SSL/TLSで暗号化されたWeb通信に対する侵入検知システム

山田 明[†1,†2] 三宅 優[†1]
寺邊 正大[†2] 橋本 和夫[†2]

サービス提供の媒体としてWebアプリケーションが広く利用されるようになったため，Webアプリケーションに対する悪意のある攻撃が社会的な問題となっている．侵入検知システムは，このような攻撃を目的としたサーバへの侵入を検知するための監視ツールであるが，通信がSSLなどにより暗号化されている場合には攻撃検知が困難となる．暗号化された通信を解析する方式としては，統計解析により，クライアントの接続先を識別する方式がある．しかし，暗号化前後の対応関係を解析するため，事前に通信を調査する必要がある．本論文では，暗号化された通信を解析する方式を侵入検知システムに適用することにより，SSLにより暗号化されたWeb通信における攻撃を検知できる新たな方式を提案する．提案方式は，クラスタリングにより通信を識別し，事前の通信事例データの収集を必要としないことを特徴とする．また，提案方式の有効性を，実際のLANゲートウェイで収集したデータおよびDARPA IDS評価データの2つのデータを用いて評価し，その結果，暗号化された状態での攻撃検知性能を確認した．

An Intrusion Detection System for SSL/TLS Encrypted Web Traffic

Akira Yamada,[†1,†2] Yutaka Miyake,[†1] Masahiro Terabe[†2]
and Kazuo Hashimoto[†2]

As web applications are widely used for a variety of services, attacks against web applications cause serious social problems. Intrusion Detection Systems (IDSes) are a tool to monitor illegal access to service providing servers, however, IDSes do not work effectively when the accesses are encrypted by protocols. This paper presents a novel method of anomaly detection for encrypted web traffic, which analyzes contents of encrypted traffic using only data size and timing without decryption. Though conventional encrypted traffic analysis methods require a pre-process that constructs the model of relationship between encrypted and unencrypted traffic, the proposed method eliminates the pre-process by adapting clustering techniques. The evaluation is conducted using an actual dataset gathered at a gateway of a network and the DARPA dataset.

1. はじめに

Webアプリケーションの普及にともない，SQLインジェクションやクロスサイトスクリプティングなどの攻撃手法を利用したWebアプリケーションへの攻撃が問題となっている．これらの攻撃を監視する方式として，侵入検知システム（Intrusion Detection System: IDS）が提案されている．IDSは，シグネチャと呼ばれる攻撃の特徴をあらかじめ登録しておき，シグネチャと通信の内容を比較することにより攻撃を検知する．

しかし，Webアプリケーションの通信がSSL[1),2)]（Secure Socket Layer）やTLS[3)]（Transport Layer Security）により暗号化されている場合，通信の内容を監視することが困難となる．SSL/TLSは，Webアプリケーションにおいてクレジットカード番号などの重要な情報が送受信される際に使用される．したがって，従来のIDSによる監視が困難となる状況は頻繁に発生し，深刻な問題となっている．

暗号化されている通信における攻撃検知の方式として，通信を復号してから従来の攻撃検知を行う方式がある．たとえば，暗号化に用いる鍵を複製してIDSに組み込み，IDSでも復号化できるようにする方式や，Webサーバに IDSを組み込み，サーバ上で復号されたデータに対して攻撃検知を行う方式がある[4)]．しか

†1 KDDI研究所
KDDI R&D Laboratories
†2 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

し，鍵の複製を利用した方式はシステム全体の安全性を低下させてしまう危険性がある．また，サーバへの組み込みも容易ではなく，簡単に適用できるとはいい難い．

一方，暗号化されている通信を復号せずに解析する方式[5)–10)]についても研究が進められている．これらの方式では，送受信されるデータサイズを統計的に解析し，通信種別や接続先を推定するが，暗号化前後のデータサイズを比較するため，あらかじめ暗号化されていない通信を収集する必要がある．しかし，実際の Web サーバにおいて暗号化前の情報を入手できる場合が少ないため，実問題への適用が難しい．

本論文では，暗号化された通信に含まれる攻撃を復号せずに検知できる新たな侵入検知方式を提案する．提案方式は，はじめに，通信からデータサイズなどの特徴ベクトルを抽出する．そして，特徴ベクトルをクラスタリングすることにより識別し，頻度および HTTP の振舞いより攻撃を検知する．教師なし学習により分類を行っているため，事前に暗号化されていない通信事例を収集する必要がない．

提案方式の有効性を示すため，LAN ゲートウェイにおいて収集したデータおよび DARPA IDS 評価データ[11),12)]の 2 種類のデータを用いて検知精度の評価を行った．評価の結果，LAN ゲートウェイデータにおいて検知率 0.97，誤検知率 0.05，DARPA データにおいて検知率 0.78，誤検知率 0.25 の性能を達成した．IDS として検知率が十分ではないものの，暗号化された通信に対して復号を行わずに攻撃を検知できることが明らかとなった．

本論文は，まず 2 章において暗号された通信に対する従来の攻撃検知方式を紹介し，3 章において提案方式を説明する．4 章において提案方式を評価し，5 章で考察する．最後に，6 章で結論を述べる．

2. 従 来 方 式

2.1 暗号化された通信を復号して監視する方式

SSL/TLS により暗号化された通信を復号してから監視する方式は，以下の 3 種類に分類される[4)]．しかし，いずれの方式にもいくつかの問題点がある．

コネクション終端型：SSL/TLS のコネクションを IDS にて終端し，IDS から Web サーバまでのコネクションを逆プロキシにより実現する．IDS から Web サーバまでの通信は暗号化されていないため解析が可能である．しかし，IDS 設置のため，ネットワーク構成を変更しなければならない点が問題である．

受動的復号型：Web サーバ内に保存されている SSL/TLS の秘密鍵を複製して，IDS の解析に利用する．RSA 暗号による鍵共有が行われる場合，通信を IDS において復号することが可能である．しかし，秘密鍵の複製を IDS に組み込むため，鍵が流出する危険性が高くなる．また，Diffie-Hellman の鍵共有が用いられる場合に適用できない．

