感染後通信検知のための通信プロファイリング技術の設計と評価

千葉 大紀† 八木 毅† 秋山 満昭† 青木 一史† 針生 剛男†

†NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11
{chiba.daiki, yagi.takeshi, akiyama.mitsuaki, aoki.kazufumi, hariu.takeo}@lab.ntt.co.jp

あらまし 近年，マルウェア感染を未然に防止するのが困難となっており，感染後の対策の重要性が増している．感染後の通信を検知するために，通信の特徴からテンプレートを生成する手法が検討されている．このような手法では，感染後の多様な通信に対応するためのテンプレートが生成されるが，テンプレートが正規の通信にも該当し誤検知が発生するという問題がある．そこで本稿では，攻撃者の保有する攻撃基盤の特性に着目し，感染後の通信のうち可変的な特徴と，不変的な特徴を特定してからテンプレートを生成する手法を提案する．実データを用いた評価では，本手法は，従来と比較して最大で約 70%誤検知数を削減しつつ，検知率を改善できた．

Design and Evaluation of a Profiling Method to Detect Post-infection Communications

Daiki Chiba† Takeshi Yagi† Mitsuaki Akiyama† Kazufumi Aoki† Takeo Hariu†

†NTT Secure Platform Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, JAPAN
{chiba.daiki, yagi.takeshi, akiyama.mitsuaki, aoki.kazufumi, hariu.takeo}@lab.ntt.co.jp

Abstract The importance of post-infection countermeasures has greatly increased. Such countermeasures include generating templates based on communications made by malware-infected hosts. However, such templates have a potential problem of falsely regarding benign communications as malicious. To tackle this problem, we propose a system to profile the variability of elements in malware-generated communications to generate network-based templates. The key idea is that malicious infrastructures, which cause such communications, have both characteristics of being reused and modified. The results of implementing a prototype of our system and validating it using real traffic data are reported here.

1 はじめに

マルウェア感染に起因するサイバー攻撃が急増している．攻撃者はユーザの端末をマルウェアに感染させ，マルウェアにより端末を不正に制御することで，端末の情報収集やサイバー攻撃を実施している．

マルウェア対策には事前対策と事後対策がある．事前対策としては，アンチウィルスソフトを利用する手法や，ネットワーク上で悪性な通信先への通信を監視する手法が利用される．しかし，日々新たなマルウェアが作成されるだけでなく，マルウェア検知用のシグネチャを回避する技術が利用されるため，アンチウィルスソフトのみですべてのマルウェア感染を防止するのは困難である．さらに，マルウェア感染に用いられる悪性サイトは攻撃者によって変更されることが多く，すべての悪性な通信先情報（IP アドレス，FQDN，URL）を事前にブラックリスト化することには限界がある．このため事後対策が必須である．

事後対策では，マルウェアを動作させて感染端末

の挙動を解析する動的解析によってマルウェアの動作や悪性な通信を特定し，これらにマッチした挙動を示す端末を感染者として検知して対策を講じる．この際，感染端末のホスト上の挙動は正確に監視できない可能性があるため，悪性な通信の監視は必須である．このため，マルウェアの通信パターンをテンプレート化する手法が多数検討されている．このような手法では，検知率を向上させるために，マルウェアによって発生する多様な通信に対応できるテンプレートの生成が要求される．しかし，過度に多様性を確保した場合，正常な通信を悪性と判定する誤検知が多発する．感染者を検知した後に発生するフォレンジックや対応処理のコストを考慮すると，誤検知は必要最低限に抑制する必要がある．

そこで本稿では，マルウェアや C&C サーバなどの攻撃基盤の特性に着目し，感染後の通信のうち可変的な特徴と不変的な特徴を特定して通信内容をテンプレート化する手法を提案する．実ネットワークを用いた評価では，提案手法は，従来手法と比較して誤検知数を最大で約 70%削減しつつ，従来手法より高い精度でマルウェア感染後の端末による通信を検知できた．

