リクエスト間隔とレスポンスのボディサイズに基づくマルウェア感染由来のHTTPトラフィック検知

小川 秀貴[1]　山口 由紀子[2]　嶋田 創[2]　高倉 弘喜[3]　秋山 満昭[4]　八木 毅[4]

概要：昨今のサイバー攻撃は巧妙化しており，マルウェア感染を未然に防ぐことが困難となっている．したがって早期のマルウェア感染検知技術が重要となっている．昨今のマルウェアはファイアウォールやプロキシでの検知を回避するために，C&C 通信に業務等で使われている HTTP を使用したものが多く，検知が困難である．そこで本研究では，特に HTTP トラフィックを対象としたアノマリ型の検知手法を提案する．提案手法では HTTP の各通信先ごとにリクエスト送信間隔とレスポンスのボディサイズから特徴量を抽出し，SVM を用いてマルウェア感染由来かどうかの判定を行う．

キーワード：マルウェア，感染検知，機械学習，トラフィック解析

Malware Originated HTTP Traffic Detection based on Request Interval and Response Body Size

Hideki Ogawa[1]　Yukiko Yamaguchi[2]　Hajime Shimada[2]　Hiroki Takakura[3]　Mitsuaki Akiyama[4]
Takeshi Yagi[4]

Abstract: Recent cyber attacks are sophisticated so that it is difficult to prevent malware infection. Therefore, early malware infection detection becomes more important. Moreover, latest malware utilizes HTTP which is widely used on business for avoiding detection by firewalls and proxies. It further makes malware infection detection harder with typical traffic analysis. In this study, we propose an anomaly detection method for malware originated HTTP traffic. In proposal, we judge HTTP traffic by SVM with utilizing newly extracted features such as HTTP request interval and response body size.

Keywords: malware, infection detection, machine learning, traffic analysis

1. はじめに

昨今，標的型攻撃に代表されるようにサイバー攻撃は巧妙化している．なぜなら，以前は攻撃者が自身の技術を披露するような愉快犯が多かったのに対して，昨今は，クレジットカード情報や銀行の決済情報，特定の組織や団体の機密情報を窃取し，金銭を得るようになったからである．特に，機密情報の中にはお金では手に入れることができないものがあり，そのような機密情報を取得するために，高度なサイバー攻撃をしかける組織が存在する．そのためには，これまで行われてきたばらまき型の攻撃ではなく，特定の組織やターゲットに特化したサイバー攻撃手法が必要となる．そこで標的型攻撃と呼ばれる手法が登場した．独立行政法人情報処理推進機構 (IPA) によると高度標的型攻撃は主に，計画立案，攻撃準備，初期侵入，基盤構築，内部侵入・調査，目的遂行，再侵入の７つの段階で構成されている [1]．計画立案では，攻撃対象を決定し，調査を行う．攻撃準備では，C&C サーバなどの攻撃の準備を行う．

[1] 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University
[2] 名古屋大学情報基盤センター
Information Technology Center, Nagoya University
[3] 国立情報学研究所
National Institute of Informatics
[4] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories

初期侵入では，標的型メールや水飲み場型攻撃などによって対象のネットワークに侵入する．基盤構築では，C&C サーバとのバックドア通信を構築し，感染端末が指令を受け取れるようにする．内部侵入・調査では，ほかの端末への侵入などによって対象となるネットワークの情報調査を行う．目的遂行では，情報窃取やシステムの破壊を行う．再侵入では，バックドアから再び侵入をして，内部調査や目的遂行を行う．

中でも初期侵入は特に巧妙化しており，ファイアウォールやアンチウイルスソフトウェア，侵入検知システムといったような従来のサイバー攻撃対策を施しても完全に防ぐことが困難となっている．そこでマルウェア感染による初期侵入を許すことは避けられないと考えられる．しかし，攻撃者のマルウェア感染による初期侵入を許したとしても内部情報の窃取といったような目的遂行を防ぐ必要があるため，マルウェア感染を早期に検知する必要がある．一般的に標的型攻撃では，目的となる情報を窃取するために，最初に感染した端末を踏み台にしてネットワーク内を探索し，さまざまな情報を収集するため，高度標的型攻撃の 7 段階における内部侵入・調査の段階において要する時間は他の段階と比べて大きいと考えられる [2]．したがって，目的遂行の前段階である，内部侵入・調査の段階でマルウェア感染を検知できる余地があり，目的遂行を防ぐという観点で重要である．

