトリストに格納する．そして，検知フェイズでホワイトリストに載っていないプロセスが起動するとアラートをあげる．PWL の性能は端末のオペレーションの影響を受けない．このため，PWL のベースライン検知率・実質検知率は同値であるため，単に「検知率」と表記する．

表 5 不審活動数
Table 5 Number of suspicious activities.

活動種類	不審活動件数
プロセス起動	8004
ポートオープン	548
内部通信	3156
Web 通信	137
実行ファイル作成	1250
ICMP Echo Request	63

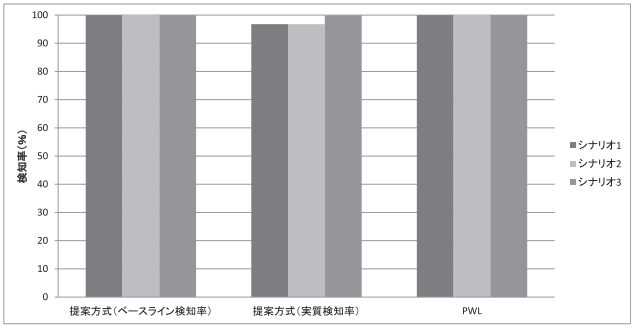


図 7 検知率評価
Fig. 7 Detection rate.

5.2 実験結果

5.2.1 基本性能評価

表 5 に誤検知頻度実験において発生した不審活動数を示す．プロセス起動に関する件数が最も多く，次いで内部通信，実行ファイル作成となっている．

図 7 に検知率の比較を示す．提案方式のベースライン検知率および PWL の検知率はすべてのシナリオに関して検知率 100%となる．一方，提案方式の実質検知率は，シナリオ 1・2 において 3%程度低下する．検知ミスは，PC2 が攻撃と無関係のオペレーションにより，PC1 から内部通信を

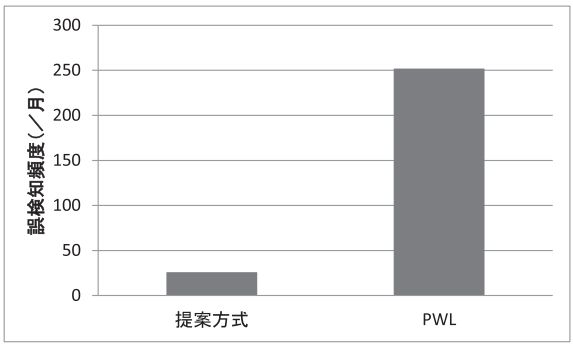


図 8 誤検知頻度
Fig. 8 False positive frequency.

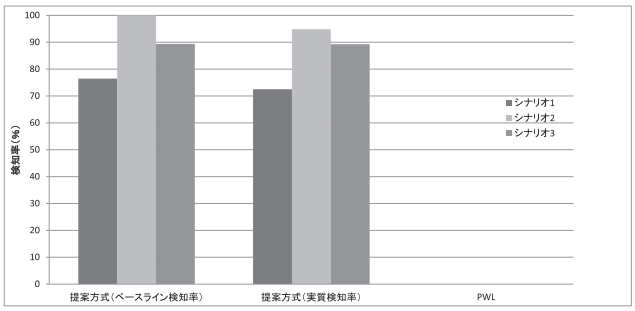


図 9 検知率評価（通信関係のみ監視）
Fig. 9 Detection rate (Communication only).

受信する前に不審端末に転換する場合に発生する．この場合，PC1 と PC2 は同一のグラフに含まれなくなる．シナリオ 3 では，PC1→PC2 のグラフに加え，PC2→PC3 のグラフも検知に利用できるため，実質検知率の低下は 0.1%程度になる．

また，提案方式では，攻撃に含まれる C&C サーバが 1 つであること前提とする既存技術 [49] では対応が難しい，複数の C&C サーバを用いたシナリオ 2 の検知が可能である．