ホストインストール型：Web サーバに IDS をインストールし，Web サーバにおいて復号された後に通信を監視する．もしくは，通信ではなくアクセス履歴を解析することにより攻撃を監視する．しかし，IDS が Web サーバの計算機資源を消費してしまうため，ホスト全体の性能劣化やライブラリの競合を引き起こす可能性がある．

したがって，いずれの方式もシステム全体の安全性や安定性を減少させてしまう問題点がある．

2.2 暗号化された通信を復号せずに監視する方式

暗号化された通信を復号せずに処理する IDS は存在しないが，暗号化された通信の内容を推測する方式は提案されている．ただし，IDS に直接適用することは困難である．以下にそれぞれの方式について説明する．

SSL/TLS のプロトコル仕様[1)–3)]において，通信を解析することにより暗号化されているデータのサイズなどを推測されることが言及されている．そして，Wagner らは，SSL/TLS のプロトコルに対する攻撃についてまとめた[13)]．また，実際の Web サイトを用いて評価し，データサイズから特定の Web サイトへの通信が判別できると報告している[9),10)]．

Hintz は，SSL/TLS により匿名性を提供するサービス Safeweb[14)] に対する攻撃法を提案した[5)]．Hintz の提案は，データサイズから通信先の特徴を抽出するための効果的な方式であり，その後，この方式を発展させた方式が提案された．Sun らは，DMOZ[15)] に登録されている 100,000 の Web サイトが識別可能であることを示した[8)]．Danezis は，隠れマルコフモデルを適用することにより精度を上げる方式を示した[16)]．そして，Bissias らは，暗号化プロトコルとして SSH，IPsec，WEP，WPA を対象とする方式を提案した[6),7)]．

しかしながら，上記の方式は，暗号化されている通信の内容を推測することを目的としているため，解析に暗号化前後のデータの両方を必要とする．つまり，暗号化前のデータを事前に収集する必要がある．特に，秘密情報を取り扱う Web サーバを監視する場合，暗号化前のデータを収集することが困難となる．

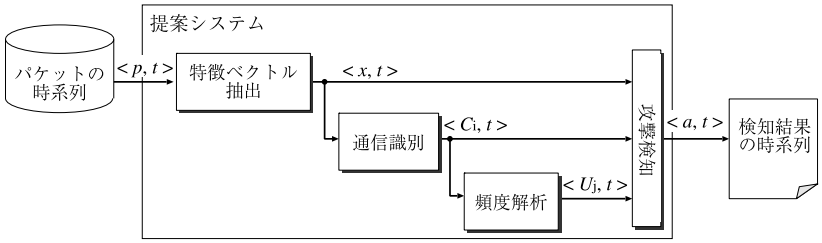


図 1 提案方式の処理の流れ
Fig. 1 Flow of the proposed system.

2.3 暗号化されていない Web 通信に対する攻撃検知

暗号化されていない通信に対する IDS の中で，Web 通信に特化した方式が提案されている[17)–22)]．これらの方式は，具体的な攻撃としてクロスサイトスクリプティングやバッファオーバフロー，スクリプトインジェクションなどを検知対象としている．そして，HTTP ヘッダのリクエストラインに含まれる URL に付随する変数や，リクエストされる URL の遷移を解析することによって攻撃を検知している．

しかし，SSL/TLS により通信が暗号化される場合，HTTP ヘッダの変数が利用できなくなってしまう．また，リクエストされる URL を識別できなくなるため，遷移を把握することもできなくなる．したがって，暗号化されても利用できる変数だけを用いる場合，検知精度の低下は避けられない．このため，上記方式を暗号化された Web 通信に適用することは困難である．

2.4 従来方式の問題点

以上をまとめると，従来方式の問題点は次のとおりである．

- 通信を復号するため Web サーバの秘密鍵を必要とする．
- 通信を識別するため暗号化されていない通信を用いた事前処理を必要とする．
- 攻撃を検知するための特徴量や頻度などの情報を抽出できない．

そこで，本論文では，暗号化されている通信を復号せずに監視する方式を応用した侵入検知方式を提案する．提案方式は，HTTP リクエストやレスポンスと予測されるデータのサイズを抽出し，HTTP ヘッダの変数の代わりに攻撃検知に利用する．また，通信の接続先を識別することによって，暗号化されている通信の接続先 URL の遷移を把握する．

また，通信の識別において教師なし学習に基づくクラスタリングを行うため，あらかじめ暗号化されていない通信事例を必要としない．また，攻撃検知のために必要な特徴量として，暗号化された通信から得られる情報のみを使用する，という特徴を持つ．

3. 暗号化されている Web 通信に対する侵入検知方式の提案

3.1 処理の流れ

図 1 に提案システムの処理の流れを示す．提案方式は，パケットの時系列 $\langle p,t\rangle$ を入力として検知結果の時系列 $\langle a,t\rangle$ を出力するシステムである．処理は，特徴ベクトル抽出，通信識別，頻度解析の部分からなる．ここで，パケット p には，IP ヘッダ，TCP ヘッダ，HTTPS ヘッダの情報が含まれており，t はパケットの観測時間である．

まず，特徴ベクトル抽出において，パケットの時系列 $\langle p,t\rangle$ は Web クライアントのアクティビティごとに分割され，特徴ベクトルの時系列 $\langle x,t\rangle$ に変換される．ここで，アクティビティとは，Web クライアントにおけるハイパーリンクのクリックなどに該当する．また，特徴ベクトル x は，各アクティビティを表す特徴量の集合である．3.2 節に詳細を示す．

次に，通信識別において，特徴ベクトルの時系列 $\langle x,t\rangle$ はクラスタリングにより識別され，クラスタの時系列 $\langle C_i,t\rangle$ に変換される．ここで，特徴ベクトルをクラスタリングした結果，同じクラスタに属するアクティビティは，同一のアクティビティと見なす．C_i については，3.3 節にて説明する．

