学習データを自動生成する未知攻撃検知システム

山田 明[†] 三宅 優[†]
竹森敬祐[†] 田中俊昭[†]

Intrusion Detection System（IDS）には，シグネチャと呼ばれる既知攻撃のパターンファイルを用いて攻撃を検知する方式や，機械学習によって得られたプロファイルを用いて攻撃を検知する方式などが提案されており，前者は実用的であるがシグネチャに存在しない未知の攻撃を検知できない欠点があり，後者は学習データの生成が難しい問題がある．そこで本論文では，シグネチャによって既知の攻撃を検知しながら，その結果を機械学習することでシグネチャに登録されていない未知攻撃を検知するハイブリッド型 IDS を提案する．機械学習のための学習データは，シグネチャによる判定結果を基にトラヒックデータに対して攻撃の有無をラベル付けすることで自動生成している．提案システムについて，HTTP を対象としたプロトタイプの設計を行い実装する．そして，提案システムの未知攻撃の検知能力を正しく評価するために，従来から用いられている評価用データ，脆弱性監査ツールで生成したデータ，企業 LAN ゲートウェイから収集したデータの 3 種類を用いることとし，未知の攻撃が確実に含まれるような加工をして評価を行う．その結果，提案システムは機械学習に十分適用できる学習データを自動生成でき，未知の攻撃の多くを検知できることを示す．

Machine Learning Based IDS with Automatic Training Data Generation

Akira Yamada,[†] Yutaka Miyake,[†] Keisuke Takemori[†]
and Toshiaki Tanaka[†]

Although many intrusion detection systems based on learning algorithms have been proposed to detect unknown attacks or variants of known attacks, most systems require sophisticated training data for supervised learning. Because it is not easy to prepare the training data, the anomaly detection systems are not widely used in the practical environment. On the other hand, misuse detection systems that use signatures to detect attacks are deployed widely. However, they are not able to detect unknown attacks or variants of known attacks. So we have proposed a new anomaly detection system, which detects the variants of known attacks without preparing the training data. In this system, we use outputs of signature-based conventional IDS to generate the training data for anomaly detection. This system identifies novel features of attacks, and generates generalized signatures from the output of IDS to detect the variant attacks. We conducted experiments on the prototype system with three types of traffic data, 1999 DARPA IDS Evaluation Data, attacks by vulnerability scanner and actual traffic. The results show that our scheme can detect the variants of attacks efficiently, which cannot be detected by the conventional IDS.

1. 序 論

近年，サイバーテロ，ネットワーク犯罪の増加にともない，ネットワーク上の不正なトラヒックを検知する侵入検知システム（IDS: Intrusion Detection System）が注目を集めている．IDS には，シグネチャと呼ばれる既知攻撃のパターンファイルを用いて攻撃を検知する方式や，機械学習によって構成される攻撃モデルを用いて攻撃を検知する方式などが提案されている．シグネチャ型 IDS[3),6),16),17),20)] は実用的であるが未知の攻撃を検知できない欠点があり，機械学習型 IDS は未知の攻撃を検知できる可能性はあるが検知率の低さと学習データの生成における困難性から実用的ではない．ここで，未知の攻撃とは，シグネチャ型 IDS におけるシグネチャに登録されていない攻撃もしくは機械学習型 IDS における学習データに存在しない攻撃とする．機械学習アルゴリズムは教師あり学習[1),10)] と教師なし学習[8),9),18),21)] に分類でき，教師あり学習を行う方式は，教師なし学習を行う方式に比べて検知率が高く，注目を集めているが，教師情報を含む学習データを用意する必要がある．この学習データとして

† 株式会社 KDDI 研究所
KDDI R&D Laboratories Inc.

頻繁に用いられている DARPA IDS evaluation data 1998，1999[12),13)] は，データ中の攻撃発生時刻や攻撃種類などの教師情報を Web サイト[11)] で公開している．しかし，DARPA データだけを用いて学習を行うと，DARPA データとは異なる性質を持つネットワーク環境において効果的な攻撃モデルを構成できないという問題がある[15)]．一方，各々のネットワーク環境から教師情報を含む学習データを得るためには，IDS の運用者が手動で作成する必要があり煩雑である．そこで，ハニーポット[19)] を利用して学習データを収集する方式[7)] が提案されている．しかし，この方式は検知の自動化を実現していない．

そこで本論文では，シグネチャによって攻撃を検知しながら，その結果を機械学習することで未知攻撃を検知するハイブリッド型 IDS を提案する．機械学習のための学習データは，シグネチャによる判定結果を基に，トラヒックデータに対して攻撃の有無をラベル付けすることで自動生成している．提案システムについて，HTTP を対象としたプロトタイプの設計を行い実装する．そして，提案システムの未知攻撃の検知能力を正しく評価するために，従来から用いられている評価用データ，脆弱性監査ツールで生成したデータ，企業 LAN ゲートウェイから収集するデータの 3 種類を用いることとし，未知の攻撃が確実に含まれるような加工をして評価を行う．その結果，提案システムは機械学習に十分適用できる学習データを自動生成することができ，その多くを検知できることを示す．

以降，2 章に従来からの IDS とその問題点を，3 章に機械学習型 IDS の評価における問題点をそれぞれ説明する．次に，4 章に 2 章の問題点を解決する提案システム，5 章に提案システムのプロトタイプの実装方法を説明する．さらに，6 章に 3 章の問題点を解決する評価方法を説明し，7 章にその評価結果を示す．最後に，8 章に結論を示すという構成である．

2. 従来の IDS とその問題点

本章では 2.1，2.2 節においてシグネチャ型 IDS と機械学習型 IDS について説明し，2.3 節において問題点をまとめる．

2.1 シグネチャ型 IDS

シグネチャ型 IDS はシグネチャと呼ばれる攻撃の特徴を収録したパターンファイルの情報を用いて攻撃を検知する．この方式はシグネチャに収録している各攻撃のパターンと検査対象のパケットを照合し，パターンと一致するパケットを攻撃と判定している．CodeRed II ワームの *ISAPI .ida attempt* 攻撃におけるパケットのペイロードとシグネチャ型 IDS である Snort[3)] のシグネチャを**図 1** に示す．このシグネチャは HTTP をデコードした後に URI に *ida?* が含まれている場合に攻撃と判定している．

```
..[.Fl..R..!..E.....@.g.Dq."5...[....P...haG‘
fP...1...GET /default.ida?XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
alert tcp any any -> any 80 (msg:"WEB-IIS
ISAPI .ida  attempt"; flow:to_server,
established; uricontent:".ida?"; nocase;
...
```

図 1 CodeRed II のパケットペイロードとシグネチャ
Fig. 1 Packet payload and signature of codered II warm.

