2004－ＣＳＥＣ－26　(19)
2004／7／20

学習データを自動生成する侵入検知システムの評価

山田　明†　三宅　優†　竹森　敬祐†　田中　俊昭†

† (株)KDDI 研究所
〒 356-8502 埼玉県上福岡市大原 2 丁目 1 番 15 号
E-mail: †{yamada.akira,miyake,takemori,tl-tanaka}@kddilabs.jp

あらまし　近年，既知の攻撃パターンファイル (シグネチャ) に基づく Intrusion Detection System (IDS) が広く利用されるようになってきたが，未知攻撃や亜種攻撃を検知できない課題がある．そこで，未知攻撃や亜種攻撃を検知するための機械学習アルゴリズムに基づく IDS が注目を集めている．しかし，実環境において学習データを作成することが困難なこと，ならびに試験環境で作成する学習データでは実環境の攻撃に対応しきれないことが問題となっている．筆者らは，シグネチャに基づく IDS の出力から学習データを自動生成し，これを学習することにより亜種の攻撃を検知できる IDS を提案している．本稿では，1999 DARPA IDS Evaluation Data の試験データを用いた検知率に関する詳細な評価を行うとともに，より実際の環境に近い状態を想定して，脆弱性スキャナによるトラヒックおよび実ネットワークのトラヒックを用いて評価を行う．検知対象として HTTP における攻撃に注目しており，すべてのデータにおいて亜種攻撃を的確に検知できることを確認する．

キーワード　侵入検知システム，機械学習，教師あり学習，異常検知，亜種攻撃，自動学習データ生成

Evaluations for Machine Learning Based IDS with Automatic Training Data Generation

Akira YAMADA†, Yutaka MIYAKE†, Keisuke TAKEMORI†, and Toshiaki TANAKA†

† KDDI R&D Laboratories Inc.
2-1-15 OHARA KAMIFUKUOKA-SHI, SAITAMA 356-8502, JAPAN
E-mail: †{yamada.akira,miyake,takemori,tl-tanaka}@kddilabs.jp

Abstract　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 conventional IDS.

Key words　Network intrusion detection, Machine learning, Supervised learning, Anomaly detection, Misuse detection, Variant attacks, Automatic training data generation.

1. はじめに

近年，ネットワークのトラヒックを監視して攻撃を検知する侵入検知システム (IDS:Intrusion Detection System) のうち，シグネチャに基づく IDS が提案され広く利用されている [1]〜[5]．シグネチャとは攻撃の特徴を登録するパターンファイルであり，トラヒックと比較することにより攻撃を検知するものである．シグネチャに基づく IDS は誤検知が少なく既知の攻撃

を確実に検知できるという点において効果的であるが，シグネチャを最新の状態に保ち続ける必要がある．つまり，新しい攻撃が出現するたびにその特徴を登録する必要があり，たとえ亜種攻撃であっても，元の攻撃と異なるパターンが利用されれば，その攻撃に対しても別のシグネチャを登録しなければならない．

一方，未知攻撃や亜種攻撃を検知するために，機械学習アルゴリズムに基づく IDS が提案されている．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，教師あり学習は教師なし学習に比べて高い検知率を示す．しかし，いずれも学習のための期間やデータを必要とするために，実環境に適用する際に問題となる．特に，教師あり学習を行う場合，その教師情報となる攻撃の有無やその発生時刻，種類といった情報を実環境において得ることは難しい．教師あり学習に基づく方式 [6], [7] では，学習データとして DARPA IDS evaluation data 1998, 1999 [8], [9] を用いて評価を行っている．DARPA データは攻撃の発生時刻や攻撃の種類を Web サイトにおいて公開しており [10]，容易に教師情報を得られるが，DARAP データだけを用いて学習を行う場合，実環境において効果的に攻撃を検知できない問題がある [11]．