2 関連研究

マルウェアの動的解析で得られる通信データ（以後，マルウェア通信データと呼称する）に基づいて，感染端末検知用のテンプレートを生成する手法が検討されている [1, 2]．感染端末は，疎通確認や解析妨害のために正規サイトへも通信する．このため，これらの手法のように感染端末の通信のみからテンプレートを生成すると，正規サイトへ通信する正常な端末を感染端末と誤検知する可能性が高い．

この問題を解決する手法として，ExecScent [3] が検討されている．ExecScent は関連研究 [1] を発展させ，URL だけでなく HTTP リクエスト全体に着目して正規表現を含むテンプレートを生成する．テンプレートはマルウェア通信データをクラスタリングすることで生成され，URL だけでなく User Agent や宛先 IP アドレスに関連する情報が記述される．この結果，URL だけに着目したテンプレートでは誤検知となる HTTP リクエストを正しく判別することができる．さらに，生成したテンプレートを防御対象のネットワークにおける通信データ（以後，ネットワーク通信データと呼称する）とマッチングする際に，テンプレートとの類似性の高さに加えて，ネットワークでの希少性の高さを考慮して判定することで，より誤検知を抑制する．ここで希少性とは，防御対象ネットワーク内での珍しさを示す指標であり，ある通信の希少性が高いということは，当該通信が防御対象ネットワークでは通常観測されにくいことを意味する．

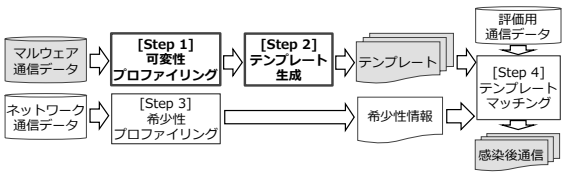

図 1: 提案システム概要

2.1 従来手法の課題

前述の従来手法 [3] では，感染端末の HTTP リクエストに含まれる文字列をデータ種別とデータ長で構成される正規表現に置き換えることで，リクエストを抽象化しテンプレートを生成する．これにより，多様なリクエストを少数のテンプレートに集約している．しかし，この手法では，抽象度を高めた弊害として，意図しない正規の HTTP リクエストがテンプレートにマッチし，誤検知を引き起こす可能性がある．例えば，テンプレート生成元の HTTP リクエストの構造と，正規の HTTP リクエストの構造が本質的には全く異なるものであっても，文字列として比較すると同じデータ種別とデータ長の正規表現に丸められてしまう場合が存在し，誤検知の原因となる．このような誤検知を回避するためには，マルウェアや C&C サーバの通信において不変的な特徴は正規表現化せず，可変的な特徴のみを正規表現に置き換える必要がある．

3 提案手法

3.1 概要

本稿では，感染端末の通信を検知するための通信プロファイル技術を提案する．提案システムの概要を図 1 に示す．本システムは，今回の提案である Step 1, 2 と，従来手法 [3] と同様の手順をとる Step 3, 4 で構成される．

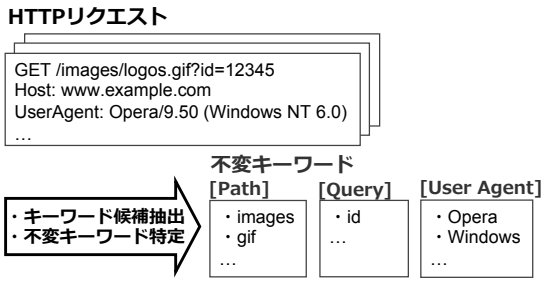

図 2: 可変性プロファイリング

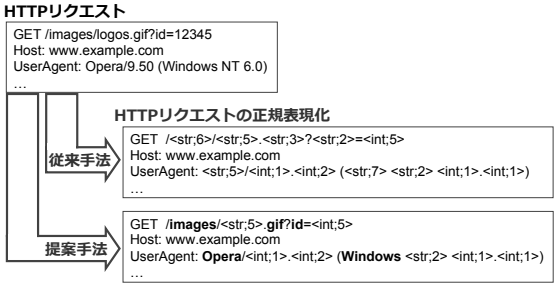

図 3: HTTP リクエストの正規表現化

表 1: 正規表現パターン

データ種別	正規表現パターン
英文字	<str; データ長>
整数	<int; データ長>
16 進数	<hex; データ長>
Base64	<base64; データ長>