図 8 に誤検知頻度を示す．PWL の誤検知頻度が 250 件/月であるのに対し，提案方式はその約 1/10 の 26 件まで精度が向上している．これは，提案方式では複数の活動を基に検知を行うため，通常オペレーション時に新しいプロセスが起動したのみではアラートをあげないためである．

発生した 26 件の誤検知のうち，OS の更新時に起動したプロセスや作成された実行ファイルが要因の 1 つとなり，誤検知が発生したケースが 3 件あった．同様に，アプリケーションの更新に起因するものが 3 件あった．なお，評価用端末ではアンチウイルスソフトがインストールされ，定期的に更新されているが，本実験中ではこれに起因する誤検知は発生しなかった．

なお，本方式の誤検知例を Appendix に示す．

次に，**図 9** に，攻撃者が高度なスキルを持ち，プロセスやファイルシステムに攻撃の痕跡が残らず，通信関係のみ監視可能な場合の検知率を示す．提案方式では $\mathrm{SUS}_{\mathrm{NET}}$ に基づき不審端末を検知する．$\mathrm{SUS}_{\mathrm{NET}}$ の閾値は 1 とする．

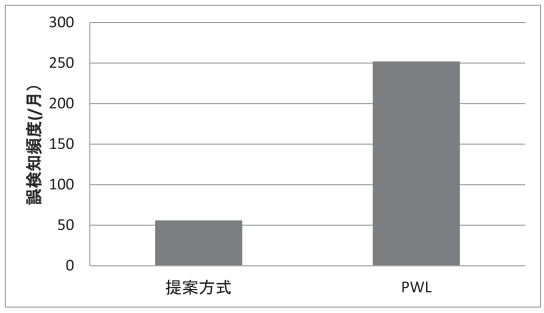


図 10 誤検知頻度（通信関係のみ監視）
Fig. 10 False positive frequency (Communication only).

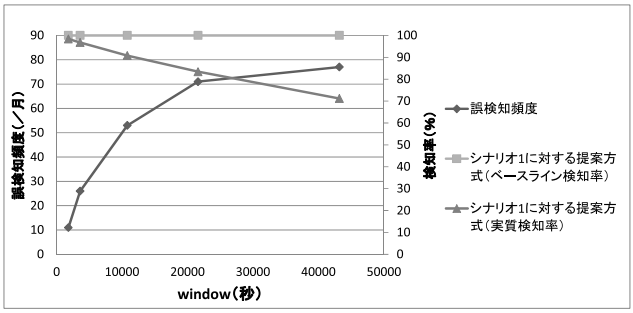


図 11 window 値の検知性能への影響
Fig. 11 Effect of window values on detection performance.

PWL ではプロセスに痕跡が残らない攻撃の検知ができないため検知率は原理的に 0%となる．一方提案方式では，通信関連の監視のみでも検知率は 70%を超える．また，シナリオ 2 ではシナリオ 1，シナリオ 3 よりも検知率が高くなる．これは，シナリオ 2 では，他のシナリオと異なり，拡散先の端末（PC2）も C2 サーバと通信を行うため，検知に利用できる通信関連の活動が多くなるためである．またシナリオ 3 では PC2→PC3 の拡散活動を検知に利用できるため，シナリオ 1 より検知率が高くなる．

プロセス起動のみに着目して検知を行う PWL と異なり，提案方式は様々な種類の活動を基に検知を行う．このため，一部の種類の活動を観測できないような攻撃であっても，本方式は対応が可能である．

図 10 に $\mathrm{SUS}_{\mathrm{NET}}$ の閾値を 1 に設定した場合の誤検知頻度を示す．誤検知頻度は 50 を超えるが，PWL の 250 に比べると依然低い値となる．

5.2.2 検知パラメータ・監視対象の影響

図 11 に window 値の検知性能への影響を示す．window 値を 30 分から 12 時間まで変化させたが，ベースライン検知率に変化はなかった．しかし，window 値を 30 分よりもさらに小さく設定すると，攻撃により生じる不審活動の頻度が 4 (= $\mathrm{TH}_{\mathrm{SUS}}$)/window を下回り，ベースライン検知率は低下すると考えられる．一方，window 値が大きくなるほど，攻撃発生の有無とは無関係に，端末が不審状態にある時間帯が増加するため，5.2.1 項で述べたのと同様の理由で実質検知率は低下する．window 値が 12 時間のとき，