この問題を解決するために，筆者らは，シグネチャに基づく IDS の検知結果から学習データを自動生成し，機械学習により亜種攻撃を検知できる IDS を提案した．そして，試験データを用いて簡単な評価を行った結果，亜種攻撃を検知できることを確認している [12]．本稿では，提案方式の有効性を確認するために，1999 DARPA IDS Evaluation の試験データによる詳細な評価を追記するとともに，新たに脆弱性スキャナによるトラヒックおよび実ネットワークのトラヒックを用いて評価を行う．検知対象は HTTP における攻撃について注目しており，すべてのデータにおいて亜種攻撃を検知できることを確認する．

本稿は 2. 章に提案方式の概要，3. 章において機械学習に基づく IDS の評価手法について検討し，4. 章に評価結果，5. 章に結論という構成である．

2. 提案方式の概要

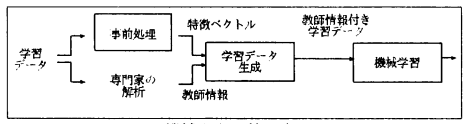
2.1 既存の機械学習に基づく IDS

機械学習に基づく IDS は教師あり学習と教師なし学習に分類される．教師あり学習は検知率が高いため注目されているが，教師情報を必要とする課題がある．

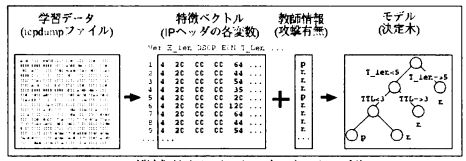
教師あり学習アルゴリズムに基づく IDS の一般的な手順を図 1(a) に示す．はじめに，事前処理を行うことにより学習データから特徴ベクトルを抽出する．次に，教師情報として攻撃の有無を特徴ベクトルに割り当てることで教師付き学習データを生成する．そして，教師付き学習データを用いて機械学習アルゴリズムを実行するにより未知攻撃，亜種攻撃検知のためのモデルを構築する．各変数の例を図 1(b) に示す．図において学習データは tcpdump ファイルであり，事前処理として IP ヘッダの各変数を抽出しており，学習アルゴリズムとして決定木を利用している．

2.2 機械学習に基づく IDS の問題点

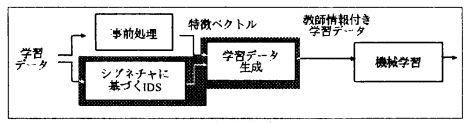
教師付き学習を行う方式の多くは教師情報として攻撃の有無



(a) 機械学習に基づく IDS



(b) 機械学習における各データの例



(c) 提案方式

図 1 既存の機械学習に基づく IDS と提案方式

Fig. 1 Conventional Machine Learning Based IDS and Proposed Architecture.

に関する情報を専門家の解析により抽出する．教師情報付き学習データとして DARPA データが頻繁に用いられる．DARPA データは攻撃の発生時刻や攻撃の種類などの解析結果を Web サイトにおいて公開しており [10]，容易に教師情報を得られるが，DARAP データだけを用いて学習を行う場合，実環境において効果的に攻撃を検知できない問題がある [11]．つまり，専門家による学習データ生成は日々公開される脆弱性に対応しなければならない煩雑な作業とコスト面で問題がある．

2.3 学習データ自動生成方式の提案

前述の問題を解決するために，提案方式 [12] は教師付き学習データを自動生成する．図 1(c) に提案方式の概要を示す．教師付き学習データを自動生成するために，シグネチャに基づく IDS から出力される攻撃の発生時間および種別を各特徴ベクトルに割り当てれる．ここでは，事前処理において特徴ベクトルだけでなく，特徴ベクトルを一意に特定できる変数を同時に抽出している．一意に特定できる変数により効率的に割り当てることができる．この変数としてプロトコルタイプ，IP アドレス，ポート番号，TCP のシーケンス番号などが利用できる．

2.4 プロトタイプの構成要素

表 1 にプロトタイプのシステム構成要素を示す．本稿では HTTP を対象とするが，提案方式は事前処理を変更することにより様々なプロトコルに適用できる．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する．シグネチャに基づく IDS として，シグネチャのカスタマイズが容易な Snort [4] を用いる．そして，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る．

決定木における変数を表 2 に示す．通常の学習データでは，攻撃を含むサンプルは非常に少なく，学習アルゴリズムの多くは，少ないサンプル数の事象を正しく分類できない．表 2 の *Weight* は攻撃サンプルに対して重みをつけることにより検知率を向上させる変数である．

表 1 プロトタイプのシステム構成要素
Table 1 System Components of Prototype.