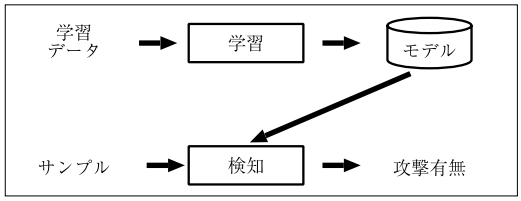

図 2 機械学習型 IDS の流れ
Fig. 2 A flow of machine learning based IDS.

2.2 機械学習型 IDS

機械学習型 IDS は学習と検知の 2 つ手順を経て攻撃を検知している（**図 2**）．学習の手順では過去の攻撃が含まれている学習データから攻撃モデルを構成し，検知の手順では攻撃モデルを利用してあるサンプルの攻撃有無を判定する．機械学習型 IDS によって構成される攻撃モデルは，特定の攻撃を対象とするものではないため，学習データに存在しない攻撃を検知できる可能性がある．

機械学習アルゴリズムは教師あり学習と教師なし学習に分類でき，教師あり学習は比較的検知率が高いが教師情報を含む学習データを必要とする．教師あり学習の手順を以下に示す．

学習 （1） 学習データから各パケットやトラヒックの特徴を表す変数の集合を特徴ベクトルとして抽出する．
（2） 教師情報を利用して，各特徴ベクトルに攻撃ありもしくは攻撃なしを割り当てる．攻撃ありの代わりに攻撃の種類を割り当ててもよい．
（3） 特徴ベクトルに学習アルゴリズムを適用することにより攻撃モデルを構成する．

検知 （1） サンプルから特徴ベクトルを抽出する．
（2） 攻撃モデルを利用して特徴ベクトルの攻撃

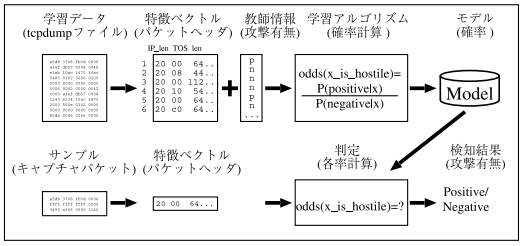

図 3 Mahoney らの方式の例
Fig. 3 An example of components that mahoney, et al.'s system employs.

の有無を判定する．

例として，**図 3** に Mahoney らの方式[14] を示す．Mahoney らの方式では，学習データが tcpdump ファイル，特徴ベクトルがパケットヘッダの各変数であり，機械学習アルゴリズムが確率計算である．確率計算では，ある事象が与えられるとき，その事象が攻撃である確率を計算している．そして，その攻撃モデルによる確率計算からサンプルの攻撃有無を判定する．

2.3 問 題 点

検知誤りの問題 検知誤りは正常パケットを攻撃と判定する FP（False Positive）と，攻撃を検知できない FN（False Negative）の 2 種類ある．シグネチャ型 IDS は，正常なトラヒックの中にシグネチャに収録しているパターンと一致する箇所が偶然含まれているとFP を発生する．たとえば，*ISAPI .ida attempt* 攻撃のシグネチャは，*ida?* を含むが攻撃を含まない URI に対して FP を発生する．ただし，攻撃が既知である場合シグネチャ型 IDS は機械学習型 IDS より高い検知率を示す．機械学習型 IDS は学習における変数を調整することで検知誤りを減らすことができるが，検知誤りの減少にともない未知の攻撃を検知する能力も低下する．また，シグネチャ型 IDS では，FP が発生したときにシグネチャを解析することにより原因を調査できるが，ニューラルネットなどの機械学習アルゴリズムを用いる機械学習型 IDS では，解析アルゴリズムの複雑さのため原因を調査することが困難である．

未知攻撃の問題 シグネチャ型 IDS はシグネチャに収録していない攻撃を検知することは困難である．したがって，新しい攻撃に対応するためにシグネチャを更新する必要がある．しかし，シグネチャ型 IDS は，そのシグネチャ更新周期が攻撃とその亜種の出現周期より遅いため，未知の攻撃を受ける可能性がある．一方，機械学習型は学習データから攻撃モデルを構成して攻撃を検知するが，学習データに含まれていない攻撃も検知する能力がある．ここで，シグネチャ型 IDS における未知攻撃とはシグネチャに収録されていない攻撃とし，機械学習型 IDS における未知攻撃とは学習データに含まれていない攻撃とする．

教師情報の問題 教師あり学習は教師なし学習に比べ検知率が高いが，教師情報を含む学習データを必要とする．IDS を評価するための代表的な学習データとして DARPA IDS evaluation data 1998，1999[12),13)] があるが，攻撃発生時刻や攻撃種類などの教師情報は Web サイト[11)] に公開されている．しかし，DARPA データだけを用いて学習を行うことは，実環境において効果的な攻撃モデルを構成できない問題がある[15)]．また，実環境において攻撃発生時刻や攻撃種別などの教師情報を含む学習データを得ることは困難である．