さらに，頻度解析において，クラスタの時系列 $\langle C_i,t\rangle$ はクラスタ間遷移の時系列 $\langle U_j,t\rangle$ に変換される．複数のクラスタ間の遷移を考慮すると，保持しなければならない遷移の数が指数的に増加してしまう．そこで，記憶量を制限しながら遷移の頻度を計算する．クラスタ間遷移 U_j については，3.4 節で説明する．

最後に攻撃検知において，$\langle x,t\rangle$，$\langle C_i,t\rangle$，$\langle U_j,t\rangle$ を使用して，攻撃の有無を判定して，検知結果の時系列 $\langle a,t\rangle$ を出力する．

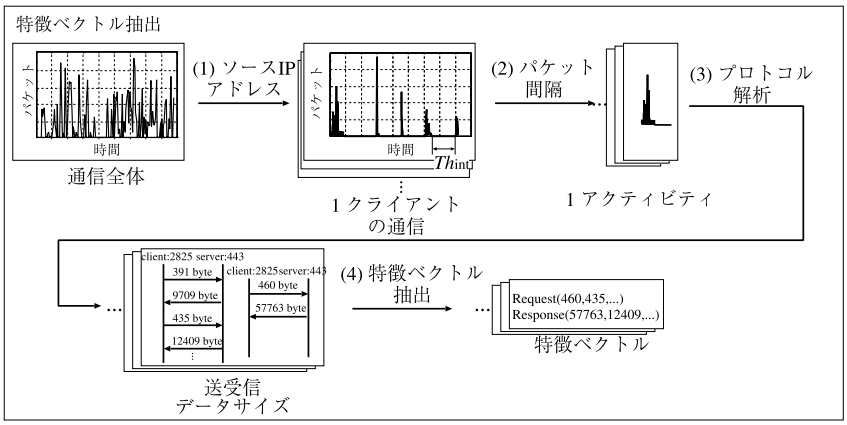


図 2 特徴ベクトルの抽出
Fig. 2 Feature vector extraction.

3.2 特徴ベクトルの抽出

特徴ベクトル抽出では，パケットの時系列 $\langle p, t\rangle$ をアクティビティごとに分割して，特徴ベクトルの時系列 $\langle x, t\rangle$ を出力する．以下に処理手順を示す（**図 2**）．

(1) ソース IP アドレスに基づいて通信を分割することにより，クライアントごとの通信を抽出する．ここでは，1 つのクライアントは 1 つの IP アドレスからサーバにアクセスすると見なしている．

(2) 1 つのクライアントからの通信のパケット到着間隔を計算する．到着間隔が Th_{inter} より大きい場合に，通信を異なるアクティビティとして分割する．つまり，連続的に観測されるパケットを 1 つのアクティビティと見なしている．

(3) 1 つのアクティビティに含まれるパケットに対して，TCP セッション再構築を行う．さらに，SSL/TLS のプロトコル解析を行うことにより，送受信されているデータサイズを抽出する．SSL/TLS のプロトコル解析については後述する．

(4) 特徴ベクトルは，1 つのアクティビティの中で送受信されているデータサイズとする．ただし，特徴ベクトルは，次元が大きくなると解析が難しくなるため，送受信のそれぞれについて最大数 Th_{fvmax} を設定しサイズを降順に並べ替えたものとする．ここで，データの順序はブラウザの実装に依存するため考慮していない．

SSL/TLS のプロトコル解析では，暗号化されている通信からデータサイズを抽出する．ここで，SSL/TLS は，**図 3** に示すように Record Layer プロトコルと Change Cipher Spec，Handshake，Alert プロトコルなどから構成される．Record Layer プロトコルのヘッダには，上位のプロトコルの種類およびデータサイズなどの情報が暗号化されずに記録される[3]．

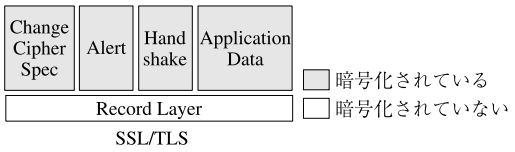


図 3 SSL/TLS のプロトコル構成
Fig. 3 SSL/TLS protocols.

そこで，データ送受信に関係のないプロトコルを無視し，データが送受信される Application Data のサイズのみを抽出する．ただし，SSL のバージョンが 2.0 では，Record Layer プロトコルのヘッダにデータサイズが記録されていない．そこで，同一方向に送信されるデータをひとかたまりのデータと見なしてデータサイズを抽出する．

たとえば，460 byte のリクエストと 58,052 byte のレスポンス，391 byte のリクエストと 9,709 byte のレスポンスが観測される場合，以下のような特徴ベクトルが抽出される．ここで，特徴ベクトルの次元数の閾値は $Th_{\text{fvmax}} = 5$ としている．

$x =$						
(Request(	460	391	0	0	0	),
Response(	58052	9709	0	0	0	))

3.3 通信識別

通信識別では，特徴ベクトルの時系列 $\langle x, t\rangle$ を K-means 法を改良したアルゴリズムにより識別してクラスタの時系列 $\langle C_i, t\rangle$ を出力する．

図 4，**図 5** に，特徴ベクトルの時系列 $\langle x, t\rangle$ から時系列が空になるまで要素を取り出して，処理を行

```
Function ClusterUPDATE(
  input : S_x:Stream of ⟨x, t⟩,
    Th_cls : クラスタ成長のための閾値,
  output : Stream of ⟨C_i, t⟩
){
  Var x : 特徴ベクトル,
    C_i : 特徴ベクトル x に対応するクラスタ,
    C : クラスタの集合,
    k : integer; // クラスタの数

  set 閾値 Th_cls;
  k ← 0;
  C ← ∅ (空集合) ;
  (x, t) ← pop(S_x);

  Repeat
    C_i ← ClusterUPDATE-1(x, Th_cls);
    output ⟨C_i, t⟩;
    // 特徴ベクトルに対応するクラスタの時系列生成
    (x, t) ←pop(S_x);
  Until (x = end of Stream);
}
```