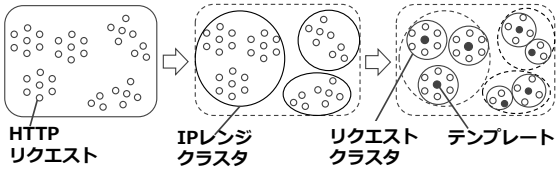

図 4: クラスタリングによるテンプレート生成

3.2 Step 1: 可変性プロファイリング

誤検知を回避しつつ感染端末を検知するために，マルウェア通信データに含まれる HTTP リクエストの構成要素（URL パスや URL クエリ，および User Agent）の可変性をプロファイリングする．図 2 に示すように，まず，マルウェア通信データに含まれる HTTP リクエストの各構成要素から 2 文字以上の連続する英文字列をキーワード候補として抽出する．次に，各要素のキーワード候補から攻撃者が同一の攻撃基盤（マルウェアや C&C サーバ）を利用することに起因して不変と推定できる文字列を不変キーワードとして特定する．具体的には，異なる複数種類のマルウェア検体を動的解析した際に共通的に利用された実績のあるキーワードを不変キーワードとして特定する．なお，マルウェア動的解析機能としては，インターネットへ接続した環境でマルウェアを動的解析する BotnetWatcher [4] を利用する．

3.3 Step 2: テンプレート生成

3.3.1 HTTP リクエストの正規表現化

マルウェア通信データに含まれる HTTP リクエストの特徴からテンプレートを生成する．このとき，HTTP リクエストの特徴を集約してテンプレート生成を効率化するために，HTTP リクエストに含まれる各構成要素の文字列を，表 1 に示す正規表現パターンに置換する．正規表現への置換例を図 3 に示す．従来手法 [3] では，HTTP リクエストのうち上記に該当する文字列をすべて正規表現に置換する．一方，提案手法では，当該文字列が Step 1 の可変性プロファイリングで特定した不変キーワードと一致する場合，正規表現に置換しない．Step 1 を利用しない従来手法の方が多くの文字列を同一の正規表現パターンに置換するため集約効率が高い．しかし，この場合，正規の通信も同一の正規表現に該当する可能性が高くなり，後述の Step 4 における誤検知が増加する．一方，提案手法の場合，従来手法と比較して集約効率は低くなるものの，感染端末に起因した HTTP リクエストの特徴を正確に表現できる．

3.3.2 クラスタリング

テンプレート生成の更なる効率化のため，従来手法では，正規表現化済の HTTP リクエストのうち類似するものを集約してテンプレートを生成する．このため，提案手法でも同様の処理を実施する．具体的には，上記で正規表現化した HTTP リクエストに対し，ニ段階のクラスタリングを実施することでテンプレートを生成する．クラスタリングの概念を図 4 に示す．ここで，テンプレートは感染端末に

よって発生する特徴的な HTTP リクエストを集約した情報に相当する．

第一段階のクラスタリングでは，宛先 IP アドレスが同一/24 ネットワークに存在する HTTP リクエストをグループ化して IP レンジクラスタを生成する．第ニ段階のクラスタリングでは，各 IP レンジクラスタに含まれる HTTP リクエストに階層的クラスタリングを適用する．階層的クラスタリングでは，HTTP リクエスト間の類似度をあらかじめ定めた定義に基づいて計算し，類似度の高いリクエストを逐次的に併合してデンドログラム（階層構造を示す樹形図）を生成する．生成したデンドログラムを経験的に決定する類似度の基準点で切断することで複数の HTTP リクエストをグループ化し，リクエストクラスタを生成する．生成した各リクエストクラスタ内の HTTP リクエストのうち，他の HTTP リクエストとの類似度の総和が最大になるものをテンプレートとして抽出する．