つまり，既存の IDS には以下の問題が存在する．

問題点 1-1：機械学習型 IDS は FP，FN が多い，

問題点 1-2：シグネチャ型 IDS は未知攻撃を検知できない．

問題点 1-3：教師あり学習は教師情報を必要とし，実環境では入手が困難である．

3. 機械学習型 IDS における評価方法と問題点

本章では 3.1 節において機械学習型 IDS における従来の評価方法を説明し，3.2 節においてその問題点についてまとめる．

3.1 機械学習型 IDS における従来の評価方法

機械学習型 IDS の評価は学習データと検査データの 2 種類のデータを用いる．はじめに学習データにより学習を行い，攻撃モデルを構成する．次に検査データにより攻撃モデルの評価を行う．ここで，検査データは学習データに含まれる攻撃だけでなく未知攻撃が含まれるように構成する．評価は検査データにおける既知攻撃，未知攻撃の検知数によって行う．従来の機械学習型 IDS は，以下の 2 種類の評価用データによって評価される．

- 1998，1999 DARPA IDS Evaluation Data[12),13)]
- 実環境から収集するデータ

3.2 問 題 点

DARPA データ[13)] は学習データとして 1，2，3 週目，検査データとして 4，5 週目が用意されている．DARPA データは学習データに 54 種類の攻撃，検査データにその 54 種類の攻撃と新たな数種類の攻撃を含んでいる．学習データに含まれない攻撃を未知攻撃とするとき，未知攻撃として評価される攻撃は数種類だけであり，未知攻撃の数が少ない点において問題がある．したがって，機械学習型 IDS における従来の評価方法には以下の問題が存在する．

問題点 2-1：評価対象になる未知攻撃の数が少ない．

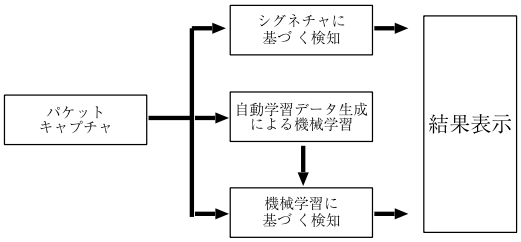

図 4 提案システム
Fig. 4 Proposed system.

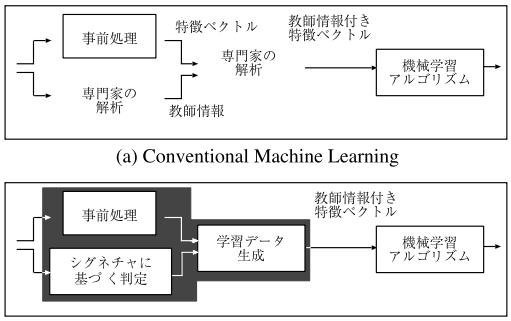

図 5 学習データ自動生成
Fig. 5 Automatic learning data generation.

問題点 2-2：評価用データが実環境と乖離する．

4. ハイブリッド型システムの提案

2 章の問題を解決するためにシグネチャによって攻撃を検知しながら，その結果を機械学習することで未知攻撃を検知するハイブリッド型 IDS を提案する．機械学習のための学習データは，シグネチャの判定結果を基に自動生成する．本章では提案システムの構成と学習データ自動生成による機械学習について説明する．

4.1 提案システムの構成

提案システムの構成を**図 4** に示す．問題点 1-1，1-2 を解決するために提案システムは既知攻撃をシグネチャにより検知し未知攻撃を機械学習により検知する．問題点 1-3 を解決するために機械学習のための学習データを自動生成する．

4.2 学習データ自動生成による機械学習

提案システムは教師情報つき学習データを自動的に生成することにより教師あり学習を実現する．従来の機械学習手順と学習データ自動生成による機械学習手順の比較を**図 5** に示す．機械学習を行うためには tcpdump ファイルを評価する単位で分割し，それぞれを表す変数の集合として特徴ベクトルを抽出し，それぞれの特徴ベクトルを攻撃に該当する特徴ベクトルと非攻撃に該当する特徴ベクトルに分離する必要がある．機械学習アルゴリズムは攻撃・非攻撃に分離された特徴ベクトルを用いて，検査データから抽出される特徴ベクトルを判定するルールを出力する．通常の機械学習では専門家が時間をかけて解析し，tcpdump ファイルのなかに含まれる攻撃を発見し，攻撃を各特徴ベクトルに割り当てることにより，攻撃・非攻撃に該当する特徴ベクトルを分離する．提案方式は，専門家による攻撃発見および特徴ベクトルへの割当てを自動的に行う方式である．ここで，攻撃発見はシグネチャに基づく判定を利用し，特徴ベクトルへの割当ては，各特徴ベクトルに IP アドレスなどに基づく ID をあらかじめ付与することにより行う．学習データ自動生成の手順を以下に示す．

(1) 事前処理において，従来の機械学習に用いられる特徴ベクトルに加えて，$ID_{vi}(i = 0, 1 \ldots)$ を割り当てる．ID は時刻，送信元 IP，宛先 IP，送信元ポート，宛先ポートから生成する．

$$ID_{vi} = \{Time_{vi}|scrIP_{vi}|dstIP_{vi}| srcPort_{vi}|dstPort_{vi}\}$$

(2) シグネチャに基づく判定において，検知結果に $ID_{aj}(j = 0, 1 \ldots)$ を割り当てる．ID は時刻，送信元 IP，宛先 IP，送信元ポート，宛先ポートから生成する．

$$ID_{aj} = \{Time_{aj}|scrIP_{aj}|dstIP_{aj}| srcPort_{aj}|dstPort_{aj}\}$$

(3) 学習データの生成において，ID_{vi} と ID_{aj} を比較することにより特徴ベクトルに検知結果を割り当てる．ここで，ID_{vi} と ID_{aj} の時間精度が異なる場合を考慮し，時刻差 $\pm\Delta T$ の範囲において割当てを行う．対応する ID_{ai} がない ID_{vj} に攻撃なしと割り当てる．

4.3 決定木による学習

提案方式は機械学習アルゴリズムとして決定木[2] を用いる．代表的な機械学習アルゴリズムとして，決定木，ニューラルネット，サポートベクターマシンがあるが，このなかで決定木の学習結果は解析することが容易である．決定木の学習結果は攻撃検知のため用いられるいくつかのルールであり，それぞれのルールは個別に解釈が可能なため，あるトラヒックが決定木によって攻撃と判定されたとき，判定に用いられたルールを調べることによりなぜ攻撃と判断されたのかを解析できる．木構造において各ノードは分割ルール，枝は分割結果，葉はクラスラベルあるいはクラス分布を示す．決定木を構成することはある基準において最良の分割変数と分割値を求めることである．一般に分割

表 1 決定木における各変数
Table 1 Parameters for learning algorithm.