図 4 通信識別アルゴリズムメイン関数
Fig. 4 Main function of traffic recognition.

```
Function ClusterUPDATE-1(
  input : x : 特徴ベクトル,
    Th_cls : クラスタ成長のための閾値,
  output : C_i : クラスタ
){
  Var i : integer,
    C_k : クラスタ,
    m_k : クラスタの C_k の中心;

  If (k =0){
    //新しいクラスタ C_1 作って x を割り当てる
    k ← 1; C_k ← {x}; m_k ← x;
    Return C_k;
  }
  else {
    i ← arg min_{i∈{1,2,...k}} dist(x, m_i);
    if(dist(x, m_i) > Th_cls){
      //新しいクラスタ C_k を作って x を割り当てる
      k ← k + 1; C_k ← {x}; m_k ← x;
      Return C_k;
    }
    else {
      //既存のクラスタ C_i に x を割り当てる
      C_i ← C_i ∪ {x};
      m_i ← 1/|C_i| (m_i × (|C_i| − 1) + x);
      Return C_i;
    }
  }
}
```

図 5 通信識別アルゴリズムサブルーチン
Fig. 5 Subroutine of traffic recognition.

うメイン関数 ClusterUPDATE とそのサブルーチン ClusterUPDATE-1 を示す．関数 ClusterUPDATE-1 は，サイドエフェクトとしてクラスタの集合 C を成長させる．

ここで，pop() はストリームから要素を 1 つずつ取り出す関数，dist() はユークリッド距離を計算する関数である．また，Th_{cls} はクラスタ成長のための閾値である．出力としてクラスタの時系列 $\langle C_i, t\rangle$ が得られる．

```
Function TransitionUPDATE(
  input : S_u : Stream of ⟨u, t⟩,
    Th_mem : 最大記憶量の閾値,
  output : Stream of ⟨U_j, t⟩
){
  Var u : クラスタ間の遷移,
    U_j : クラスタ間の遷移,
    U : クラスタ間の遷移の集合,
    ℓ : integer; //クラスタ間の遷移の数

  set 閾値 Th_mem;
  ℓ ← 0;
  U ← ∅ (空集合) ;
  (u, t) ←pop(S_u);

  Repeat
    U_j ← TransitionUPDATE-1(u, Th_mem);
    output ⟨U_j, t⟩ ;
    // クラスタ間の遷移の時系列生成
    (u, t) ←pop(S_u);
  Until (u= end of Stream);
  }
```

図 6 頻度解析アルゴリズムメイン関数
Fig. 6 Main function of frequency analysis.

3.4 頻度解析

頻度解析では，クラスタの時系列 $\langle C_i, t\rangle$ を入力として，クラスタ間における遷移の時系列 $\langle U_j, t\rangle$ を出力する．ここで，クラスタ間の遷移をあらたに $u \in C \times C$ として置き直す．

図 6，**図 7** にクラスタ間の遷移の時系列 $\langle u, t\rangle$ から時系列が空になるまで要素を取り出して，処理を行うメイン関数 TransitionUPDATE とそのサブルーチン TransitionUPDATE-1 を示す．出力として遷移の時系列 $\langle U_j, t\rangle$ とその頻度 $count\{U_j\}$ が得られる．また，クラスタ間遷移の最大記憶量を Th_{mem} とする．

3.5 攻撃検知

攻撃検知では，各通信の特徴ベクトルの時系列 $\langle x, t\rangle$，クラスタの時系列 $\langle C_i, t\rangle$，クラスタ間の遷移時系列 $\langle U_j, t\rangle$ のすべてを使用して，攻撃の有無を判定して検知結果の時系列 $\langle a, t\rangle$ を出力する．検知結果 a は，攻撃ありの場合 1，なしの場合 0 となる．

$$a = \begin{cases} 1 & \text{if 攻撃あり} \\ 0 & \text{if 攻撃なし} \end{cases}$$

攻撃の判定のために，それぞれの変数 $\langle x, C_i, U_j, t\rangle$

```
Function TransitionUPDATE-1(
  input : u : クラスタ間の遷移,
    Th mem : 最大記憶量の閾値,
  output : U_j : クラスタ間の遷移
){
  Var j : integer,
    time{U_j} : クラスタ間遷移の観測時間,
    count{U_j} : クラスタ間遷移の頻度;

  If (ℓ =0){
    //新しい遷移 U_ℓ を作ってカウンタを初期化する
    ℓ ← 1; U_ℓ ← u; U ← {U_ℓ};
    time{U_ℓ} ← t;
    count{U_ℓ} ← 1;
    Return 1;
  }
  else{
    for(j = 1; j <= ℓ; j++){
      if(U_j == u){
        //既存の遷移 U_j のカウンタを加算する
        time{U_j} ← t;
        count{U_j} ← count{U_j} + 1;
        Return U_j;
      }
    }
    If(ℓ < Th mem){
      //新しい遷移 U_{ℓ+1} を作ってカウンタを初期化する
      ℓ ← ℓ + 1; U_ℓ ← u; U ← U ∪ {U_ℓ};
      time{U_ℓ} ← t;
      count{U_ℓ} ← 1;
      Return U_ℓ;
    }
    else{
      //最も古い遷移を消去して，新しい遷移を記録する
      j = arg min_{j={1,2,...,ℓ}} time{U_j};
      U_j =time{U_j} =count{U_j} = ();
      U_j ← u;
      time{U_j} ← t;
      count{U_j} ← 1;
      Return U_ℓ;
    }
  }
}
```

図 7 頻度解析アルゴリズムサブルーチン関数
Fig. 7 Subroutine of frequency analysis.