マルウェア感染検知技術としては主にシグネチャ型の検知手法とアノマリ型の検知手法の 2 つがある．前者のシグネチャ型の検知手法ではシグネチャに一致するようなマルウェアを検知する手法のことである．シグネチャに一致すれば検知できるため，誤検知は低く抑えられるが，シグネチャを回避するようなマルウェアには対処できない．後者のアノマリ型の検知手法ではトラフィックデータやログデータなどから振る舞いを学習させることで，検知する手法である．この手法はシグネチャ型の検知を回避するような亜種のマルウェアや新種のマルウェアを検知できる可能性が高いという利点がある反面，誤検知が多いという欠点も存在する．したがって，誤検知を抑えたアノマリ型の検知手法が求められてきている．

従来のマルウェアは C&C サーバとの通信に IRC プロトコルや独自プロトコルを利用するものが多く，アノマリ検知においてこのような特徴量をもとに検出をすることは比較的容易である．また，ファイアウォールやプロキシにおいて遮断することも容易であった．しかしながら近年のマルウェアは，検知を免れるために，頻繁に使用されるプロトコルを通信に用いるようになってきている．例えば，Emdivi などで利用される HTTP は業務等で多用されるため，ファイアウォールやプロキシを設置しての検知や，汎用的なアノマリ検知アルゴリズムでの検知が困難となってきている [3]．したがって，HTTP で通信するマルウェアに特化したアノマリ型のマルウェア感染検知手法が求められる．この問題に対し，我々は HTTP リクエスト間隔とレスポンスのボディサイズの違いをもとに検知を実施できないかと考えた．これは，マルウェア感染時の通信では C&C サーバと定期的にコマンドのやり取りを行うものが主体となるのに対して，近年の正常通信では Web コンテンツのリッチ化により，Ajax 利用コンテンツの多数のセッションの同時利用や動画コンテンツのような大容量コンテンツの利用が多くなっていると考えられるからである．そこで提案手法では，トラフィックデータから取得した HTTP リクエストおよびレスポンス情報から特徴量を抽出し，Support Vector Machine（SVM）によって 2 値判定を行うことでマルウェア感染由来の HTTP トラフィック検知を行う．

2. 先行研究

鮫島らは，トラフィックデータから単位時間であるタイムスロット単位で特徴量を抽出し，クラスタリングの結果から得られた状態遷移情報を元にマルウェア感染検知を行った [4]．従来ではセッション単位やフロー単位での検知が主であったが，時間情報を考慮していないという問題点があった．そこでタイムスロット単位で特徴量を抽出し，単一のクラスタのみではなく，クラスタ遷移を考慮することで解決した．また，感染時の挙動に基づき詳細に分類し，感染時の通信パターンを作成することで検知制度が大幅に向上することを確かめた．この手法では特徴量としてタイムスロットごとにパケットサイズの最小，SYN パケット数，TCP パケットに対する SYN パケットの割合，ACK パケット数といったようなトラフィックデータのヘッダ情報から得られる特徴量を元に検知を行っており，プライバシー保護の観点からも有効な手法である．しかし，現状の特徴量では，昨今の HTTP 通信を模倣するようなマルウェアに対応できているとはいえない．さらに，タイムスロット単位という特性上，たとえマルウェア感染によって通信が発生していても，正常通信が多量に発生していれば，正常通信に埋もれて検知されにくくなることが考えられる．また，マルウェア感染の疑いのあるタイムスロットを検知しても，そこからさらに感染の疑いのあるホストや怪しい通信先を特定する必要がある．

大月らはトラフィックデータのペイロード情報から HTTP の GET メソッドや POST メソッドといったような文字列の出現頻度や文字コードの出現頻度，HTTP リクエスト長を特徴量としてクラスタリングすることでマルウェア感染検知を行った [5]．この手法では HTTP 通信を考慮しているため，昨今の HTTP 通信を模倣したマルウェアにも適用可能な手法といえる．しかし，HTTP ヘッダのみではなく HTTP のボディ部分を参照しているため，プライバシー保護の観点から HTTP のボディ部分を参照しない手法が求められる．また，HTTPS 通信ではないにして

も，マルウェアが C&C サーバと通信する際のコマンドは暗号化されて HTTP のボディ部分やヘッダ部分に埋め込まれていることが多く，ボディ部分を参照して文字コードの出現頻度から特徴量をとっている場合こういったケースに対応できず，検知できないという問題点が挙げられる．

鳥居らは，昨今の標的型攻撃でも用いられている RAT を検知する手法を提案した [6]．プロキシを回避するような HTTP 通信をする RAT を検知するために，CONNECT メソッドが指定されたリクエストのみを監視対象としている．具体的には，プロトコル違反があるかどうか，User-Agent が偽装されているかどうか，パケットサイズといった情報が不審な特徴を持つ通信から得られたとしている．RAT が行う通信にはパケットサイズが規則的なものが多いという有効な特徴量を発見することができているが，一般的に正常通信と比べて C&C サーバとは規則的な間隔で通信するものもあるため，その点を考慮していない点が課題である．