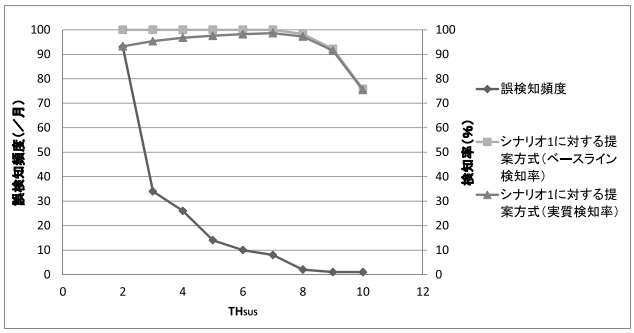


図 12 TH_{SUS} 値の検知性能への影響
Fig. 12 Effect of TH_{SUS} values on detection performance.

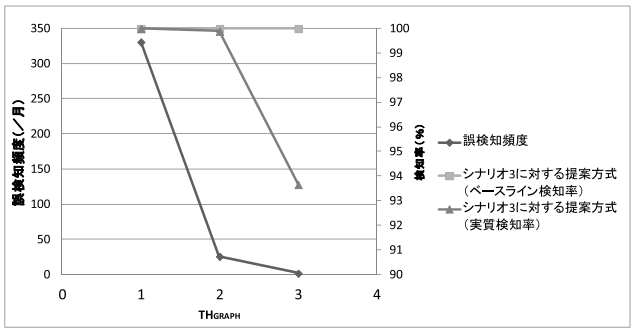


図 13 TH_{GRAPH} 値の検知性能への影響
Fig. 13 Effect of TH_{GRAPH} values on detection performance.

実質検知率は 71%程度となる．

一方，window 値が大きくなるほど，通常オペレーションを行っている端末が不審端末に転換することが多くなるため，誤検知頻度は増加する．

図 12 に TH_{SUS} 値の検知性能への影響を示す．TH_{SUS} が大きくなるにつれ誤検知頻度は低下する．一方，TH_{SUS} が 8 を超えると検知率が低下する．このため今回の実験では TH_{SUS} は 6〜8 程度の値であるとき，検知・誤検知の両面で最適な結果が得られることが分かる．

図 13 に TH_{GRAPH} 値の検知性能への影響を示す．TH_{GRAPH} が大きくなるにつれ誤検知頻度は低下する．しかしその一方で，シナリオ 3 において，$TH_{GRAPH} = 3$ の場合，$TH_{GRAPH} = 2$ のときと比べて，検知率は 6%程度低下する．また，シナリオ 1・2 のように，拡散先端末数が少ない攻撃を検知できなくなる．

このため，各端末での不審活動数が非常に少なく（1〜3 個程度），かつ多数（3 以上）の端末に拡散するような攻撃を検知することを目的とする場合に限り TH_{GRAPH} 値を大きくすることが妥当と考える．

また $TH_{GRAPH} = 1$ のとき，誤検知頻度は 340 件を超える．$TH_{GRAPH} = 1$ は，個々の端末が不審端末状態になった場合にアラートをあげることを意味する．TH_{GRAPH} がデフォルト値の 2 であるとき，本方式は，拡散活動時に発生する 2 つの端末が連続して不審端末状態になるという現象をとらえることで，誤検知頻度を 340 件から 26 件まで削減しているといえる．

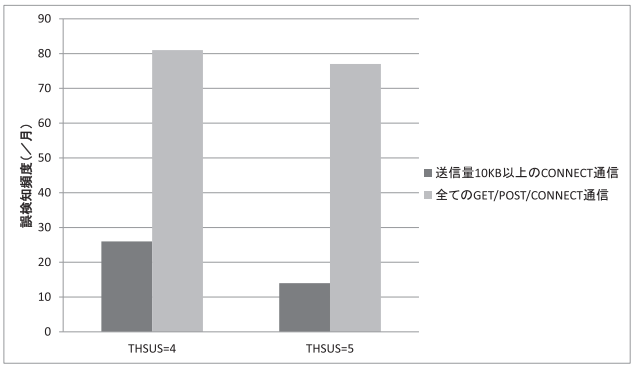


図 14 Web 通信の監視対象への影響
Fig. 14 Effect of monitoring targets on Web communication.