Preprocessor	HTTP Request Header
Signature Based IDS	Snort 1.9.1
Machine Learning Algorithm	Decision Tree With Gini Index

表 2 決定木における各変数
Table 2 Parameters for Learning Algorithm.

Splitting Criterion	Gini Index
Weight	Each Class
Maximum Branches in Each Node	2
Maximum Elements in Terminal Node	0
Maximum Depth of Tree	∞

```
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) HTTP Request

	Request-line	Connection	User-Agent	Host	Accept	Accept-Encoding	Accept-Language	...
1	29	14	32	20	35	9	6	...
2	22	14	32	20	35	9	6	...
3	3	0	0	0	0	0	0	...
4	30	14	32	20	35	9	6	...
...	...							

(b) Example of Feature vectors for HTTP Request

図 2 HTTP リクエスト
Fig. 2 HTTP Request.

2.5 HTTP リクエストの特徴ベクトル抽出

HTTP リクエストの特徴ベクトルとして以下の変数を抽出する.

- リクエストラインの長さ [bytes].
- field-value の長さ [bytes].
- message-body の長さ [bytes].
- HTTP リクエストの大きさ [bytes].

ここで, HTTP リクエストにおいて, 該当する field-name が存在しない場合は field-value の長さを 0 byte にする. field-value が数値である場合は大きさではなく field-value そのものを特徴ベクトルとする. 図 2(a)(b) に HTTP リクエストおよび特徴ベクトルの例を示す.

3. 機械学習に基づく IDS の評価

3.1 評価データ

機械学習に基づく IDS の評価に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検知のためのモデルを構成する. そして, 学習データに含まれていない攻撃も含む検査データにより学習結果の評価を行う. ここで, 学習データに含まれている攻撃を既知の攻撃, 含まれていない攻撃を未知の攻撃と呼ぶこととする. 評価は検査データに含まれている既知攻撃, 未知攻撃が何件検知できるかによって行う. ここで, DARPA データなどの評価データには攻撃は種類が同一であっても, 細部が異なる攻撃が存在するため, すべてが同時に検知できるとは限らない.

3.2 従来の評価方法

従来の機械学習に基づく方式は, 以下の 2 種類の評価データによって評価されている.

- 1998, 1999 DARPA IDS Evaluation Data [8], [9] の評価データ
- 実環境などから収集する評価データ

DARPA データ [9] 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. データ全体に攻撃が 64 種類しか含まれておらず, 数種類しか未知の攻撃が存在しないため, 未知攻撃を検知する能力を評価するには不十分である. さらに, DARPA データのみを用いる学習は実環境とは乖離した学習結果を生成すると報告されている [11].

一方, 実環境から収集されるデータにより評価を行う場合, 未知の攻撃を含む検査データを用意できない. 既知攻撃を検知するとき, シグネチャに基づく IDS は機械学習に基づく IDS に比べ高い検知率を示す. 既知攻撃を解析することによりシグネチャを作成し, 該当の攻撃を検知できるようになる. たとえば, Snort の場合は, 付録に示すシグネチャを用いることにより, DARPA データに含まれる HTTP の攻撃を誤検知することなく検知できる.

したがって, 未知攻撃に関する評価を行うためには以下の条件が必要である.

- 学習データおよび検査データを用意する.
- 検査データには未知攻撃を含む.
- 実環境に即したデータを利用した評価も行う.

3.3 提案評価方法

本稿において, 提案方式の未知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.

(i) 1999 DARPA IDS evaluation data を再構成したデータ.

(ii) Nessus により生成されるデータを再構成したデータ.

(iii) ある企業 LAN ゲートウェイにおいて収集したデータ.

3.3.1 DARPA IDS Evaluation Data

検査データには学習データに含まれない攻撃が含まれる必要がある. そこで, 本評価では DARPA データの 4, 5 週目の HTTP リクエストのから 1 種類の攻撃を除いたデータを学習データとし, 取り除いた攻撃を検査データとしている. データには 64 種類の攻撃が含まれているが, プロトタイプでは HTTP リクエストを対象としているため, 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は 7 組構成し, 評価は 7 回行う. 評価に使用したデータの詳細を表 3 に示す. 例えば, 攻撃 *apache2* において, **51.102700** および **(1195)** はそれぞれ [9] において割り当てられている ID および HTTP リクエスト数である. **51.102700** は第 5 週 1 日目の 10 時 27 分 00 秒を意味する.