3. 提案手法

3.1 概要

昨今の多くのマルウェアにおいて HTTP を模倣した通信によって C&C サーバとやり取りが行われている一方，マルウェア感染検知手法においては HTTP 通信を考慮しないものが一般的である．また，HTTP 通信を考慮している検知手法においても，HTTP のボディ部分のみを参照しているものが多く，プライバシー保護の点から好ましくないという問題点やボディ部分を暗号化するマルウェアの通信に対応できないという問題点もある．また，C&C サーバと通信するマルウェアはコマンドのやり取りを規則的な時間間隔で実施するものがあり，また，ボディサイズが大きく変化しないものが多いと考えられる．ゆえに，正常通信とマルウェア感染時に発生する HTTP 通信ではリクエスト間隔やレスポンスのボディサイズの分布が異なると考えられるが，これまでの手法では考慮されていないことが多かった．

それに対して，本稿では正常通信とマルウェア感染時の HTTP のリクエスト間隔やレスポンスのボディサイズの分布が異なるという観点から，HTTP のリクエスト間隔とレスポンスのボディサイズから特徴量を抽出して SVM にて判定することにより，マルウェア感染由来の HTTP トラフィックの検知手法を提案する．また，提案手法では HTTP ボディ部分を参照することなく検知処理を実施可能であるため，他の HTTP 通信を利用する手法と比較してプライバシー保護という点でも優位性がある．

提案手法の流れを図 1 に示す．提案手法では学習およびテストの 2 段階に分かれている．学習段階ではトラフィックデータから，HTTP のリクエストとレスポンスペアの情報を構成し，さらにホストの通信先ごとの組み合わせである通信ホストペアごとにリクエスト/レスポンスペアの分割をする．続いて，特徴ベクトルの抽出および正規化を行って，識別器に学習をさせる．テスト段階も同様にトラフィックデータから HTTP リクエスト/レスポンスペアの情報の構成，通信ホストペアごとの分割，特徴ベクトルの抽出・正規化を行い，学習済みの識別器でマルウェア感染由来の HTTP トラフィックかどうかの判定をする．以下で各段階の詳細について述べる．

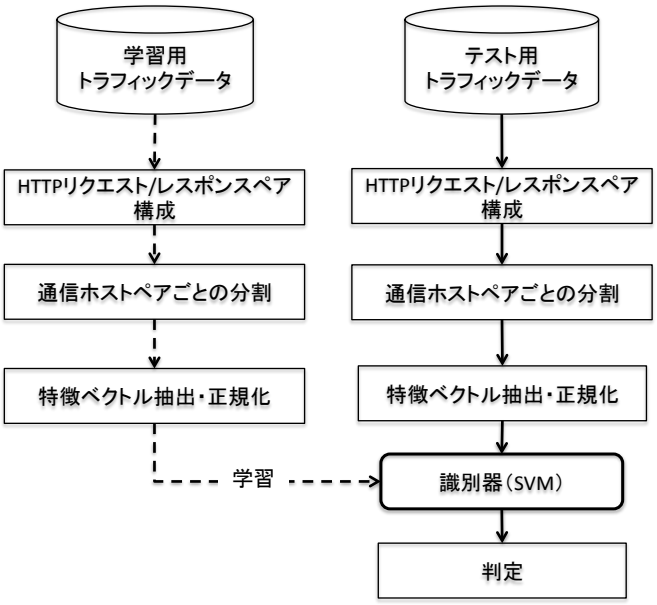

図 1 提案手法の流れ
Fig. 1 Flow of proposed method

3.2 HTTP リクエスト/レスポンスペア構成

まず与えられたトラフィックデータから Bro[*1] を用いて，HTTP トラフィックを抽出してリクエストとそれに対応するレスポンスがペアとなった情報を構成してまとめる．これを HTTP リクエスト/レスポンスペアと呼ぶ．その際に構成する情報は図 2 に示すように，リクエストを出した時間（request time），リクエストの送信元 IP アドレス（src），リクエストを受信する IP アドレス（dst），レスポンスのボディサイズ（response body size）の 4 つである．

3.3 通信ホストペアごとの HTTP リクエスト/レスポンスペア分割

図 3 で示すように，図 2 で抽出した HTTP のリクエスト/レスポンスペアを通信ホストペアごとに分割する．通信ホストペアごとに分割をする理由は，仮にマルウェア感染によってある通信先との間でトラフィックが発生していても，その他の通信先との間で正常なトラフィックが多く発生していた場合，抽出する特徴量が正常トラフィックの影響を大きく受けてしまう可能性が考えられるためである．