に対して閾値を設ける．閾値は，攻撃データセットがあらかじめ与えられている場合，分類学習により求めることができる．また，システム管理者が各変数に対して発見的に閾値を設定することもできる．本論文では，発見的に閾値を決定する．

実際の攻撃を調査した結果，以下のような傾向が見られた．

- 一般的に運用されているサーバでは，正常な通信の方が異常な通信よりも頻度が高い．
- HTTP においては，クライアントがサーバから画像などの大きなサイズのコンテンツを取得するため，リクエストサイズが小さくレスポンスサイズが大きい．

そこで，これらを発見するために有益なパラメータとして，以下の閾値を設ける．

最大リクエストサイズの閾値：特徴ベクトル x に含まれる最大のリクエストサイズに対する閾値 Th_{req}

最大レスポンスサイズの閾値：特徴ベクトル x に含まれる最大のレスポンスサイズに対する閾値 Th_{res}

クラスタ頻度の閾値：最小のクラスタ C_i の頻度に対する閾値 Th_{freq}

これらの閾値の決定については，4.2 節で詳しく述べる．**図 8** に攻撃検知アルゴリズムを示す．

```
Function Detection(
  input : S : stream of ⟨x, C_i, U_j, t⟩)
    Th req : リクエストサイズの閾値,
    Th res : リスポンスサイズの閾値,
    Th freq : クラスタ頻度の閾値,
  output : Stream of ⟨a, t⟩
){
  Var a : 検知結果;

  (x, C_i, U_j, t) ← pop(S);

  Repeat
    a ← 0;
    foreach ⟨x, C_i, U_j, t⟩
    if( x.request.max > Th req   ||
        x.response.max > Th res   ||
        |C_i| < Th freq
        ){
        a ← 1;
    }
    output ⟨a, t⟩;
  Until (⟨x, C_i, U_j⟩ = end of Stream);
  }
}
```

図 8 攻撃検知アルゴリズム
Fig. 8 Attack detection algorithm.

4. 評　　価

4.1 データセット

提案方式の評価をするために，LAN のゲートウェイにおいて収集したデータおよび DARPA IDS 評価データ[12] の 2 種類を用いる．暗号化された評価データは，正常な通信と攻撃に分類することが困難である．そこで，本論文では HTTPS（HTTP over SSL/TLS）の代わりに HTTP を用いて評価し，暗号化によるパディングの影響を別途評価した．

LAN ゲートウェイにおいて収集したデータ，正常

表 1 データセット
Table 1 Datasets.

		MByte	リクエスト数	アクティビティ数
LAN ゲートウェイデータ	通常の通信	749.8	81,386	11,977
	攻撃	1.9	499	463
DARPA データ	通常の通信	2,666.8	428,630	56,261
	攻撃	172.0	2,933	481

表 2 LAN ゲートウェイデータに含まれる攻撃
Table 2 Attacks in LAN gateway.

攻撃	説明
HTTP スキャン	Web サーバの存在を見つけるためのスキャン
Proxy スキャン	プロキシサーバの存在を見つけるためのスキャン
CVE-2005-1921	PHP PEAR XML-RPC 実装における脆弱性を利用したコードインジェクション攻撃
MS03-51	Microsoft FrontPage Server Extensions のバッファオーバランの脆弱性を利用したコードの実行
MS04-007	ASN.1 の脆弱性によるコードの実行

な通信として LAN からインターネット上の正規サイトへの通信を，攻撃としてインターネット上に設置したハニーポットへの通信を使用する．また，DARPA IDS 評価データは，攻撃の情報が得られる 4，5 週目の TCPDUMP データを用いる．攻撃として HTTP に関する 6 種類を，正常な通信として攻撃を取り除いた HTTP 通信を使用する．**表 1** に評価に用いたデータセットの内訳を示す．

表 2，**表 3** に評価データに含まれる攻撃とその説明を示す．攻撃には，スキャン，コードインジェクション，バッファオーバフローなどの種類の攻撃が含まれている．また，攻撃名は，Common Vulnerabilities and Exposures [⋆1] および，Microsoft Security Bulletin [⋆2,⋆3] によるものである．

4.2 閾値決定

3.5 節の攻撃検知において，リクエストサイズの閾値 Th_{req}，レスポンスサイズの閾値 Th_{res}，クラスタ頻度の閾値 Th_{freq} の 3 種類の閾値により攻撃を検知している．閾値は，それぞれの値を変化させることにより最適な値を設定する．

表 3 DARPA IDS 評価データに含まれる攻撃
Table 3 Attacks in DARPA dataset.

攻撃	説明
Apache2	Web サーバ Apache に対する DoS 攻撃
Back	Web サーバ Apache に対する DoS 攻撃
Phf	CGI スクリプトに対するコマンドインジェクション攻撃
Crashiis	Web サーバ IIS に対する DoS 攻撃
Mscan	総合的な脆弱性監査ツール
Ntinfo	Windows NT サーバに対する脆弱性監査ツール

ただし，頻度のみによる検知を行う場合，頻繁に発生する攻撃を誤検知してしまう可能性が高い．そこでまず，クラスタ頻度の閾値を $Th_{\text{freq}} = \infty$ に固定して，リクエスト，レスポンスの閾値 Th_{req}，Th_{res} を変化させて最適な値を決定する．そして，リクエスト，レスポンスの閾値を最適値に固定して，頻度の閾値を変化させる．

また，アルゴリズム実行のために必要なその他の閾値はそれぞれ以下のように設定した．それぞれの値は，データセットの通常の通信において値を変化させ，通信の識別率が高くなる値に設定した．また，プログラムの実行において記憶量が十分であったため Th_{mem} の制限を設けなかった．

- 特徴ベクトル抽出におけるパケット間隔の閾値 Th_{inter}：1 sec.
- 特徴ベクトル抽出における特徴ベクトルの次元数の閾値 Th_{fvmax}：10
- 通信識別におけるクラスタ生成のための閾値 Th_{cls}：100 byte
- 頻度解析における記憶量の閾値 Th_{mem}：∞