図 14 に Web 通信での監視対象メソッドを 10 KB 以上の CONNECT メソッドではなく，すべての HTTP メソッドに変えた場合の検知性能への影響を示す．監視対象が増えるほど誤検知頻度は増加するが，PWL に比べると依然低い値である．

6. 考察

提案方式は，各パラメータが最適に設定された場合，(i) 拡散活動をともない，(ii) 各端末で数時間に数件程度の不審活動を行う標的型攻撃を検知でき，誤検知頻度を 1 カ月で 1 桁台まで抑えることができる．たとえば，攻撃者が C&C サーバにアクセスして実行ファイルをダウンロード・起動，起動したプロセスが他の内部端末にアクセスするだけで，最大計 4 回の不審活動として観測される．このため，本方式に発見されないように攻撃を構成するのは困難といえる．また，本方式は特定のアプリケーションプロトコルや振舞いに依存しないため，未知の脆弱性を突き拡散活動を行う攻撃も検知できる．

本提案では 6 種類の活動を基に検知を行った．個々の監視項目は比較的シンプルであるが，拡散活動に着目した相関分析を行うことで優れた性能を実現できる．また，レジストリ作成やサービス起動など，他の活動にも着目することで，検知精度を向上させることができる可能性がある．

Web 通信において，どの種類の HTTP メソッドを監視するかは，検知したい攻撃の種類に依存する．情報窃取を目的とした攻撃の場合，数十 KB 以上のデータ送信が発生するのは自然と考えられるため，送信量 10 KB 以上の CONNECT メソッドの監視が妥当といえる．一方，C&C サーバへのデータ送信がほとんどなく，侵入先ネットワークのシステム破壊を主な目的とするような攻撃を想定する場合，すべてのメソッドを監視することが妥当といえる．

誤検知頻度評価実験で発生した誤検知 26 件の約 23%にあたる 6 件は，OS・アプリケーションの更新に起因するものであった．OS・アプリケーションの更新による誤検

知を低減させる方法としては，2つのアプローチが考えらえる．1つ目のアプローチは，ネットワーク管理者が，管理下にある端末に対して行うOSやアプリケーションの更新時に追加されるファイルをあらかじめ把握しておき，検知の除外対象とする方法である．エンタープライズネットワークではOSなどの更新ファイルを，管理者がローカルに設置したサーバから配布することが多いため，OSやアプリケーションの種類によっては実現可能と考える．2つ目のアプローチは，ファイルのコード署名を検証して，アンチウイルスベンダなど信頼できる組織が作成したファイルを除外対象とする方法である．ただし，Psexecのようにコード署名がついたファイルを攻撃者が悪用するケースもあるため，攻撃検知の観点で信頼できる組織をどのように選定するかという課題がある．

なお，検知率評価実験で評価したベースライン検知率は，攻撃に関係しない通常オペレーションがいっさい発生しないクローズドな環境における検知率を評価したものであるため，OS・アプリケーションの更新による影響は考慮していない．一方で，実質検知率に関しては，誤検知頻度評価実験の期間に評価用端末30台で実際に発生したOS・アプリケーションの更新の影響を反映して，検知率を算出している．

評価用端末数が増加した場合の検知精度への影響，および拡散活動検知装置への負荷に関しては，今後の検討課題である．また，今回の実験では，学習フェイズ・検知フェイズとも1カ月という比較的短い期間で分析を行った．今後は，様々な環境において，より長期間にわたる実験を行い，各端末の活動が時間経過に従いどのように変化するのかを分析し，より効果的な通常プロファイルの作成・更新方法についても検討を進める．