ここで，HTTP リクエスト a と b の類似度 $Sim(a,b)$ を以下の式で定義する．

$$Sim(a,b) = \frac{1}{k} \cdot \sum_{k} s_k(a,b) \tag{1}$$

なお，$s_k(1 \leq k \leq 5)$ は HTTP リクエストに含まれる各要素の類似度を算出する関数である．s_1 を a と b の URL パスの類似度と定義し，標準化編集距離を用いて算出する．s_2 を a と b の URL クエリに含まれるパラメータの組合せの類似度と定義し，Jaccard 係数を用いて算出する．s_3 を a と b の URL クエリの値の類似度と定義し，データ種別とデータ長の一致割合を類似度とする．s_4 を a と b の User Agent の類似度と定義し，標準化編集距離を用いて算出する．s_5 を a と b の宛先ネットワークの類似度と定義し，a と b の宛先 IP アドレスが同一/24 ネットワークの場合 $s_5 = 1$ とし，それ以外の場合 $s_5 = 0$ とする．

3.4 Step 3: 希少性プロファイリング

Step 2 で生成したテンプレートを防御対象ネットワークで適用する際の誤検知を可能な限り削減するため，当該ネットワークでの希少性情報を生成する．ここで，防御対象ネットワーク内では，感染端末による HTTP リクエストが正常な端末のそれと比較して非常に少ないと仮定する．この場合，当該ネットワーク内で希少性が高い要素が新たに出現した際に，その要素は感染端末によって出現した可能性が高いとみなすことができる．各要素の希少性は防御対象ネットワークごとに異なることから，防御対象ネットワークごとに希少性情報を生成する必要がある．このため，Step 2 で生成したテンプレートに含まれる各要素の文字列の希少性をネットワーク通信データから算出し，当該ネットワークの希少性情報を生成する．本稿で希少性情報の生成対象とする要素は，URL パス，URL クエリ，User Agent，宛先 IP アドレス，宛先 FQDN である．従来手法 [3] ではこれらに加え HTTP リクエストヘッダフィールドの組合せも希少性の算出対象とするが，本稿では利用するデータセットの制約から上記の 5 つの要素のみを対象とする．ある要素 k の文字列の希少性 σ_k は，以下の式で算出する．

$$\sigma_k = 1 - \frac{n}{\max_k n_k} \tag{2}$$

ここで，n は当該文字列を含む HTTP リクエストを生成した送信元ホスト数，$\max_k n_k$ はある要素 k を含む HTTP リクエストを生成した送信元ホスト数のうちの最大値とする．

3.5 Step 4: テンプレートマッチング

Step 3 までで生成したテンプレートと希少性情報を用いて，評価用通信データの中から感染端末によって発生した HTTP リクエストを検知する．具体的には，まず，評価用通信データ含まれる各 HTTP リクエスト r と各テンプレート t の一致性を以下に定義するスコア $S(r,t)$ で算出する．

$$S(r,t) = \frac{\sum_k \omega_k(s_k,\sigma_k) \cdot s_k(r_k,t_k)}{\sum_k \omega_k(s_k,\sigma_k)} \cdot \sigma_d \tag{3}$$

ここで，$s_k(1 \leq k \leq 5)$ は 3.3.2 節で定義した類似度と同一のものとする．また，σ_k は希少性情報で定義した要素 k の希少性であり，σ_d は宛先 FQDN の希少性である．なお，ω_k は s_k と σ_k に応じた重みを算出する関数であり，以下の式で定義する．

$$\omega_k(s_k,\sigma_k) = \omega' \cdot (1 + \frac{1}{(2 - s_k \cdot \sigma_k)^n}) \tag{4}$$

ここで，ω' と n は固定値のパラメータであり，評価の際に経験的に決定する．このスコア $S(r,t)$ は，HTTP リクエスト r とテンプレート t との類似性が高く，かつ r の対象ネットワークでの希少性が高い