分岐	Gini 指数
重み	クラスごと
各ノードの最大枝数	2
末端ノードの最大要素数	0
木の最大の深さ	∞

基準に用いる不均衡度関数（impurity function）は情報利得，情報利得比，Gini 指数などがある．今回のプロトタイプシステムでは Gini 指数を用いる．ここで，Gini 指数とは，ノードにおいて最も大きなクラスを見つけ出し，このクラスを他のクラスから分離する分割ルールを求める関数である．この性質は攻撃を正常トラヒックから分離するルールを見つけることに適している．ここで，S，C_j を分割前におけるノードおよびクラス，$|S|$，$|C_j|$ を各々のデータ数，A を分割に用いる変数とするとき，分割前の Gini 指数 $gini(S)$ は以下のように定義される．

$$gini(S) = 1 - \sum_j P(S, C_j)^2$$

$$P(S, C_j) = \frac{|C_j|}{|S|}$$

また，分割後の Gini 指数 $giniSplit(S, A)$ は S_j を分割後におけるノード，$|S_j|$ をデータ数とするとき以下のようになる．

$$giniSplit(S, A) = \sum_j \frac{|S_j|}{|S|} \times gini(S)$$

よって，分割による改善度は以下のようになる．

$$gain(S, A) = gini(S) - giniSplit(S, A)$$

決定木を実行するときに用いる変数を**表 1** に示す．決定木は頻度が少ないクラスを分離するルールを生成することが困難であるため，ルートノードにおいてすべてのクラスが等しい頻度となるように重み付けする．つまり，学習データにおいて各攻撃のトラヒックは正常トラヒックに比べて少ないため，それぞれの攻撃トラヒックに重みを付けて，正常トラヒックと等しい頻度となるように調整する．

5. 実　　装

提案方式に基づきプロトタイプを実装した．本章ではプロトタイプの構成要素および実装方法を説明する．

5.1 プロトタイプの構成要素

表 2 にプロトタイプの構成要素を示す．事前処理として，HTTP リクエストヘッダから特徴ベクトルを抽出し，シグネチャ型 IDS として Snort [3] を用い，機械学習アルゴリズムとして Gini 係数による決定木を用い，GUI は Gtk により作成した．本プロトタイプでは HTTP を対象とするが，事前処理の変更により様々なプロトコルに適用できる．

表 2 プロトタイプの構成要素
Table 2 System components of prototype.

事前処理	HTTP リクエストヘッダ
シグネチャ型 IDS	Snort 1.9.1
機械学習アルゴリズム	決定木，Gini 指数
GUI	Gtk

```
GET /welcome.htm HTTP/1.0\r\n
Connection: Keep-Alive\r\n
User-Agent: Mozilla/4.08 [en] (WinNT; I)\r\n
Host: www.eyrie.af.mil\r\n
Accept: image/gif, image/x-xbitmap, */*\r\n
Accept-Encoding: gzip\r\n
Accept-Language: en\r\n
Accept-Charset: iso-8859-1,*,utf-8\r\n
\r\n
Data (15 bytes)
```

(a) An Example of HTTP Request

ID									
08/12/04-20:24:08.881860 194.27.251.21 1931 172.16.114.50 80									
Request-line	Connection	User-Agent	Host	Accept	Accept-Encoding	Accept-Language	...		
29	14	32	20	35	9	6	...		

(b) An Example of Feature vector for HTTP Request

図 6 HTTP リクエスト
Fig. 6 HTTP request.

5.2 HTTP における事前処理

IP フラグメンテーション，TCP ストリームを再構成した後に HTTP の解析を行う．HTTP リクエスト[5] の例を**図 6** (a) に示す．また，特徴ベクトルの例を図 6 (b) に示す．提案システムの特徴ベクトルは以下の値とする．

(1) ID として，時間，IP アドレス，ポート番号．
(2) Request-Line のサイズ [bytes].
(3) general-header, request-header, entity-header の各 field-name に対する field-value のサイズ [bytes]．ただし，ヘッダ中に該当する field-name がないときは 0 とする．また field-value が整数値である場合はその値とする．学習データに存在する field-name すべてを対象とする．
(4) message-body もしくは上記ヘッダ以下にある ASCII デコードできないデータのサイズ [bytes].
(5) HTTP リクエストの総サイズ [bytes].

5.3 GUI

図 7 にプロトタイプシステムのスクリーンショットを示す．GUI はシグネチャによる検知結果と機械学習による検知結果を表示し機械学習の各種変数を変更できる．

図 7 グラフィカルユーザインタフェース
Fig. 7 Graphical user interface.

6. 評価データの作成

本章では，3 章の問題点を解決するために，評価用データを再構成する方法を示す．評価用データについて説明し，次に各々のデータの再構成方法を説明する．

6.1 本論文における評価方式

本論文において，未知攻撃の検知能力を評価するために 3 種類のデータを用いて評価を行う．3.2 節における問題点 2-1 解決するために，DARPA データおよび脆弱性スキャナ Nessus [4] のデータの各々を再構成する．また，DARPA や Nessus のデータはベンチマークの意味合いが強いため，問題点 2-2 を解決するため，実際の LAN ゲートウェイで収集したトラヒックを利用して評価を行う．つまり，以下の 3 種類のデータを用いて評価を行う．

(i) 1999 DARPA IDS evaluation data.
(ii) Nessus [4] により生成されるデータ.
(iii) ある企業 LAN ゲートウェイにおいて収集するデータ.