4.3 検知精度

図 9，**図 10** にリクエストサイズの閾値 Th_{req} とレスポンスサイズの閾値 Th_{res} を変化させたときの検知率（Detection rate/TPR: True Positive Rate）および誤検知率（FPR: False Positive Rate）の変化を示す．

図 9 は LAN ゲートウェイデータである．実線が Th_{res} を 400 に固定し Th_{req} を変化させたときの結果であり，点線が Th_{req} を 3000 に固定し Th_{res} を変化させたときの結果である．$Th_{\text{req}} = 3000$，$Th_{\text{res}} = 400$ とするとき，誤検知率 0.05，検知率 0.97 を達成する．

図 10 は DAPRA IDS 評価データの結果である．実線が Th_{res} を 800 に固定し，点線が Th_{req} を 2000 に固定したときの結果である．$Th_{\text{req}} = 3000$，

⋆1 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1921
⋆2 http://www.microsoft.com/technet/security/Bulletin/MS04-007.mspx
⋆3 http://www.microsoft.com/technet/security/Bulletin/MS03-051.mspx

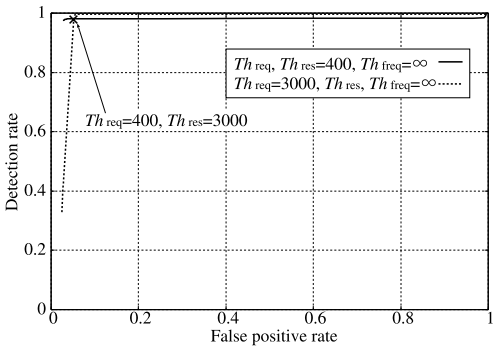


図 9 LAN ゲートウェイデータにおける検知率
Fig. 9 Detection rate for LAN data.

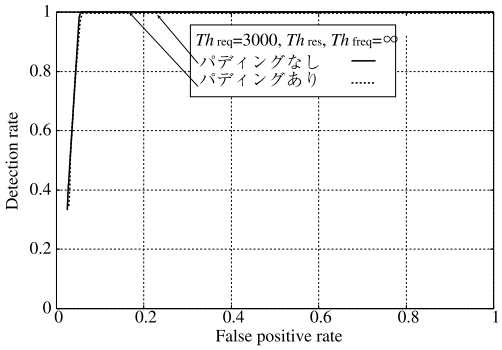


図 11 パディングによる検知率の変化
Fig. 11 Detection rate for random padded dataset.

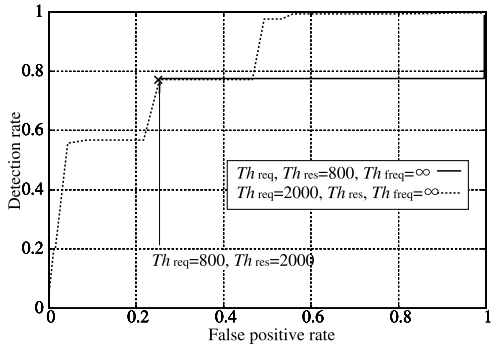


図 10 DARPA データにおける検知率
Fig. 10 Detection rate for DARPA data.

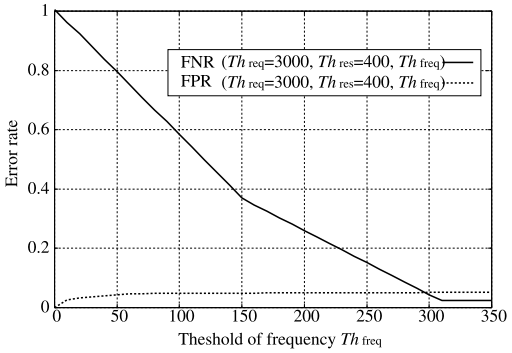


図 12 LAN ゲートウェイデータにおける頻度の閾値に対する検知率の変化
Fig. 12 Result of frequency analysis for LAN data.

$Th_{\mathrm{res}} = 400$ とするとき，誤検知率 0.25，検知率 0.78 を達成する．

図 11 に LAN ゲートウェイデータに対して，パディングを適用した場合の検知率，誤検知率の変化を示す．パディングは，ブロック暗号を利用する場合，暗号化の対象となるデータがブロックサイズの整数倍になるように調整する目的で利用される．SSL の場合は，ブロックサイズの整数倍にするための最小の値が追加されるが，TLS の場合は，ブロックサイズの整数倍となるのであれば，0-255 byte までの任意の値をパディングできる．

評価では TLS のパディングを適用した．つまり，ブロックサイズが 8 byte（64 bit）の場合，3 byte のデータに対して，TLS は，5, 13, 21, ... byte のいずれかを追加した．評価の結果，パディングによる検知精度の劣化がほとんどみられないことが確認された．

図 12，**図 13** にクラスタ頻度の閾値 Th_{freq} を変化させたときの見逃し率（FNR: False Negative Rate），誤検知率の変化を示す．見逃し率は，1 から検知率を引いた値である．

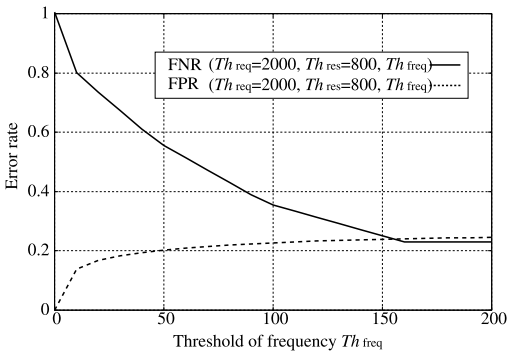


図 13 DARPA データにおける頻度の閾値に対する検知率の変化
Fig. 13 Result of frequency analysis for DARPA data.