場合，高い数値となる．スコア $S(r,t)$ が事前に設定したしきい値以上の場合，その HTTP リクエスト r を感染端末によって生成されたものとみなし，送信元を感染端末として検知する．

4 提案手法の評価

4.1 概要

提案手法の評価を，実ネットワークの通信データを用いて実施する．本評価は，提案手法のフィージビリティ評価という観点と，提案手法と従来手法 [3] の検知性能の比較という観点で実施する．前者は，可変性プロファイリングの出力結果と，従来方式と比較した際のテンプレート数の増減から評価する．後者は，提案手法と従来手法を同一の評価用通信データに適用した際の検知性能から評価する．

4.2 データセット

本評価では，提案手法の実ネットワーク上での効果を評価するために，実際のマルウェア通信データやネットワーク通信データを評価用通信データとして利用する．ただし，この場合，実ネットワークに感染端末が存在し，かつ感染端末の通信を正解データとして事前に識別しておく必要がある．このため，本評価では，マルウェア通信データとネットワーク通信データを各々二分割し，一方（以後，訓練用データと呼称する）を既知のマルウェア通信データやネットワーク通信データとしてテンプレートや希少性情報の生成に使用し，もう一方（以後，評価用データ）を未知の感染端末や正常な端末の通信データとして利用して評価を行った．

マルウェア通信データは，2011 年 8 月から 2013 年 8 月までの間にハニーポットで収集したマルウェア検体を BotnetWatcher で動的解析することで取得した．マルウェア通信データに含まれるマルウェア検体数，HTTP リクエスト数を表 2 に示す．表 2 のうち訓練用データは重複のない 2,507 個のマルウェア検体の HTTP リクエストであり，評価用データは重複のない 5,058 個の検体の HTTP リクエストである．なお，訓練用データと評価用データのマルウェア検体間に重複はない．さらに，MWS Datasets 2014 [5] で提供されている D3M データセットに含まれるマルウェア通信データ（フルキャプチャデータ）から HTTP リクエストを抽出したものを評価用データとして利用した．データセットの内訳を表 3 に示す．

表 2: マルウェア通信データ

	収集期間	検体数	リクエスト数
訓練用	2011/08/27〜2012/12/31	2,507	598,534
評価用	2013/01/01〜2013/08/20	5,058	188,797

表 3: D3M マルウェア通信データ（評価用）

データセット	データ収集日	検体数	リクエスト数
D3M 2013	2012/08/02	1	2
D3M 2013	2012/10/02	2	13
D3M 2013	2013/02/26	6	16
D3M 2014	2013/04/12	2	86
D3M 2014	2013/08/30	2	1,520
D3M 2014	2014/04/10	2	16

表 4: ネットワーク通信データ

	収集期間	SrcIP 数	リクエスト数
訓練用	2012/12/01〜2012/12/31	5,261	95,438,564
評価用	2013/01/01〜2013/08/20	8,055	723,903,639

ネットワーク通信データは，2012 年 12 月から 2013 年 8 月までの間に実ネットワーク環境で収集した．ネットワーク通信データに含まれる送信元 IP アドレス数，HTTP リクエスト数を表 4 に示す．なお，初期 1ヵ月で取得したネットワーク通信データを訓練用データとし，その後取得したネットワーク通信データを評価用データとした．

このようなデータセットを生成することで，あるネットワーク環境に感染端末が存在するという仮定のもと，その感染端末による HTTP リクエストを正しく検知できるかを評価することができる．

4.3 可変性プロファイリングのフィージビリティ評価

本評価では，可変性プロファイリングを実施した際の結果を示す．具体的には，表 2 に記載された訓練用のマルウェア通信データを用いて可変性プロファイリングを実施し，HTTP リクエストに含まれる URL パスや URL クエリ，および User Agent から