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 [requests].

Attack	ID (the number of requests)	Count
apache2	51.102700 (1195), 51.140100 (409), 53.171350 (412), ?? (1045)	3061
back	52.103409 (47), 53.102617 (40), 53.152648 (40), 55.084452 (40)	167
crashiis	41.161308 (1), 42.210410 (1), 51.183623 (1), 52.135003 (1), 54.115701 (1), 55.081418 (1), 55.125112 (1), 55.125830 (1)	8
mscan	54.195951 (894)	894
ntinfoscan	44.080000 (11), 54.110416 (17), 54.183002 (11),	39
phf	42.094131 (1), 44.161242 (1), 53.085717 (1), 54.090101 (1)	4
ps	41.084031 (1), 52.112045 (1)	2
normal		431975
Total		436100

表 4 Nessus によって生成される HTTP リクエスト
Table 4 HTTP Request Generated by Nessus [requests].

	Alert class	Count
Detectable	WEB-MISC	696
	WEB-CGI	444
	WEB-IIS	297
	WEB-PHP	30
	WEB-FRONTPAGE	16
	WEB-COLDFUSION	12
Undetectable		2728
Total		4723

3.3.2 脆弱性スキャナ Nessus

様々な攻撃を発生させることができる脆弱性スキャナ Nessus [13] を利用して攻撃を送信させ，この攻撃を検知させることにより，評価を行う．

本稿では，Snort によって検知できる攻撃を学習データとし，検知できない攻撃を検査データとする．また，学習データとしてある企業の LAN ゲートウェイにおいて収集したデータを同時に用いる．このような構成によって，学習データに存在しない未知攻撃を含む検査データを構成できる．ここで，LAN ゲートウェイにおいて収集するデータを 3.3.3 で説明する．この評価において 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しているため，840 種類の Nessus による攻撃を対象とした．これらは 4723 件の HTTP リクエストを発生する．このリクエストを用いて Snort による検知を行ったところ，1495 件のリクエストは Snort によって検知可能であり，2728 件のリクエストは検知不可能であった．これは，Nessus が通常の HTTP リクエストから得られる情報を利用してスキャンしているためである．Nessus によって生成される HTTP リクエストの詳細を表 4 に示す．

表 5 ある企業 LAN ゲートウェイにおける HTTP リクエスト [requests].
Table 5 HTTP Request Captured at a Corporate LAN Gateway

Name	Alert	Count
CodeRed	WEB-IIS	19
Nimda	WEB-IIS	32
Other	WEB-CGI	81
	WEB-MISC	17
	WEB-PHP	1
Normal		42435
Total		42584

3.3.3 LAN ゲートウェイ

ここでは，学習によって構成される決定木の分岐ルールと学習データ生成に利用したシグネチャを比較する．学習データとしてある企業 LAN ゲートウェイにおいて収集したデータを使用し，検査データは用意しない．学習データには 4258 件の HTTP リクエストが含まれており，149 件のリクエストが Snort により攻撃として判定される．データの詳細を表 5 に示す．専門家による詳細解析を行った結果，Nimda ワームのアクセスが 14 件，CodeRed II ワームのアクセスが 19 件発見された．そして，Snort によって検知されるその他のリクエストは誤検知であった．

4. 評価結果

4.1 結果概要

3.3 節で説明した各評価データを使用して提案方式の評価を行った．その結果を表 6 に示す．また，既存の IDS として Snort の結果を併記する．Snort はシグネチャに基づく IDS であるため，既知攻撃を検知でき，未知攻撃を検知できない．提案システムにおいて多くの未知攻撃が検知されている．この理由は，HTTP ヘッダに対する特徴ベクトルが適切に抽出されているからである．一般に機械学習に基づく IDS の検知率は特徴ベクトルの抽出方法に大きく依存している．特徴ベクトルとして，攻撃の特徴を適切に表現できる変数の集合を抽出することにより機械学習アルゴリズムは高い検知率を示す．以下にそれぞれの結果に関して考察する．

4.2 DARPA IDS Evaluation Data

提案システムは 3 種類の攻撃 *apache2*, *ntinfoscan*, *phf* をそれぞれ *back*, *phf*, *ntinfoscan* の亜種として検知できる．しかし，4 種類の *back*, *crashiis*, *mscan*, *ps* を検知できない．亜種攻撃として検知できる理由は DARPA データに含まれている攻撃は HTTP ヘッダに，攻撃の成功に直接影響はない特徴があるためである．具体的には *Host*, *Connect*, *User-Agent*, *Referer*, *Accept* が存在しなかったり，特定の値であるなどの特徴である．