6.2 DARPA IDS Evaluation Data

本論文では DARPA データの 4，5 週目の HTTP リクエストを利用して評価を行う．再構成として，学習データは DARPA データから 1 種類の攻撃を除いて構成し，検査データはその取り除いた攻撃を集めて構成する．この評価データの再構成により，すべての攻撃を学習データに存在しない未知攻撃として扱うことができる．つまり，従来の評価手法のすべての攻撃を未知攻撃として評価できないために攻撃数が少ないという問題点 2-1 は解決される．本プロトタイプは HTTP リクエストを対象にするため，7 種類の攻撃 *apache2*, *back*, *crashiis*, *mscan*, *ntinfoscan*, *phf*, *ps* を対象とする．よって，学習データと検査データを 7 組構成し，各々のデータに対して 7 回の評価を行う．評価用データの詳細を**表 3** に示す．

表 3 DARPA データに含まれる HTTP リクエスト
Table 3 HTTP request included in DARPA data.

攻撃	小計
apache2	3,061
back	167
crashiis	8%
hline mscan	39
phf	4
ps	2
通常	431,975
合計	436,100

表 4 Nessus による HTTP リクエスト
Table 4 HTTP request generated by nessus.

	攻撃	小計
検知可能	WEB-MISC	696
	WEB-CGI	444
	WEB-IIS	297
	WEB-PHP	30
	WEB-FRONTPAGE	16
	WEB-COLDFUSION	12
検知不可能		2,728
合計		4,723

6.3 脆弱性スキャナ Nessus

本評価では脆弱性スキャナ Nessus [4] により発生させる攻撃を利用して評価を行う．学習データは Snort によって検知できる攻撃とし，検査データは Snort によって検知できない攻撃として再構成する．ただし，決定木による機械学習では，攻撃と判定される特徴ベクトルと非攻撃と判定される特徴ベクトルを入力として，それらのベクトルを分離するためのルールを出力する．正常と判断される特徴ベクトルを得るために企業 LAN ゲートウェイにおいて収集するデータを利用する．ここで，LAN ゲートウェイにおいて収集するデータを 6.4 節で説明する．また，Snort のシグネチャはデフォルトのものを用いる．プロトタイプは HTTP リクエストのみを対象とするため，840 種類の攻撃，4,723 件の HTTP リクエストを対象とする．Snort は 1,495 件のリクエストを検知可能であり，2,728 件のリクエストを検知不可能である．Nessus による HTTP リクエストの詳細を**表 4** に示す．

6.4 LAN ゲートウェイ

本評価ではある企業 LAN ゲートウェイから収集するデータを利用して評価を行う．ただし，この評価の目的は学習結果が実環境に適していることを確認することである．したがって，検査データを使用せず学習データのみを使用し，学習の結果構成される決定木と学習データ生成に使用するシグネチャを比較する．

表 6 評価結果
Table 6 Evaluation summary.

		シグネチャに基づく検知				機械学習に基づく検知			
		TP	TN	FP	FN	TP	TN	FP	FN
(i)DARPA	既知攻撃 (apache2)	3,935	429,104	0	0	3,732	429,104	203	0
	未知攻撃 (apache2)	0	0	0	3,061	3,061	0	0	0
	未知攻撃 (back)	0	0	0	167	0	0	0	167
	未知攻撃 (crashiis)	0	0	0	8	0	0	0	8
	未知攻撃 (mscan)	0	0	0	894	0	0	0	894
	未知攻撃 (ntinfoscan)	0	0	0	39	39	0	0	0
	未知攻撃 (phf)	0	0	0	4	4	0	0	0
	未知攻撃 (ps)	0	0	0	2	0	0	0	2
(ii) Nessus	既知攻撃	1,494	0	0	0	793	0	0	701
	未知攻撃	0	0	0	2,727	2,687	0	0	40
(iii) LAN Gateway	既知攻撃	51	42,435	98	0	33	42,531	2	18

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

表 5 ある企業 LAN ゲートウェイにおける HTTP リクエスト.
Table 5 HTTP request captured at a corporate LAN gateway.

	攻撃	小計
CodeRed	WEB-IIS	19
Nimda	WEB-IIS	32
その他	WEB-CGI	81
	WEB-MISC	17
	WEB-PHP	1
通常		42,435
合計		42,584

学習データには 4,258 件の HTTP リクエストが含まれ，149 件のリクエストが Snort で攻撃と判定される．LAN ゲートウェイから収集したデータの詳細を**表 5** に示す．専門家による詳細解析の結果，Nimda ワームのアクセスが 32 件，CodeRed II ワームのアクセスが 19 件発見された．そして，その他の検知結果は Snort による検知誤りである．

7. 評価結果

本章では結果概要を示し，各々の結果について考察する．

7.1 結果概要

評価結果を**表 6** に示す．提案方式は機械学習に基づく検知によっていくつかの未知攻撃を検知できている．つまり，シグネチャに基づく検知結果から，自動的に生成した学習データを用いてシグネチャに登録されていない未知攻撃を検知できるようになったことが分かる．また，提案方式はハイブリッド型であるため，シグネチャに基づく検知により既知攻撃も高い精度で検知できる．DARPA データでは 7 種類の攻撃のなかで 3 種類の未知攻撃を検知でき，Nessus によるデータではシグネチャに登録されていない 2,687 個の攻撃を検知できる．

7.2 DARPA IDS Evaluation Data

機械学習による検知は，*apache2* を *back*，*ntinfoscan* を *phf*，*phf* を *ntinfoscan* のように，亜種の攻撃として検知できる．しかし，4 種類の攻撃 *back*，*crashiis*，*mscan*，*ps* を検知できない．検知できる理由は，DARPA データに含まれる攻撃は HTTP ヘッダに攻撃の成功に直接影響はない特徴が含まれるためである．具体的には，*Host*，*Connect*，*User-Agent*，*Referer*，*Accept* が存在しない，もしくは特定の値であるという特徴がある．ここで，提案システムは学習データに含まれるすべての HTTP ヘッダの field-name から生成している．