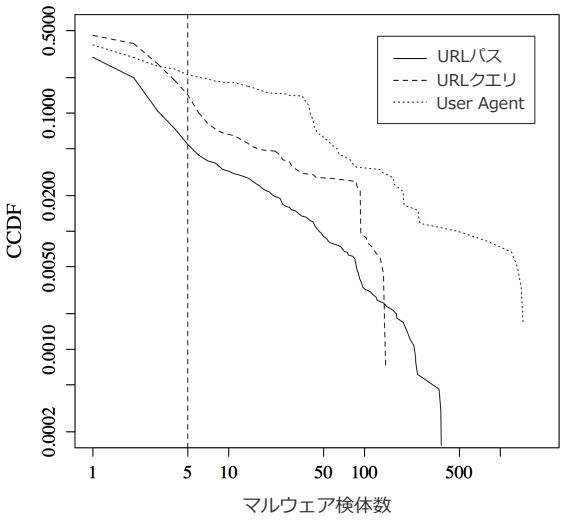

図 5: 各キーワード候補を利用する検体数

抽出した各キーワード候補と，各キーワードを利用するマルウェア検体数を調査する．あるキーワードを利用するマルウェア検体数が多いほど，当該キーワードがより多くの異なるマルウェアによって再利用される不変的なキーワードであるとみなすことができる．ここで，URL パスと URL クエリ，および User Agent の各々のキーワード候補と，各キーワードを利用していたマルウェア検体数の関係を，相補累積分布関数（Complementary Cumulative Distribution Function, CCDF）を用いて図 5 に示す．なお，CCDF はグラフ横軸の各検体数以上の検体に利用されたキーワードが全体に占める割合を示すものである．

図 5 から，感染端末によって生成される HTTP リクエストには，複数のマルウェア検体によって再利用されている不変的なキーワードが存在することが明らかになった．ここで，検体数の基準を設定し，基準以上の検体数に共通的に利用された実績のあるキーワードを不変キーワードとして特定する．なお，今回は実験的に検体数の基準を 5 検体と設定して以後の評価を行う．検体数の基準については 4.5.1 節で議論する．表 5 に Step 1 で出力した各キーワード候補数と，不変キーワードとして特定されたキーワード数を示す．

また，表 6 に，不変キーワードを利用する提案手法と，不変キーワードを利用しない従来手法で生成されるテンプレート数を示す．3.3.1 節に示した通り，提案手法は，不変キーワードを利用して正規表現化するため，テンプレート数が多くなる．しかし，増加数は約 4%程度であるため，計算量的には無視できると考えられる．

表 5: キーワード候補数と不変キーワード数

	URL パス	URL クエリ	User Agent
キーワード候補数	6,521	1,365	601
不変キーワード数	483	259	142

表 6: テンプレート数の変化

	入力リクエスト数	出力テンプレート数
従来手法	598,534	2,580
提案手法	598,534	2,683

4.4 検知性能評価

本評価では，同一のデータセットを利用して提案手法と従来手法の検知性能を比較した結果を示す．具体的には，Step 1 で生成した不変キーワードを利用して Step 2〜Step 4 を実施した評価結果と，不変キーワードを利用せずに Step 2〜Step 4 を実施した評価結果の比較を行う．なお，不変キーワードを利用しない場合は，従来手法 [3] と同等の機能を評価することに相当する．

表 7 と表 8 に，しきい値を変動させたときの各手法の誤検知数・誤検知率と検知数・検知率を示す．ここで，しきい値とは $S(r,t)$ に適用されるしきい値である．各手法では，しきい値以上の $S(r,t)$ が算出された HTTP リクエストを，感染端末によって発生した HTTP リクエストとして検知する．なお，誤検知率は，正常な端末による HTTP リクエストを誤って検知した率を示し，検知率は，感染端末によって発生する HTTP リクエストを検知した率を示している．

表 7 より各手法の誤検知率を比較すると，すべての場合で提案手法が優れており，特にしきい値が 0.75 の場合，誤検知数を約 70%削減できている．これは，提案手法における不変キーワードの効果が想定通り得られており，不変キーワードを用いた抽象化を行うことで，従来手法よりも誤検知となる場合を抑制することができていることを示している．な