4.3 脆弱性スキャナ Nessus

既存の IDS は 2727 件の HTTP リクエストを検知できない．一方，提案システムは，この 2727 件のリクエストのなかで 2687 件のリクエストを攻撃として検知した．この理由も DARPA データの結果と同様である．Nessus は異なる攻撃に対して同

表 6 評価結果
Table 6 Evaluation Summary.

		Conventional IDS				Proposed system			
		TP	TN	FP	FN	TP	TN	FP	FN
(i)DARPA	Known attacks (apache2)	3935	429104	0	0	3732	429104	203	0
	Unknown attack (apache2)	0	0	0	3061	3061	0	0	0
	Unknown attack (back)	0	0	0	167	0	0	0	167
	Unknown attack (crashiis)	0	0	0	8	0	0	0	8
	Unknown attack (mscan)	0	0	0	894	0	0	0	894
	Unknown attack (ntinfoscan)	0	0	0	39	39	0	0	0
	Unknown attack (phf)	0	0	0	4	4	0	0	0
	Unknown attack (ps)	0	0	0	2	0	0	0	2
(ii) Nessus	Known attacks	1494	0	0	0	793	0	0	701
	Unknown attacks	0	0	0	2727	2687	0	0	40
(iii) LAN Gateway	Known attacks	51	42435	98	0	33	42531	2	18

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

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.

Attack	Rules	Explanation
CodeRed II	$-\infty$<Connnection≦0.011, 0.073<Content-type≦33.0, 54.500<Request≦239.000	*Content-type*; a lower-case letter is included in the field-name *Content-Type*
Nimda	0.011<Connnection≦5.000	*Connnection*; Misspelling of field-name *Connection*

様の HTTP ヘッダを伴ったリクエストを生成するため，特徴ベクトルが類似してしまい，Nessus の特徴を持つリクエストを攻撃と判断する．また，提案システムが 701 件の誤検知を発生する理由は，上記のように Nessus は攻撃種別に関わらず同様の HTTP ヘッダを伴ったリクエストを生成するため，適切な攻撃種別に分類できないためである．

4.4 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を表 7 に示す．一方，各々のシグネチャを表 8 に示す．等しい攻撃を検知するルールであるにも関わらず，表 7 と表 8 は大きく異なる．これはシグネチャは攻撃の特徴を記述しているが，分岐ルールはワームの特徴を記述しているからである．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 には亜種が存在するが同様の特徴をもっている．これにより，亜種攻撃も検知できることとなった．

4.5 考　　察

以上の評価結果を考慮すると，提案方式はワームや攻撃ツールの特徴を抽出することができ，亜種攻撃を検知するための効果的なシステム構成であることが分かる．これに対し，既存のシグネチャに基づく IDS は，亜種攻撃を検知するために，亜種攻撃に対応するシグネチャをすべて登録しなければならない．

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.

Attack	Signature
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;)

つまり，亜種攻撃が発生する度にシグネチャを更新する必要がある．近年の状況においては，ワームなどの攻撃は数時間以内に全世界に攻撃が拡散するため，シグネチャの更新タイミング以前に攻撃を受ける確率が非常に高くなっている．提案方式はシグネチャに基づく IDS を学習に利用するため，シグネチャに登録されている攻撃の亜種攻撃を検知できることになる．また，シグネチャを更新することにより新しい攻撃に対応する学習データを生成できる．

また，評価 (iii) において，ワームの新しい特徴を発見できた．これらの特徴を表 9 に示すようなシグネチャとしてシグネチャに基づく IDS にフィードバックすることによりシグネチャに基づく IDS の精度を向上させることができる．

5. 結　　論

筆者らはこれまでシグネチャに基づく IDS を利用することにより学習データを自動的に生成し，これを用いて学習を行うことにより亜種攻撃を検知できる IDS を提案してきた．提案方式はシグネチャが更新される場合でも最新の攻撃に対する学習データを生成できる．

本稿では提案方式の評価に関して，DARPA のトラヒックを