7.3 脆弱性スキャナ Nessus

シグネチャによる検知は 2,727 件の HTTP リクエストを検知できないが，機械学習による検知は 2,727 件のリクエストの中で 2,687 件のリクエストを検知できた．この理由も DARPA データの結果と同様である．Nessus は異なる攻撃に同様の HTTP ヘッダを持つリクエストを生成するため，類似する特徴ベクトルを発生する．機械学習による検知はその Nessus の特徴を持つリクエストを攻撃と判定している．しかし，適切な攻撃種別に分類できないため検知誤りが 701 件発生してしまう．この評価によって，提案方式は Snort により検知できない，シグネチャに登録されていない未知攻撃を検知でき，また，攻撃ツール Nessus の特徴を抽出することもできた．

7.4 LAN ゲートウェイ

Nimda，CodeRed II の攻撃に対する決定木の分岐ルールと各々のシグネチャを**表 7**，**表 8** に示す．これらは同じ攻撃を検知するが，大きく異なる．これは，シグネチャは攻撃の特徴を記述しているのに対し

表 7 Nimda と CodeRed II に対する決定木分岐ルール
Table 7 Splitting rules for Nimda and CodeRed II.

攻撃	分岐ルール	説明
CodeRed II	$-\infty$<Connnection≤0.011, 0.073<Content-type≤33.0, 54.500<Request≤239.000	*Content-Type* の *t* が小文字である.
Nimda	0.011<Connnection≤5.000	*Connection* のスペルミス

表 8 Nimda と CodeRed II に対する Snort シグネチャ
Table 8 Snort signatures for Nimda and CodeRed II.

攻撃	シグネチャ
CodeRed II	alert tcp any any -> any 80 (msg:"WEB-IIS ISAPI .ida attempt"; flow:to_server,established; uricontent:".ida?"; nocase; reference:arachnids, 552; classtype:web-application-attack; reference:bugtraq, 1065; reference:cve,CAN-2000-0071; sid:1243; rev:8;)
Nimda	alert tcp any any -> any 80 (msg:"WEB-IIS cmd.exe access"; flow:to_server,established; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:5;)

て，分岐ルールはワームの特徴を記述しているためである．たとえば，Nimda ワームは field-name として *Content-Type* の代わりに *Content-type* を使用する．また，CodeRed II ワームは *Connection* の代わりに誤ったスペル *Connnection* を使う．Nimda および CodeRed に対するすべて亜種は同様の特徴を持つため，この学習結果によりすべての亜種攻撃は検知される．

7.5 考　　察

以上の評価結果から，提案システムは機械学習のための学習データをシグネチャ型 IDS の判定結果により自動生成でき，生成する学習データは機械学習に十分適用できることが分かる．特に，ワームや攻撃ツールの特徴を抽出することができるため，シグネチャが更新される以前の既知の攻撃に類似する攻撃を検知できることを確認した．また，シグネチャの更新により新しい攻撃に対応する学習データを生成でき，実際に表 7 に示すワームの新しい特徴を発見できることも判明した．たとえば，本システムの運用者がこれらの特徴のなかから適切な情報を抽出することで，**表 9** に示すシグネチャとして利用できるようになる．このシグネチャは CodeRed や Nimda の亜種攻撃を検知することを目的としているが，攻撃の本質的な特徴ではないため，FP を増やす可能性もある．FP 増加に関する評価は今後の課題である．

表 9 Nimda と CodeRed II の亜種ための新しいシグネチャ
Table 9 A new signature for Nimda and CodeRed II variant.

攻撃	新しいシグネチャ
CodeRed II	alert tcp any any -> any 80 (msg:"CodeRed Variants"; flow:established;content:"Content-type ";)
Nimda	alert tcp any any -> any 80 (msg:"Nimda Variants"; flow : established; content : "Connnection" ;)

8. 結　　論

本論文では，シグネチャによって攻撃を検知しながら，その結果を機械学習することでシグネチャに登録されていない未知攻撃を検知するハイブリッド型 IDS を提案した．また，機械学習型 IDS における学習データに存在しない未知攻撃の検知能力に注目した評価を行うための手法について検討し，評価用データを作成した．評価の結果，DARPA データにおいて学習データに含まれていない 7 種類の攻撃の中で 3 種類を，Nessus データにおいて Snort において検知できない 2,727 件の攻撃の中で 2,687 件の攻撃をそれぞれ検知できており，提案システムは機械学習に十分適用できる学習データを自動生成できることを示した．提案システムはワームや攻撃ツールなどの特徴を自動的に抽出できる性質があり，ある攻撃のシグネチャを登録しておくことで，その攻撃を応用したワームや攻撃ツールを検知できる効果がある．また，抽出される特徴にはシグネチャとして利用できるものが含まれていることが分かった．今後の課題として，機械学習により抽出される特徴をシグネチャとして自動的にフィードバックする機能について検討を進める．

参 考 文 献

1) Barbara, D., Wu, N. and Jajodia, S.: Detecting Novel Network Intrusions Using Bayes Estimators, *Proc. 1st SIAM International Conference on Data Mining* (*SDM-01*) (2001).
2) Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J.: *Classification and Regression Trees*, CRC Pr I Llc (1984).
3) Caswell, B. and Roesch, M.: Snort, The Open Source Network Intrusion Detection System. http://www.snort.org
4) Deraison, R.: Nessus. http://www.nessus.org
5) Fielding, R., Irvine, U., Gettys, J., Mogul, J., Compaq, Frystyk, H., Masinter, L., Xerox, Leach, P., Microsoft and Berners-Lee, T.: Hypertext Transfer Protocol — HTTP/1.1, rfc