最後に，より長期間にわたって活動を行う攻撃，より複雑な攻撃のシナリオについても検討を進めたい．

7. おわりに

本稿では，拡散活動を不審活動グラフとして抽出することで標的型攻撃を検知する方式を提案した．性能評価実験を通じ，提案方式は検知性能および高度な攻撃に対する有用性の点で既存方式より優れていることを確認した．今後は，監視対象とする不審活動の拡充，様々なネットワーク環境・攻撃シナリオを用いた性能評価を通じて，提案方式の有用性をさらに検証していく．

参考文献

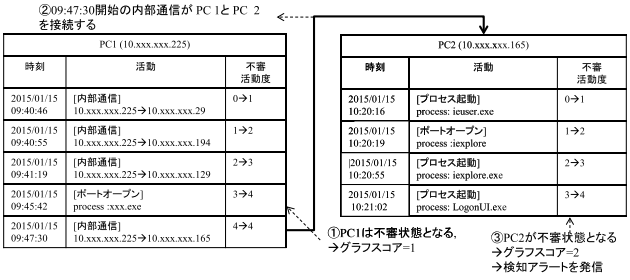
[1] Le Blond, S. et al.: A Look at Targeted Attacks Through the Lense of an NGO, *Proc. 23rd Usenix Security* (2014).
[2] Mandiant.: M-Trends Report 2010 (2010).
[3] Chins, E.H. et al.: Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, *Proc. CIW2011* (2011).
[4] INFOSEC INSTITUTE: Foreign Hackers Constantly Target US Critical Infrastructure, available from ⟨http://resources.infosecinstitute.com/foreign-hackers-constantly-target-us-critical-infrastructure/⟩ (accessed 2015-05-25).
[5] BUSINESS INSIDER: The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought, available from ⟨http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11⟩ (accessed 2015-05-25).
[6] Wu, J. et al.: An Effective Architecture and Algorithm for Detecting Worms with Various Scan Technique, *Proc. NDSS'04* (2004).
[7] Gu, G. et al.: Worm detection, early warning and response based on local victim information, *Proc. ACSAC 2004* (2004).
[8] Bielski, J.: Looking Ahead: The State of Incident Detection and Response in 2015, available from ⟨https://www.mandiant.com/blog/state-incident-detection-response-2015/⟩ (accessed 2015-05-25).
[9] Lindorfer, M. et al.: Detecting Environment-Sensitive Malware, *Proc. RAID 2011* (2011).
[10] SECURITY WEEK: Dropbox Abused in Targeted Attacks Using PlugX RAT With Time Bomb, available from ⟨http://www.securityweek.com/dropbox-abused-targeted-attacks-using-plugx-rat-time-bomb⟩ (accessed 2015-05-25).
[11] LAC, co.ltd.: Cyber GRID View Vol.1 (2014).
[12] トレンドマイクロ：国内標的型サイバー攻撃分析レポート2015年度版 (2015).
[13] BAYE Systems: PIANOS PROTECTING INFORMATION ABOUT NETWORKS, THE ORGANISATION and ITS SYSTEMS, available from ⟨http://www.cpni.gov.uk/Documents/Publications/2014/2014-04-23-pianos_report.pdf⟩ (accessed 2015-05-25).
[14] TrendMicro: LUCKYCAT REDUX Inside an APT Campaign with Multiple Targets in India and Japan, available from ⟨http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_luckcat_redux.pdf⟩ (accessed 2015-05-25).
[15] Trend Micro: LATERAL MOVEMENT: How Do Threat Actors Move Deeper Into Your Network? (2013).
[16] Ortolani, S. et al.: KILMAX: Profiling Memory Write Patterns to Detect Keystroke-Harvesting Malware, *Proc. RAID 2011* (2011).
[17] Willems, C. et al.: Toward Automated Dynamic Malware Analysis Using CWSandbox, *IEEE Security and Privacy Magazine*, Vol.5, No.2 (2007).
[18] Inoue, D. et al.: Automated Malware Analysis System and its Sandbox for Revealing Mal-ware's Internal and External Activities, *IEICE Trans. Information and Systems*, Vol.E92-D, No.5 (2009).
[19] Lanzi, A. et al.: AccessMiner: Using System-Centric Models for Malware Protection, *Proc. ACM CCS'2010* (2010).
[20] Bayer, U. et al.: Improving the Efficiency of Dynamic Malware Analysis, *Proc. ACM Symposium on Applied Computing 2010* (2010).
[21] Rieck, K. et al.: Cujo: Efficient Detection and Prevention of Drive-by-Download Attacks, *Proc. ACSAC'10* (2010).
[22] Wuchner, T. et al.: Malware Detection with Quantitative Data Flow Graphs, *Proc. ACMCCS'2014* (2014).