[*1] The Bro Network Security Monitor
https://www.bro.org

unix time	src	dst	request or response	body size
100	192.168.7.2	A.B.C.D	request(GET)	0
101	A.B.C.D	192.168.7.2	response	50
200	192.168.7.2	A.B.C.D	request(GET)	0
201	A.B.C.D	192.168.7.2	response	50
300	192.168.7.13	A.B.C.D	request(GET)	0
301	A.B.C.D	192.168.7.13	response	50
400	192.168.7.2	E.F.G.H	request(GET)	0
401	E.F.G.H	192.168.7.2	response	10000
500	192.168.7.13	A.B.C.D	request(GET)	0
501	A.B.C.D	192.168.7.13	response	50
600	192.168.7.2	E.F.G.H	request(GET)	0
601	E.F.G.H	192.168.7.2	response	10000
...	...	...	...	...

request time	src	dst	response body size
100	192.168.7.2	A.B.C.D	50
200	192.168.7.13	A.B.C.D	50
300	192.168.7.2	A.B.C.D	50
400	192.168.7.2	E.F.G.H	10000
500	192.168.7.13	A.B.C.D	50
600	192.168.7.2	E.F.G.H	10000
...	...	...	...

図 2 HTTP リクエスト/レスポンスペア構成
Fig. 2 HTTP request/response pairs construction

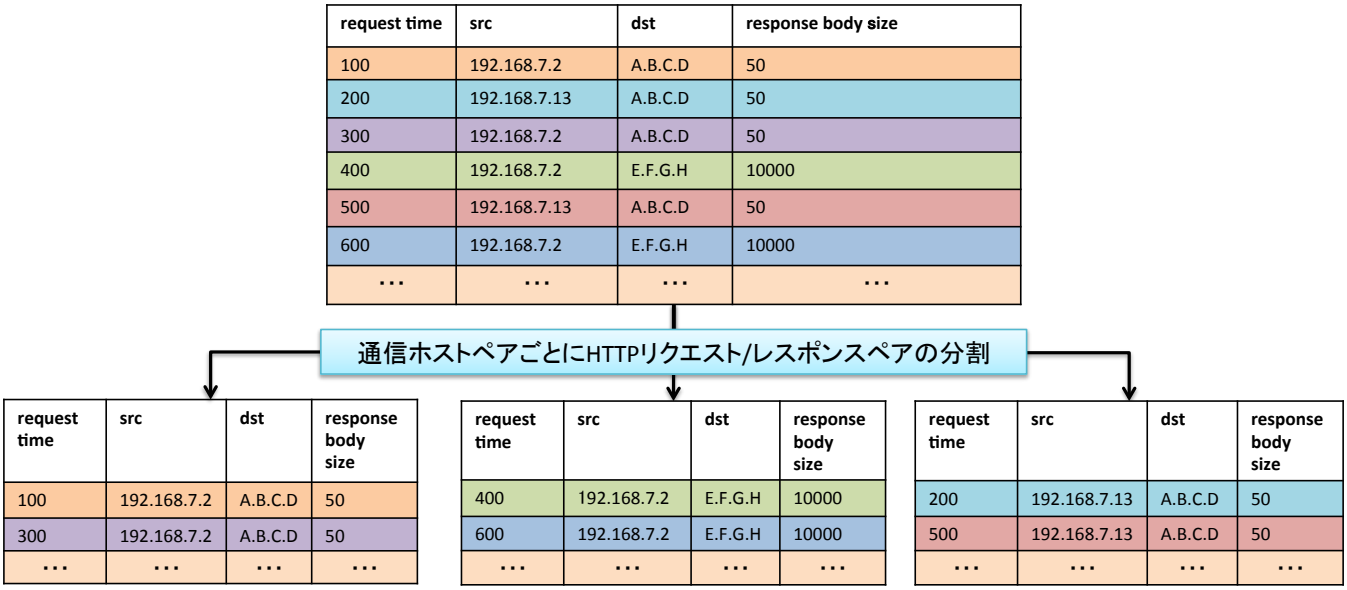

図 3 通信ホストペアごとの HTTP リクエスト/レスポンスペア分割
Fig. 3 Dividing HTTP request/response pairs by communication host pairs

3.4 特徴ベクトル抽出・正規化

3.4.1 特徴ベクトル抽出

通信ホストペアごとに分割した HTTP リクエスト/レスポンスペアから，特徴ベクトルを抽出する流れを図 4 に示す．まずはリクエスト間隔およびレスポンスボディサイズのリストを抽出する．続いて，抽出されたリクエスト間隔およびレスポンスサイズについて，昇順にソートされたリストを作成する．そしてソート済みのリクエスト間隔およびレスポンスのボディサイズのリストから，表 1 に記す 14 個の特徴量を抽出する．

表 1 抽出される特徴量
Table 1 Extracted feature

リクエスト間隔	レスポンスサイズ
最小値	最小値
25 パーセンタイル値	25 パーセンタイル値
中央値	中央値
75 パーセンタイル値	75 パーセンタイル値
最大値	最大値
平均値	平均値
標準偏差	標準偏差