図 12 は LAN ゲートウェイデータで評価したときの結果である．Th_{req}，Th_{res} をそれぞれ 3000，400 に固定し，Th_{freq} を変化させたときの見逃し率，誤検知率が示されている．図 13 は DAPRA IDS 評価データの結果であり，Th_{req}，Th_{res} をそれぞれ 2000，800 に固定し，Th_{freq} を変化させたときの見逃し率，誤検知率を示している．

クラスタ頻度の閾値により，トレードオフの関係にある見逃し率，誤検知率を制御できることが分かる．クラスタ頻度の制約により，攻撃検知アルゴリズムは，頻度が低いアクティビティのみを攻撃と判定する．そこで，同じ種類の誤検知が頻繁に発生するとき，これを利用して誤検知を削減する効果がある．ただし，逆に繰り返し行われる攻撃を見逃してしまう可能性もある．このような攻撃を検知するためには，攻撃のシグネチャとしてクラスタを登録する必要がある．

5. 考　　察

5.1 誤検知率

LAN ゲートウェイデータの結果は，DARPA データに比べて攻撃検知の精度が高かった．そこで，LAN ゲートウェイデータにおいて攻撃検知に効果があったルールを調査した．

LAN ゲートウェイデータでは，最大レスポンスのルールにより多くの攻撃が検知されていいた．ハニーポットにより収集された攻撃には，攻撃失敗によるエラーレスポンスが多く含まれている．最大レスポンスのルールは，その特徴をとらえられることが分かった．

また，DAPRA データにおいて見逃しが発生した攻撃を詳細に調査した．その結果，小さなサイズのスクリプトを挿入することによりサーバのパスワードファイルを奪取する Phf 攻撃であることが分かった．暗号化された場合，Phf 攻撃における通信は，HTTP においてサーバのコンテンツを取得する通信が類似してしまい，検知されなかった．

5.2 暗号化されていない Web 通信に対する攻撃検知率

参考のために，暗号化されていない Web 通信に対する既存の侵入検知方式の検知率を示す．通信の内容をすべて解析に利用できるため，提案方式に比べ検知率は高い．しかし，暗号化された場合ほとんどの情報が検知に利用できなくなるため，検知率が大幅に下がると思われる．

KV04[17]，EGD05[21]，RVKK06[22] は，HTTP リクエストから文字の分布や長さなどの値を抽出し，マルコフモデルや統計解析により攻撃を検知する．また，MC02[24]，SGFSTYZ02[25] は，HTTP だけでなく他のプロトコルも対象とする方式であり，確率モデルやプロトコルの状態遷移により攻撃を監視する．

表 4，**表 5** に各々の方式の評価データおよび検知率を示す．表 4 は，評価データに実際の通信を用いている方式である．攻撃には，既知の Exploit コード，攻撃ツールや攻撃データベースを使用している．暗号化されていない通信を解析できるため，提案方式に比べ高い検知率を達成している．

表 4　既存の方式の誤検知（1）
Table 4　Error rate of conventional anomaly detection (1).

方式	KV04[17]	EGD05[21]	RVKK06[22]
正常な通信	大学の Web サーバ	DARPA データ	大学の Web サーバ
攻撃	実際の exploit コード 11 種類	攻撃データベースの攻撃 86 種類	実際の Exploit コード 10 種類[23]
FPR	2.12×10^{-4}	0.01	0.0145
FNR	0	0	0

表 5　既存の方式の検知率（2）
Table 5　Error rate of conventional anomaly detection (2).

方式		MC02[24]	SGFSTYZ02[25]
検知率	Apache2	3/3	2/2
	Back	0/4	3/3
	Crashiis	5/7	–
	Ntinfoscan	2/3	–
	Mscan	1/1	1/1
	Phf	2/3	–

表 5 は，評価データに DARPA IDS 評価データを用いている方式である．これらの論文では，検知率ではなく検知数のみを記載している．表 5 によると，いくつかの攻撃について提案方式の方が検知率が高い．したがって，提案方式は，通信が暗号化されているにもかかわらず，MC02[24] や SGFSTYZ02[25] と同等の検知率を達成していることが分かる．

6. ま と め

本論文では，SSL により暗号化された Web 通信に対して，暗号化された通信を解析する方式を応用した侵入検知システムを提案した．提案方式は，クラスタリングにより分類，識別するため，従来の暗号化された通信の解析における問題点である事前の通信収集を必要としない，という特徴を持つ．

提案方式の有効性を確認するために，LAN ゲートウェイにおいて収集したデータおよび DARPA IDS 評価データを用いて検知率の評価を行った．その結果，実際のネットワークにおける攻撃を高い精度で検知できることを確認した．また，DARPA IDS 評価データに含まれる攻撃も特定の攻撃を除いて検知できることを確認した．

今後の課題は，多様な攻撃を用いて評価を行うことと，暗号化された場合に通常の通信と類似している攻撃の検知方式を検討することである．

謝辞　本論文を執筆するにあたって多くの方々の手

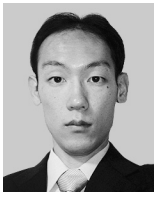
助けを受けた．特に，論文をまとめる際に多くの貴重な助言をいただいた Carnegie Mellon University の Adrian Perrig 助教授，Dawn Song 助教授，Ahren Studer 氏，暗号プロトコルの攻撃検知について助言をいただいた東北大学の Tarik Taleb 助教，論文の執筆にあって助言をいただいた（株）KDDI 研究所の竹森敬祐氏に感謝する．