[23] 川口ほか：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，情報処理学会論文誌，Vo.53, No.11 (2012).
[24] 川口ほか：ユーザの PC 利用時間帯を考慮したマルウェア対策ユーザサポートシステムの性能評価，情報処理学会論文誌，Vol.54, No.4 (2013).
[25] 川口ほか：マルウェア解析システムの検知結果の相関性に基づくマルウェア統合検知方式，情報処理学会論文誌，Vol.55, No.10 (2014).
[26] 嶌田ほか：ライブネットにおける低速スキャン検知手法，CSS2014 予稿集 (2014).
[27] Gu, G. et al.: BotHuner: Detecting malware infection through ids-driven dialog correlation, *Proc. Usenix Security 2007* (2007).
[28] Takemori, K. et al.: Detection of of Infected PCD Using Destination-based IP Address and Domain Name Whitelists, *IPSJ Journal*, Vol.52, No.4, pp.1706–1716 (2011).
[29] Bilge, L. et al.: EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis, *Proc. NDSS'2011* (2011).
[30] Perdisci, R. et al.: Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces, *Proc. ACSAC 2009* (2009).
[31] Hsu, C.H., Huang, C.Y. and Chen, K.: Fast-Flux Bot Detection in Real Time, *Proc. RAID 2010* (2010).
[32] Antonakakis, M. et al.: Building a Dynamic Reputation System for DNS, *Proc. USENIX Security 2010* (2010).
[33] 中里ほか：ホスト型 IDS を用いた不審プロセスの特定，SCIS2015 予稿集 (2015).
[34] Benito Camiña, J. et al.: Towards a Masquerade Detection System Based on User's Tasks, *Proc. RAID2014* (2014).
[35] Schonlau, M. et al.: Detecting masquerades in intrusion detection based on unpopular commands, *Inf. Process Lett.*, Vol.76, No.1-2, (2000).
[36] Aiello, W. et al.: Analysis of communities of interest in data networks, *Proc. Passive and Active Measurement Workshop 2005* (2005).
[37] Pang, R. et al.: A first look at modern enterprise traffic, *Proc. IMC'05* (2005).
[38] Yen, T.F. et al.: Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks, *Proc. ACSAC'13* (2013).
[39] 津田ほか：標的型攻撃再現のための攻撃シナリオ定義インタフェースの実装，CSS2014 予稿集 (2014).
[40] Kawaguchi, N. et al.: A Distributed Detection of Hit-list Worms, *Proc. IEEE ICC'08* (2008).
[41] Ellis, D.R. et al.: A behavioral approach to worm detection, *Proc. 2004 ACM workshop on Rapid Malcode* (2004).
[42] Ellis, D.R. et al.: Graph based worm detection on operational enterprise networks, MITRE TECHNICAL REPORT (2006).
[43] Toth, T. et al.: Connection-history based anomaly detection, *Proc. 3rd IEEE Information Assurance Workshop* (2002).
[44] Staniford-chen, S. et al.: GrIDS: A Graph-Based Intrusion Detection System for Large Networks, *Proc. 19th National Information Systems Security Conferences* (1996).
[45] Collins, M.P. et al.: Hit-list worm detection and bot identification in large networks using protocol graphs, *Proc. RAID2007* (2007).
[46] Cao, H. et al.: CBSTM: Cloud-based Behavior Similarity Transmission Method to Detect Industrial Worms, *Proc. IEEE ICCC'13* (2013).
[47] Gu, G. et al.: BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection, *Proc. Usenix Security 2008* (2008).
[48] Oliner, A.J. et al.: Community Epidemic Detection using Time-Correlated Anomalies, *Proc. RAID 2010* (2010).
[49] 山田ほか：組織内ネットワークにおける標的型攻撃の振る舞い検知に向けた複数センサの連携手法，SCIS2015 予稿集 (2015).
[50] Ramsbrock, D. et al.: A Frist Step Towards Live Botmaster Traceback, *Proc. RAID 2008* (2008).
[51] Zhang, Y. et al.: Detecting Stepping stone, *Proc. USENIX Security 2000* (2000).
[52] 独立行政法人情報処理推進機構（IPA）：「標的型メール攻撃」対策に向けたシステム設計ガイド (2013).
[53] Verizon.: 2013 DATA BREACH INVESTIVATIONS REPORT, available from ⟨http://www.secretservice.gov/Verizon_Data_Breach_2013.pdf⟩ (accessed 2015-05-25).
[54] 寺田ほか：研究用データセット「動的活動観測 2014」の検討，CSS2014 予稿集 (2014).
[55] Paexec, available from ⟨http://www.poweradmin.com/paexec/⟩.
[56] SDelete, available from ⟨https://technet.microsoft.com/ja-jp/sysinternals/bb897443.aspx⟩