3.4.2 正規化

特徴量間でスケールが大きく異なる場合が起こりうるので，平均 0，分散 1 となるように，データセットから抽出した m 個の特徴量データ $\{f_1, f_2, ..., f_m\}$ の，各特徴ベクトル $f_i = (d_1^i, d_2^i, ..., d_{14}^i)$ の各値について，平均値 (式 1) および標準偏差 (式 2) を求め，個別に正規化 (式 3) を行う．

$$avg_j = \frac{1}{m}\sum_{i=1}^{m} d_j^i \tag{1}$$

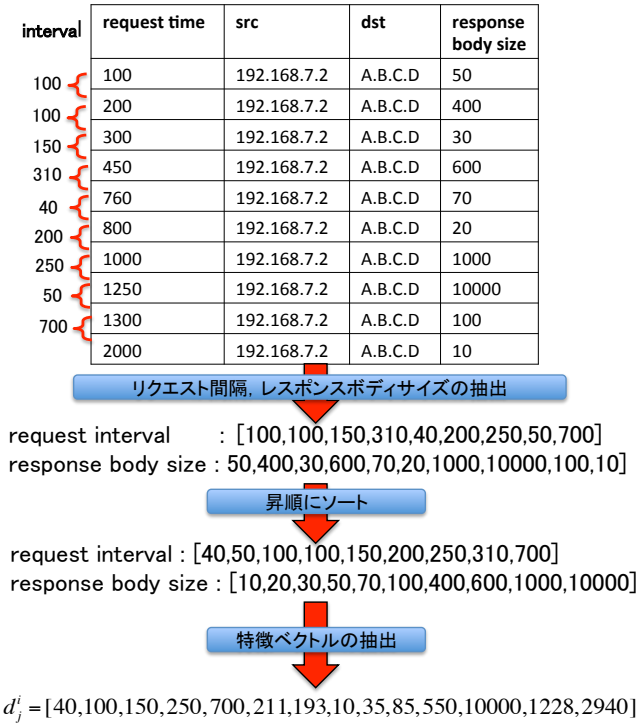

図 4 特徴ベクトル抽出
Fig. 4 Feature vector extraction

$$std_j = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(d_k^i - avg_j)^2} \tag{2}$$

$$normalize(d_j^i) = \frac{d_j^i - avg_j}{std_j} \tag{3}$$

3.5 学習

学習データから抽出された正規化済みの特徴ベクトルを SVM に学習させる．学習の際には SVM のパラメータ C と γ を設定する．

SVM とは教師あり学習によって 2 値判定を行う機械学習手法である．与えられた特徴量を 2 クラスに分類するための分離面を決定することで 2 値判定を行う．図 5 に，2 次元平面において SVM を用いて集合を 2 クラスに分類する様子とそのときの分離面の例を示す．他クラスとの最も近い距離にある点をサポートベクトルと呼び，サポートベクトルと分離面との距離をマージンと呼ぶ．このマージンが最大となるように分離面を決定するマージン最大化を行うことで 2 値分類を行う．SVM には C と γ という 2 つのパラメータが存在する．C は誤分類をどれくらい許容するかを表すパラメータであり，大きければ大きいほど誤分類を許容する．また，γ は SVM の識別面の複雑さを決定するパラメータであり，大きければ大きいほど識別面は複雑になり，汎化能力が低下する．

3.6 判定

テストデータも学習データと同様に 3.2 節から 3.4 節までに示した方法で特徴ベクトル抽出・正規化までを行う．次に，個々の特徴ベクトルを学習済みの SVM に入力して対象の HTTP トラフィックが正常通信に近いものかマルウェア感染由来の通信に近いものかの判定を行う．

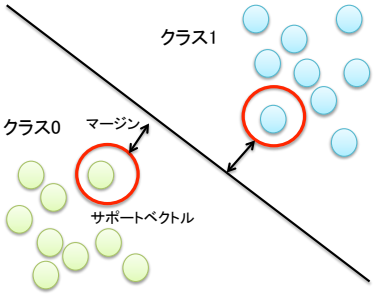

図 5 SVM による分類
Fig. 5 Classification with SVM

4. 実験

4.1 データセット

4.1.1 概要

本研究ではリクエスト間隔とレスポンスボディサイズの各パーセンタイル値をとるため，HTTP リクエストが 5 個以上発生している通信ホストペアから特徴量を抽出してデータセットを作成した．後述する正常および感染トラフィックから抽出した，HTTP リクエストが 1 個以上の通信ホストペアは 9846 個存在し，HTTP リクエストが 5 個以上の通信ホストペアは 2945 個存在した．