表 7: 誤検知数・誤検知率（ネットワーク通信データ）

	しきい値	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
従来手法	誤検知数	211,824,081	179,275,589	101,216,108	28,125,055	3,313,211	589,616	316,777	79,510
	誤検知率	29.26%	24.77%	13.98%	3.89%	0.46%	0.08%	0.04%	0.01%
提案手法	誤検知数	181,281,597	106,558,990	36,345,439	8,558,301	1,435,060	444,982	209,441	51,740
	誤検知率	25.04%	14.72%	5.02%	1.18%	0.20%	0.06%	0.03%	0.01%

表 8: 検知数・検知率（マルウェア通信データ）

	しきい値	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
従来手法	検知数	149,395	142,831	87,813	82,285	69,557	50,604	8,789	8,754
	検知率	79.13%	75.65%	46.51%	43.58%	36.84%	26.80%	4.66%	4.64%
提案手法	検知数	143,585	139,841	88,126	79,930	69,264	51,956	8,972	8,857
	検知率	76.05%	74.07%	46.88%	42.34%	36.69%	27.52%	4.75%	4.69%

表 9: 検知数・検知率（D3M マルウェア通信データ）

	しきい値	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
従来手法	検知数	1,617	1,617	171	162	160	27	22	22
	検知率	97.82%	97.82%	10.34%	9.80%	9.68%	1.63%	1.33%	1.33%
提案手法	検知数	1,618	1,615	167	166	162	31	22	22
	検知率	97.88%	97.70%	10.10%	10.04%	9.80%	1.88%	1.33%	1.33%

お，誤検知率がしきい値が 0.95 の場合に約 0.01%と非常に低い一方で，誤検知数が最低でも数万件存在する原因について 4.5.2 節で議論する．

表 8 より各手法の検知率を比較すると，手法間の差が小さいことと，しきい値が 0.70 の場合を除き，しきい値が低い場合（〜0.80）は従来手法の検知率が提案手法を上回り，しきい値が高い場合（0.85〜）は提案手法の検知率が従来手法を上回る傾向があることがわかる．後者の現象は，希少性を考慮したスコア計算に起因するものである．従来手法では，不変キーワードを利用せずに HTTP リクエストを抽象化した状態でテンプレートを生成し，テンプレートにおける各要素の希少性を算出する．このため，従来手法は提案手法と比較して抽象度が高くなる一方で希少性が低くなり，結果としてスコアが低くなる．したがって，特にしきい値が高い領域では，提案手法の検知率が従来手法を上回る．

以上の結果より，提案手法は従来手法よりも誤検知率を抑制しながら，検知率を維持できたことがわかる．

ここで，D3M データセットを利用して検知数・検知率を評価した結果を表 9 に示す．表 8 と同様に，一部のしきい値の場合（0.65, 0.70）以外では，提案手法が従来手法を上回った．また，本評価では，両手法の検知率において，しきい値の 0.65 と 0.70 の間に大きな差が生じた．これは，表 3 に示すように本データセットにおいて支配的だった 2013/08/30 収集分の HTTP リクエスト（1,520 件）の内容に起因している．具体的には，当該日の HTTP リクエストの大部分のスコアが一様に約 0.69 となっているために，しきい値を 0.70 以上にすると検知率が極端に減少した．この原因を 4.5.3 節で詳細に解説する．

4.5 議論

4.5.1 可変性プロファイリングの検体数基準

今回は提案手法の Step 1 の可変性プロファイリングにおける検体数の基準を 5 検体と設定して評価を実施した．検体数の基準は不変キーワード数に影響する．具体的には，図 5 に示した通り基準を増加させると不変キーワード数は減少し，基準を減少させると不変キーワード数は増加する．不変キーワード数が増加すると検知数・誤検知数ともに減少し，提案手法の検知性能に直接的に影響するため，最適な検体数基準の決定は今後の検討課題である．