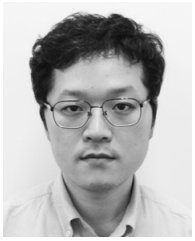
付　　録

A.1 誤検知頻度評価実験において誤検知された不審活動グラフ

誤検知頻度評価実験において誤検知された不審活動グラフを以下に示す．



PC1 は時刻 09:45:42 に不審状態となり，時刻 09:47:30 に PC2 に内部通信を開く．PC2 はその後，時刻 10:21:02 に不審状態となり，PC1，PC2 から構成される不審活動グラフが構築される．PC2 は学習フェイズにおいていっさい使用していなかった Internet Explorer を検知フェイズで起動したため，いくつかの不審活動が検出され，検知アラートが発信された．

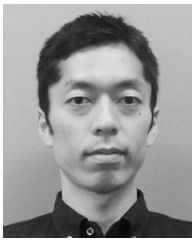


川口 信隆 （正会員）

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）．同年 4 月株式会社日立製作所に入社．同社研究開発グループシステムイノベーションセンタにてサイバーセキュリティおよびマルウェア対策の研究開発に従事．2008 年 IPSJ 論文船井若手奨励賞，2012 年 DICOMO シンポジウム優秀論文賞受賞．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（2011～2014 年）およびコンピュータセキュリティ研究会運営委員（2013 年～）を歴任．CISSP，IEEE，ACM 各会員．

築地原 護

2006 年 3 月九州東海大学大学院工学研究科（現・東海大学大学院）修士課程修了．修士（工学）．2006 年 4 月株式会社日立アドバンストシステムズに入社．同社にて情報システムの設計開発に従事．



井手口 恒太

2006 年 3 月東京大学大学院理学系研究科博士課程修了．博士（理学）．2006 年 4 月株式会社日立製作所に入社．同社研究開発グループにて情報セキュリティおよび暗号技術の研究開発に従事．

谷川 嘉伸 （正会員）

1993 年京都大学大学院理学研究科修士課程修了．同年株式会社日立製作所に入社．同社横浜研究所にて，サイバー攻撃対策に関わるセキュリティ応用技術の研究開発業務に従事．2015 年から情報・通信システム社情公システム事業部勤務．

冨村 英勤

2007 年 3 月東京理科大学大学院理学研究科物理学専攻修士課程修了．2007 年 4 月株式会社日立製作所に入社．同ディフェンスシステム社にてセキュリティシステムの設計・開発に従事．