なお，本研究のデータセットで使用する正常および感染トラフィックデータは，異なる環境で収集されたものを使用した．詳細は 4.1.2 節，4.1.3 節で後述する．

4.1.2 正常データ

正常トラフィックデータは，事前告知を行った上で研究室 LAN 内で 12 時間を 1 区切りとして 5 回分，合計 2.5 日分の収集を行った．なお，HTTP リクエストが 1 個以上存在する通信ホストペアは 7138 個存在し，そのうち 5 個以上存在していて特徴量を抽出することができた特徴ベクトル 2378 個をデータセットとして用いた．

4.1.3 感染データ

感染トラフィックデータは NTT セキュアプラットフォーム研究所の動的解析環境である BotnetWatcher[7] でマルウェアを動作させたときに収集したものを利用した．なお，HTTP リクエストが 1 個以上存在する通信ホストペアは 2708 個存在し，そのうち 5 個以上存在していて特徴量を抽出することができた特徴ベクトル 567 個をデータセットとして用いた．

4.2 手順

本研究では，正常・感染データ合わせて 2945 個のデータからなるデータセットについて，ランダムに 5 分割して

表 2 評価結果の分類
Table 2 Classification of evaluation result

		予測	
		正常	感染
正解	正常	TN	FP
	感染	FN	TP

交差検証を行った．すなわち，5 分割された各々のデータをテストデータ，残りを学習データとする評価実験を 5 回実施した．

マルウェア感染検知では正常データをマルウェア感染と誤検知をすることよりも，感染データを正常と判定して見逃しをすることのほうが深刻であると考えられるため，より多くのマルウェア感染データを正しく判定するというポリシーのもと，グリッドサーチでパラメータを求める基準を Recall 値が最大となるようにに設定した．また，パラメータ C と γ はそれぞれ-10000 から 10000 の 10 倍刻みの値を選択した．なお，SVM におけるカーネル関数は RBF カーネルを用いた．

4.3 評価方法

本研究のデータセットでは正常通信と感染由来の通信のデータ数に偏りがあるため，すべて正常と判定しても高い正解率が出てしまい，正解率のみの評価では識別器の性能を正しく評価できない．そこで，数に偏りがあっても評価することが可能な基準である Precision と Recall を用いて評価を行う．

TP（True Positive），FP（False Positive），TN（True Negative），FN（Flase Negative）の定義を示す．TP は実際は感染由来の通信であるものを正しく感染と判定できているものである．FP は実際は正常であるが，予測では誤って感染と判定したものである．TN は実際は感染であるものを正しく正常と判定することである．FN は実際は感染であるものを誤って正常と判定することを示す．

$Precision_N$ は正常データの Precision 値であり，正常データと判定したデータの中で正しく正常と判定できた割合を示し，$Precision_P$ は感染データの Precision 値であり，感染と判定した中で正しく感染と判定で来た割合を示し，$Recall_N$ は正常データの Recall 値であり，正常データの中で正しく正常データと判定できた割合を示し，$Recall_P$ は感染データの Recall 値であり，感染データの中で正しく感染と判定できた割合を示す．それぞれ式 4 から式 7 により求める．

$$Precision_N = \frac{TN}{TN + FN} \tag{4}$$

$$Precision_P = \frac{TP}{TP + FP} \tag{5}$$

$$Recall_N = \frac{TN}{TN + FP} \tag{6}$$

表 3 実験結果の分類
Table 3 Classification of experimental result

	TN	FP	FN	TP	$sample_N$	$sample_P$
1 回目	476	6	19	88	482	107
2 回目	461	5	15	108	466	123
3 回目	454	14	19	102	468	121
4 回目	466	8	20	95	474	115
5 回目	482	6	14	87	488	101

表 4 実験結果
Table 4 Experimental results

	$Precision_N$	$Precision_P$	$Recall_N$	$Recall_P$
1 回目	0.96	0.94	0.99	0.82
2 回目	0.97	0.96	0.99	0.88
3 回目	0.96	0.88	0.97	0.84
4 回目	0.96	0.92	0.98	0.83
5 回目	0.97	0.94	0.99	0.86
平均	0.96	0.93	0.98	0.85

$$Recall_P = \frac{TP}{FN + TP} \tag{7}$$

4.4 結果

実験を行った結果の TN，FP，FN，TP およびテストデータで使用した正常データの数である $sample_N$，および感染データの数である $sample_P$ を表 3 に示す．今回の 5 分割交差検証では，正常・感染データを合わせたデータセットをランダムに分割したが，表 3 からわかるように，全 5 回のテストにおいて正常と感染データのサンプル数の割合は同じくらいであることがわかる．

続いて実験結果から Precision および Recall を算出し，表 4 に示す．

$Precision_N$ および $Precision_P$ について，正常と判定したものについては，そのうち 96 %が正常であると正しく判定できており，感染と判定したものについては，そのうち 93 %が正しく判定できているため，誤判定が少ないと言える．また，$Recall_N$ や $Recall_P$ について，全正常データのうち 98 %は正しく判定できており，全感染データのうち 85 %は正しく判定できているため，見逃しが少ないということが言える．