参 考 文 献

1) Hickman, K.: SSL 2.0 Protocol Specification (1995). Available at: http://www.netscape.com/eng/security/SSL_2.html
2) Freier, A., Karlton, P. and Kocher, P.: The SSL Protocol Version 3.0 (1996). Avaiable at: http://home.netscape.com/eng/ssl3/
3) Dierks, T. and Allen, C.: The TLS Protocol Version 1.0, RFC 2246 (1999).
4) Web Application Firewall Evaluation Criteria (2006). http://www.webappsec.org/projects/wafec/
5) Hintz, A.: Fingerprinting Websites Using Traffic Analysis, *Workshop on Privacy Enhancing Technologies* (2002).
6) Bissias, G., Liberatore, M., Jensen, D. and Levine, B.: Privacy Vulnerabilities in Encrypted HTTP Streams, *Workshop on Privacy Enhancing Technologies* (2005).
7) Liberatore, M. and Levine, B.: Inferring the Source of Encrypted HTTP Connections, *ACM Conference on Computer and Communications Security* (2006).
8) Sun, Q., Simon, D., Wang, Y., Russell, W., Padmanabhan, V. and Qiu, L.: Statistical Identification of Encrypted Web Browsing Traffic, *IEEE Symposium on Security and Privacy* (2002).
9) Cheng, H. and Avnur, R.: Traffic Analysis of SSL Encrypted Web Browsing (1998). Available at: http://www.cs.berkeley.edu/~daw/teaching/cs261-f98/projects/final-reports/ronathan-heyning.ps
10) Mistry, S. and Raman, B.: Traffic Analysis of SSL-Encrypted Web Browsing (1998). Available at: http://bmrc.berkeley.edu/people/shailen/Classes/SecurityFall98/paper.ps
11) Lippmann, R., Haines, J., Fried, D., Korba, J. and Das, K.: The 1999 DARPA Off-line Intrusion Detection Evaluation, *Computer Networks*, 34 (2000).
12) DARPA Intrusion Detection Evaluation. http://www.ll.mit.edu/SST/ideval/
13) Wagner, D. and Schneier, B.: Analysis of the SSL 3.0 Protocol, *2nd USENIX Workshop on Electronic Commerce* (1996).
14) SafeWeb (2002). http://www.safeweb.com
15) DMOZ Open Directory Project (1998). http://dmoz.org
16) Danezis, G.: Traffic Analysis of the HTTP Protocol over TLS (2002). Available at: http://homes.esat.kuleuven.be/~gdanezis/TLSanon.pdf
17) Kruegel, C. and Vigna, G.: Anomaly Detection of Web-based Attacks, *ACM Conference on Computer and Communications Security* (2003).
18) Vigna, G., Robertson, W., Kher, V. and Kemmerer, R.: A Stateful Intrusion Detection System for World-Wide Web Servers, *Annual Computer Security Applications Conference* (2003).
19) Kim, H., Cho, S., Seo, J., Lee, Y. and Cha, S.: Use of Support Vector Machine (SVM) in Detecting Anomalous Web Usage Patterns, *Symposium on Information and Communications Technology* (2004).
20) Konno, T. and Tateoka, M.: Accuracy Improvement of Anomaly-Based Intrusion Detection System Using Taguchi Method, *Symposium on Applications and the Internet Workshops* (2005).
21) Estevez-Tapiador, J., Garcia-Teodoro, P. and Diaz-Verdejo, J.: Detection of Web-Based Attacks Through Markovian Protocol Parsing, *IEEE Symposium on Computers and Communications* (2005).
22) Robertson, W., Vigna, G., Kruegel, C. and Kemmerer, R.A.: Using Generalization and Characterization Techniques in the Anomaly-based Detection of Web Attacks, *Network and Distributed System Security Symposium* (2006).
23) Vigna, G., Robertson, W. and Balzarotti, D.: Testing Network-based Intrusion Detection Signatures Using Mutant Exploits, *ACM Conference on Computer and Communications Security* (2004).
24) Mahoney, M. and Chan, P.: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks, *ACM SIGKDD* (2002).
25) Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H. and Zhou, S.: Specification-based Anomaly Detection: A New Approach for Detecting Network Intrusions, *ACM Conference on Computer and Commu-*

nications Security (2002).

(平成 19 年 6 月 12 日受付)
(平成 19 年 12 月 4 日採録)



山田　明

2001 年神戸大学大学院自然科学研究科電気電子工学専攻博士前期過程修了．同年ケイディディアイ（株）入社．現在，（株）KDDI 研究所ネットワークセキュリティグループ研究主査．2007 年東北大学工学部情報科学研究科博士後期課程入学．暗号プロトコル，インターネットセキュリティの研究に従事．電子情報通信学会．ACM の各会員．

三宅　優（正会員）

1990 年慶應義塾大学大学院理工学研究科電気工学専攻前期博士課程修了．同年国際電信電話（株）入社．現在，（株）KDDI 研究所ネットワークセキュリティグループリーダー．高速通信プロトコルの実装，インターネットアクセス，インターネットセキュリティの研究に従事．1989 年度電気・電子情報学術振興財団猪瀬学術奨励賞，1995 年度情報処理学会学術奨励賞受賞．電子情報通信学会，情報処理学会の各会員．

寺邊　正大

1995 年京都大学大学院工学研究科修士課程修了．同年（株）三菱総合研究所入社．現在，科学・安全政策研究本部主任研究員．2001 年大阪大学大学院工学研究科博士後期課程修了．博士（工学）．2006 年から東北大学大学院情報科学研究科客員助教授を併任，2007 年客員准教授．現在に至る．データマイニング，エージェントベースシミュレーションの基礎・応用研究，および安全知識マネジメントシステム，リスクマネジメントシステムの構築等の業務に従事．1999 年度計測自動制御学会学術奨励賞受賞．人工知能学会，計測自動制御学会，AAAI，ACM の各会員．

橋本　和夫（正会員）

1979 年東北大大学院修士課程修了．同年国際電信電話（株）入社．1986 年ブラウン大学計算機科学科修士課程修了，2001 年東北大学大学院情報科学博士後期課程修了．博士（工学）．2001 年から 2005 年まで KDDI 米国研究所所長，2006 年より東北大学大学院情報科学研究科教授．現在に至る．機械翻訳，エキスパートシステム等の人工知能技術の通信への応用研究に従事．現在，ネットワークセキュリティ，データマイニング，マルチメディア情報検索を統合する Web コミュニケーションの研究を推進．1999 年人工知能学会研究奨励賞，2001 年電子情報通信学会論文賞，2002 年電波功績賞受賞．電子通信情報学会，人工知能学会，AAAI，IEEE の各会員．