2616. http://www.ietf.org/rfc/rfc2616.txt
6) Internet Security Systems, Inc.: realsecure. http://www.iss.net
7) Kreibich, C. and Crowcroft, J.: Honeycomb — Creating Intrusion Detection Signatures Using Honeypots, *Proc. 2nd Workshop on Hot Topics in Networks* (*HotNets-II*) (2003).
8) Kruegel, C., Toth, T. and Kirda, E.: Service Specific Anomaly Detection for Network Intrusion Detection, *Proc. 2002 ACM symposium on Applied computing table of contents* (2004).
9) Kruegel, C. and Vigna, G.: Anomaly Detection of Web-based Attacks, *Proc. 10th ACM conference on Computer and communication security* (*CCS 2003*) (2003).
10) Lee, W. and Stolfo, S.J.: A framework for constructing features and models for intrusion detection systems, *ACM Trans. Information and System Security* (*TISSEC*), Vol.3, No.4, pp.227–261 (2000).
11) Lincoln Laboratory, Massachusetts Institute of Technology: LINCOLN LABORATORY. http://www.ll.mit.edu/IST/ideval/
12) Lippmann, R., Cunningham, R.K., Fried, D.J., Graf, I., Kendall, K.R., Webster, S.E. and Zissman, M.A.: Results of the DARPA 1998 Offline Intrusion Detection Evaluation, *Recent Advances in Intrusion Detection* (1999).
13) Lippmann, R., Haines, J.W., Fried, D.J., Korba, J. and Das, K.: The 1999 DARPA offline intrusion detection evaluation. Computer Networks, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.34, No.4, pp.579–595 (2000).
14) Mahoney, M.V. and Chan, P.K.: Learning nonstationary models of normal network traffic for detecting novel attacks, *Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002).
15) Mahoney, M.V. and Chan, P.K.: An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, *Proc. Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003*, Pittsburgh, PA, USA, September 8-10, 2003 (2003).
16) NFR Security, Inc.: Network flight recorder. http://www.nfr.com
17) Paxson, V.: Bro: a system for detecting network intruders in real-time, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.31, No.23–24, pp.2435–2463 (1999).
18) Porras, P.A. and Neumann, P.G.: EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances, *Proc. 20th National Information Systems Security Conference* (1997).
19) Provos, N.: Honeyd — A Virtual Honeypot Daemon, *Proc. 10th DFN-CERT Workshop* (2003).
20) Roesch, M.: Snort — lightweight intrusion detection for networks, *Proc. 13th Conference on Systems Administration* (*LISA-99*) (1999).
21) Staniford, S., Hoagland, J.A. and McAlerney, J.M.: Practical Automated Detection of Stealthy Portscans, *Journal of Computer Security*, Vol.10, No.1/2, pp.105–136 (2002).

付 録

A.1 DARPA データに対する分岐ルールとシグネチャ

LINCOLIN 研究室の Web サイト[11] の情報を利用して Snort のためにシグネチャを作成した（**表 10**）．このシグネチャは DARPA データのみに有効なシグネチャとなっており，IP アドレスを含んでいるが，本論文の評価は誤り率の検証ではなく，学習データの生成に関する評価であるため，便宜的に IP アドレスを含むシグネチャを使用している．このシグネチャを用いることで，評価に用いた 7 種類の攻撃のすべてを検知できる．DARPA データを用いる学習によって構成される決定木の分岐ルールを**表 11** に，決定木の構成例を**図 8** に示す．この決定木は DARPA データによる評価において *apache2* 攻撃を除いた学習データにより構成したものである．

(平成 16 年 11 月 26 日受付)
(平成 17 年 6 月 9 日採録)

表 10 DARPA データに対する Snort シグネチャ
Table 10 Snort signatures for DARPA data.

Attack	Signature
apache2	alert tcp any any -> any 80 (msg:"apache2"; flow:established; content:"|47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 73 69 6f 75 78 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 73 69 6f 75 78 0d 0a|";)
apache2	alert tcp any any -> any 80 (msg:"apache2"; flow:established; content:"|55 53 45 52 20 6f 74 74 6f 62 0a 50 41 53 53 20 49 72 53 5a 71 49 73 65 0a 53 54 41 54 0a 52 45 54 52 20 31 0a 44 45 4c 45 20 31 0a 51 55 49 54 0a 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 73 69 6f 75 78 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 73 69 6f 75 78 0d 0a|";)
back	alert tcp any any -> any 80 (msg:"back"; flow:established; content:"GET ////";)
back	alert tcp any any -> any 80 (msg:"back"; flow:established; content:"GET /cgi-bin////";)
crashiis	alert tcp any any -> any 80 (msg:"crashiis"; flow:established; content:"../..";)
mscan	alert tcp 207.136.86.223 any -> any 80 (msg:"mscan"; flow:established; content:"GET /cgi-bin/phf";)
mscan	alert tcp 207.136.86.223 any -> any 80 (msg:"mscan"; flow:established; content:"GET /cgi-bin/test-cgi";)
mscan	alert tcp 207.136.86.223 any -> any 80 (msg:"mscan"; flow:established; content:"GET /cgi-bin/handler";)
ntinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"HEAD / HTTP/1.0";)
ntinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /*.idc HTTP/1.0";)
ntinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /cgi-bin/ HTTP/1.0";)
ntinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /scripts/ HTTP/1.0";)
ntinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /cgi-bin/perl.exe?-v HTTP/1.0";)
ntfinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /scripts/perl.exe?-v HTTP/1.0";)
ntfinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /scripts /tools /newdsn.exe HTTP/1.0";)
ntfinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content: "GET /_vti_bin/fpcount.exe?" "Page=default.htm"; content:"Image=3"; content:"Digits=15 HTTP/1.0";)
ntfinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /scripts/*%0a.pl HTTP/1.0";)
ntfinfoscan	alert tcp 206.48.44.18 any -> 172.16.112.100 80 (msg:"ntinfo"; flow:established; content:"GET /samples/search/queryhit.htm HTTP/1.0";)
phf	alert tcp any any -> any 80 (msg:"phf"; flow:established; content:"phf?";)
ps	alert tcp 172.16.112.50 any -> 209.154.98.104 80 (msg:"ps"; flow:established; content:"tester.tar";)

表 11 DARPA データに対する分岐ルール
Table 11 Splitting rules for DARPA data.