4.5 考察

4.5.1 正常データと感染データの取得環境の違い

先行研究 [4][5] と同様に本研究でも正常データと感染データの取得環境は異なる．しかし，先行研究では通信ホストペアごとに特徴量をとっているのではなく，タイムスロット単位で特徴量をとっているものが多く [4][5]，同一タイムスロット中に発生する正常通信に影響を大きく受けることがある．具体的には，マルウェア感染由来の通信が発生し

ている同一のタイムスロットにおいて正常通信が多く発生している場合，抽出した特徴量に対してのマルウェア感染由来の通信の影響は弱まると考えられる．したがって，タイムスロット単位で特徴量を取る場合はデータの取得環境の違いによる影響を無視できないのに対して，本研究ではHTTP リクエスト/レスポンスペアを通信ホストペアごとに分割して特徴量を抽出しているため，同一タイムスロットに正常通信がどれだけ多く発生しても抽出する特徴量には影響を及ぼさない．したがって，データの取得環境の違いによる影響を無視することができると考えられる．

4.5.2 結果について

特に今回のデータセットは正常データ数と感染データ数の比が大きいため，$Precision_N$ と $Recall_N$ はかなり高い値となっていても，$Precision_P$ および $Recall_P$ も高い値を示していないと識別器の性能は良いとは言えない．しかし，今回の結果から $Precision_P$ および $Recall_P$ が高い値を示しているので，すべて正常データと判定する傾向を強くすることなく，なおかつ高精度に判定ができているといえる．特に $Recall_P$ に着目すると，5 回のテストの平均が 85 %であり，感染データのうち 15 %しか見逃しが発生していないことがわかる．したがって，一般的に検知が困難なマルウェア感染由来の HTTP トラフィックを高精度にアノマリ検知できたということが言える．

実環境において本稿で提案した手法を適用した場合について考える．実環境においては，正常データが大多数を占めており，感染データは正常データに比べてはるかに少ないと考えられる．したがって，正常データと感染データで数の比が非常に大きくなっても，感染データの見逃しを限りなく少なくする必要がある．実験では正常データ数と感染データ数の比が大きかったにもかかわらず，低い見逃し率で感染データを検知することができた．実験においては別々の環境で収集したトラフィックデータを元に評価を行ったが，正常端末と感染端末が同一の環境に存在し，感染端末が正常通信も同時に実施する実環境において提案手法を適用する場合においても，通信ホストペアごとに分割して特徴ベクトルを抽出するため，実験環境と同じ形の特徴ベクトルを生成できることは自明のため，高精度に検知することが可能であると考えられる．

4.5.3 マルウェアタイプごとの見逃し率について

評価実験の結果より，$Recall_P$ が高いことから，マルウェア感染由来の通信を正常通信と判定する見逃しが少ないことがわかった．しかしながら，近年の高機能マルウェアでは HTTP を用いた C&C サーバとのバックドア通信が行われることが多いため，この通信をいち早く検知して，攻撃者による目的遂行を食い止める必要がある．そこで，本稿で提案した手法がバックドア通信にも有効かどうかを確認するため，評価実験においてマルウェア感染由来の HTTP トラフィックを正常と判定した見逃し数をマルウェアタイプ名ごとに集計した．集計結果を表 5 に示す．なお，マルウェアタイプ名は特徴ベクトルを抽出したトラフィックを発生したマルウェア名で，検知名は Kaspersky[*2] の検知名におけるタイプ名を用いている．

表 5 マルウェアタイプごとの見逃し率
Table 5 False negative rate of malware types

マルウェアタイプ名	見逃し数/データ数	見逃し率
HEUR-Monitor	0/1	0.00
Monitor	0/1	0.00
Trojan-DDos	0/1	0.00
Worm	0/1	0.00
Packed	3/5	0.60
Trojan-Dropper	0/9	0.00
Trojan-Ransom	6/10	0.60
Trojan-PSW	5/12	0.42
Trojan-Downloader	1/21	0.05
HEUR-Downloader	0/24	0.00
WebToolbar	1/26	0.04
Trojan-Spy	6/50	0.12
Trojan	13/58	0.22
Downloader	1/77	0.01
AdWare	29/135	0.21
Backdoor	22/136	0.16
合計	87/567	0.15

表 5 よりマルウェアタイプ名が Backdoor の見逃し率が 16 %であることから，他のタイプのマルウェアと同等の検知精度が得られていることが確認できた．ゆえに本稿で提案した手法は C&C サーバとのバックドア通信を十分に検知できていると考えられる．また，サンプル数が少ない Packed や Trojan-PSW，Trojan-Ransom の 3 つのタイプを除けばどのタイプにおいても見逃し率が 22 %以下に抑えられていることから，特に検知が苦手なタイプのマルウェアが存在しない，有効な検知手法であることが言える．