4.5.2 ネットワーク通信データにおける誤検知数・誤検知率

表 7 に示した提案手法と従来手法の誤検知率は，しきい値が 0.95 の場合に約 0.01%と，非常に低い．一方，誤検知数は，評価用リクエストが表 4 の通り 7 億件以上となっているため，誤検知率が 0.01%だとしても数万件となる．誤検知後に発生する対応処理に必要となるコストを考慮すると，誤検知数はより少ないことが望まれる．

従来は Alexa [7] に掲載されている上位サイトや通信先のレピュテーション [4] を考慮した分析により，更なる誤検知数の削減を図るが，本稿では手法自体の評価を行うために，本分析処理を除外している．なお，提案手法を用いた場合，表 7 に示す通り，本分析の対象となる誤検知数は最小でも 27,000 件以上（しきい値が 0.95 の場合）削減できている．このため，提案手法を実運用することで，誤検知数の削減だけでなく，分析コストの削減効果も期待できる．

4.5.3 D3M マルウェア通信データにおける検知数・検知率

D3M データセットの 2013/08/30 に収録されているマルウェア通信データは，PUSHDO[6] と呼ばれるマルウェア検体に感染した際に発生する HTTP リクエストである．PUSHDO は，C&C サーバの隠ぺいのために，C&C サーバと多数の正規ドメインに対して同様の内容を保有する HTTP リクエストを送付する．本評価では，当該 HTTP リクエストは提案手法および従来手法で生成したテンプレートとの類似性が高いものの，URL パスや URL クエリ，および User Agent のそれぞれの希少性が低いために，スコアが約 0.69 と相対的に低くなった．この結果，しきい値 0.70 以上の場合の検知率が極端に低くなった．

PUSHDO への感染時には同一の HTTP リクエストを多数のドメイン宛に高い送信レートで送付するため，感染端末自体を発見するのは容易である．しかし，本稿での提案手法および従来手法を単独で利用するだけでは同検体への感染を検知することが困難である場合がある．このため，両手法とも，レート検知等の従来手法との併用することで，相乗効果が期待できると考えられる．

5 おわりに

本稿では，マルウェアや C&C サーバなどの攻撃基盤の特性に着目し，感染後の通信のうち可変的な特徴と不変的な特徴を特定して通信内容をテンプレート化する手法の設計と評価を行った．マルウェアの動的解析や実ネットワークの通信データを用いて提案手法を評価した結果，提案手法は，従来手法と比較して誤検知数を最大で約 70%削減しつつ，従来手法より高い精度でマルウェア感染後の端末による通信を検知できた．さらに，提案手法における今後の検討課題や手法自体の制約条件を明らかにした．

参考文献

[1] R. Perdisci, W. Lee, and N. Feamster, “Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces,” Proc. USENIX NSDI, p.26, Apr. 2010.

[2] M. Z. Rafique, and Juan Caballero, “FIRMA: Malware Clustering and Network Signature Generation with Mixed Network Behaviors,” Proc. RAID, Oct. 2013.

[3] T. Nelms, R. Perdisci, and M. Ahamad, “ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates,” Proc. USENIX Security, pp.589–604, Aug. 2013.

[4] K. Aoki, T. Yagi, M. Iwamura, and M. Itoh, “Controlling Malware HTTP Communications in Dynamic Analysis System Using Search Engine,” 2011 Third International Workshop on Cyberspace Safety and Security (CSS), pp.1–6, Sep. 2011.

[5] 秋山満昭, 神薗雅紀, 松木隆宏, 畑田充弘, “マルウェア対策のための研究用データセット〜MWS Datasets 2014〜,” 情報処理学会 研究報告コンピュータセキュリティ (CSEC), Vol. 2014-CSEC-66, No. 19, pp. 1–7, 2014.

[6] M. Antonakakis, B. Stone-Gross, J. Demar, K. Stevens, and D. Dagon. “Unveiling The Latest Variant of Pushdo Mv20: A case study on the new Pushdo-DGA,” Technical Report, Damballa Inc., Sep. 2012.

[7] Alexa Top Sites, `http://www.alexa.com/topsites/`