Attack	Splitting Rules
apache2	609.938<total<=1460.000, -∞<method<=14.008
back	609.938<total<=1460.000, 14.008<method<=239.000
crashiis	-∞<total<=609.938, -∞<data<=1.804, -∞<User-Agent<=23.030, -∞<Host<=7.013, -∞<method<=14.008
mscan	-∞<total<=609.938, 1.804<data<=1443.000
ntinfoscan, phf	∞<total<=609.938, -∞<data<=1.804, -∞<User-Agent<=23.030, -∞<Host<=7.013, 14.008<method<=239.000
ps	-∞<total<=609.938, -∞<data<=1.804, 23.030<User-Agent<=47.000, 0.013<Connection<=10.000, -∞ < method < = 26.030
normal	-∞<total<=609.938, -∞<data<=1.804, -∞<User-Agent<=23.030, 7.013<Host<=34.000
normal	-∞<total<=609.938, -∞<data<=1.804, 23.030<User-Agent<=47.000, -∞<Connection<=0.013
normal	-∞<total<=609.938, -∞<data<=1.804, 23.030<User-Agent<=47.000, 0.013<Connection<=10.000, 26.030 < method < = 239.000

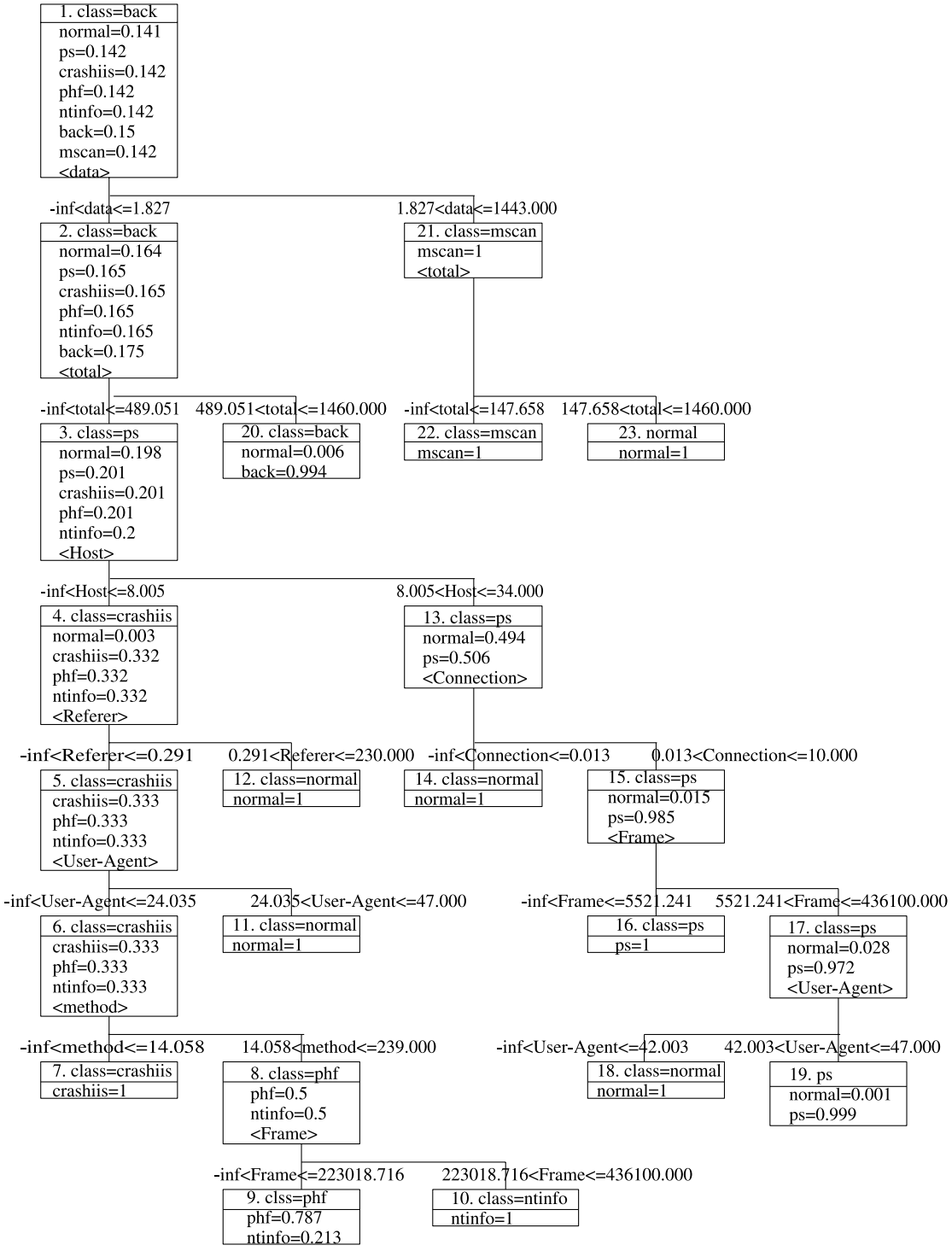

図 8 決定木
Fig. 8 Decision tree.

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課程修了．同年KDDI（株）入社．現在，（株）KDDI研究所セキュリティグループ研究員．タイムスタンプ，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KDD（株）入社．現在，（株）KDDI研究所セキュリティグループ主任研究員．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．電子情報通信学会会員．

竹森　敬祐（正会員）

1994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KDD（株）入社．2004年慶應義塾大学大学院博士課程修了．現在，（株）KDDI研究所セキュリティグループ研究主査．通信ネットワークおよびインターネットセキュリティの研究に従事，2002年度電子情報通信学会学術奨励賞受賞．電子情報通信学会員．

田中　俊昭（正会員）

1986年大阪大学大学院工学研究科通信工学専攻前期博士課程修了．同年KDD（株）入社．現在，（株）KDDI研究所セキュリティグループリーダー．暗号プロトコル，著作権保護，モバイルセキュリティ，インターネットセキュリティの研究に従事，電子情報通信学会員．