4.5.4 複数の悪性通信ホストペアに基づく検知について

本稿では，単一通信ホストペアの HTTP トラフィックがマルウェア感染由来かどうかを判定する手法を提案しているが，実験結果において 15 %の感染由来のデータを正常と判定していることからもわかるように，感染由来のデータを見逃してしまうケースが考えられる．そこで，ある通信ホストペアの HTTP トラフィックが誤って正常由来と判定された場合でも，同一ホストの別の通信ホストペアが感染由来と判定されれば，再判定することによって，正しく感染由来と判定できると考えられる．

そこで，1 つのマルウェアを実行した際に，悪性な通信ホストペアがどれだけ発生するのかについて評価実験で使用したデータセットの調査を行った．この際の悪性な通信の基準としては，マルウェアが行ったリクエストが 5 個以

[*2] Kaspersky
http://www.kaspersky.com

上の通信ホストペアを悪性とした．

その結果，567 個の特徴ベクトルは，334 個の pcap データから抽出されていることがわかった．334 個の pcap データのうち，60 個からは複数の特徴ベクトルが抽出され，残り 274 個からは単一の特徴ベクトルのみが抽出されたことから，複数の悪性通信先を持つマルウェアが約 18 %存在していることがわかる．したがって，複数の悪性通信が発生しているホストに関しては，仮に感染によって発生した単一の通信ホストペアを正常通信と判定して見逃した場合でも，別のペアの HTTP トラフィックが感染由来と判定された場合に再判定することで，正しく感染由来の通信と判定できる余地があると考えられる．

5. おわりに

本稿では，リクエスト間隔とレスポンスボディサイズに基づくマルウェア感染由来の HTTP トラフィック検知手法を提案した．提案手法では，5 回の交差検証の結果，マルウェア感染由来の HTTP トラフィック検知の見逃し率を平均 15 %に収めることができた．

また，昨今の高機能なマルウェアでは C&C サーバとのバックドア通信に HTTP を用いられることが多いが，提案手法で検知することが可能かを調査するために，マルウェアタイプごとにマルウェア感染由来の HTTP トラフィック検知の見逃し率を算出したところ，Backdoor というマルウェアタイプに関して見逃し率が 16 %であった．この数値は他の全マルウェアの見逃し率と差がないため，HTTP を用いたバックドア通信を検知することが可能であると考えられる．また，どのマルウェアタイプに対しても概ね見逃しが少なく検知できることを示したことから，バックドア通信に限らず，多くのマルウェアタイプにも適用可能な汎用的な手法であると考える．

現状では，リクエスト間隔およびレスポンスのボディサイズのみから特徴量をとっているため，正常通信と感染由来の通信で似たような特徴量を持つことが発生し，見逃しが起きていると考えられるため，より正常と感染を識別できるような新たな特徴量を発見することが課題である．

また, 現状では個々の通信ホストペアから発生した HTTP トラフィックがマルウェア感染由来かどうかを判定しているが，正常通信と似たようなリクエスト間隔やレスポンスサイズの場合見逃しが発生する可能性がある．この問題に対して，同一ホストの別の通信ホストペアでマルウェア感染由来と判定しているかどうか，あるいは複数の悪性と疑われる通信が見られるかという新たな指標を設けることで，感染の見逃しを少なくできると考えられるため，当該指標を利用した仕組みの実装と評価をすることも課題である．

参考文献

[1] 独立行政法人情報処理推進機構:「高度標的型攻撃」対策に向けたシステム設計ガイド，https://www.ipa.go.jp/files/000046236.pdf (2014).

[2] 特定非営利活動法人日本セキュリティ監査協会: APT 対策入門：新型サイバー攻撃の検知と対応，インプレス R&D (2012).

[3] マクニカネットワークス株式会社: 標的型攻撃の実態と対策アプローチ第 1 版，http://www.macnica.net/file/security_report_20160613.pdf (2016).

[4] 鮫島礼佳，畑田充弘，吉浦裕，市野将嗣: 感染挙動の時系列情報のクラスタリングに基づくマルウェア検知手法，コンピュータセキュリティシンポジウム 2015，pp. 536-543 (2015).

[5] 大月優輔，市野将嗣，川本研治，畑田充弘，吉浦 裕: マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価，コンピュータセキュリティシンポジウム 2012，pp.691-698 (2012).

[6] 鳥居悟，清水聡，森永正信: RAT 通信監視手法の提案: コンピュータセキュリティシンポジウム 2012，pp.571-578 (2012).

[7] 青木一史，川古谷裕平，岩村誠，伊藤光恭: 半透性仮想インターネットによるマルウェアの動的解析，コンピュータセキュリティシンポジウム 2009，pp.1-6 (2009).