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.

Attack	New SignatureRules
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" ;)

用いたより詳細な解析結果を示すとともに，Nessus により生成したトラヒック，ある企業 LAN のゲートウェイにおいて収集されたトラヒックによる解析を示した．その結果，提案方式によって生成する学習データは機械学習アルゴリズムに適用でき，機械学習からワームや攻撃ツールの特徴を抽出することが可能であることを示した．また，新たに加えた評価データに対しても，提案方式は亜種攻撃を検知できることが分かった．今後の課題として，提案方式を HTTP 以外のプロトコルに対して適用する方法の検討を進める．

文　献

[1] Roesch, M.: Snort – lightweight intrusion detection for networks. In: Proceedings of the 13th Conference on Systems Administration (LISA-99). (1999)

[2] Paxson, V.: Bro: a system for detecting network intruders in real-time. Computer Networks: The International Journal of Computer and Telecommunications Networking **31** (1999) 2435–2463

[3] NFR security, Inc., network flight recorder, `http://www.nfr.com` (2004)

[4] Snort, the open source network intrusion detection system, `http://www.snort.org` (2004)

[5] Internet security systems, Inc., realsecure, `http://www.iss.net` (2004)

[6] Lee, W., Stolfo, S.J.: A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security (TISSEC) **3** (2000) 227–261

[7] Barbara, D., Wu, N., Jajodia, S.: Detecting novel network intrusions using bayes estimators. In: Proceedings of the 1st SIAM International Conference on Data Mining (SDM-01). (2001)

[8] Lippmann, R., Cunningham, R.K., Fried, D.J., Graf, I., Kendall, K.R., Webster, S.E., Zissman, M.A.: Results of the DARPA 1998 offline intrusion detection evaluation. In: Recent Advances in Intrusion Detection. (1999)

[9] Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. computer networks. Computer Networks: The International Journal of Computer and Telecommunications Networking **34** (2000) 579–595

[10] LINCOLN LABORATORY, `http://www.ll.mit.edu/ist/ideval/` (1999)

[11] Mahoney, M.V., Chan, P.K.: An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. In: Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003, Pittsburgh, PA, USA, September 8-10, 2003, Proceedings. (2003)

[12] 山田 明, 三宅 優, 田中 俊昭: 亜種攻撃を検知できる侵入検知システムの提案. コンピューターセキュリティシンポジウム. (2003)

[13] Nessus, `http://www.nessus.org` (2004)

付　録

1. DARPA データに対する分岐ルールとシグネチャ

LINCOLN 研究室の Web サイト [10] の情報を利用して Snort のためにシグネチャを作成した (表 A·1)．このシグネチャを用いることにより評価に用いた 7 種類の攻撃はすべて検知するができる．また，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 A·2 に示す．また，決定木の例を図 A·1 に示す．この決定木は (i) の評価において *apache2* 攻撃を除いた学習データにより構築したものである．

表 A·1 DARPA データに対する Snort シグネチャ

Table A·1 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 A·2 DARPA データに対する分岐ルール

Table A·2 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

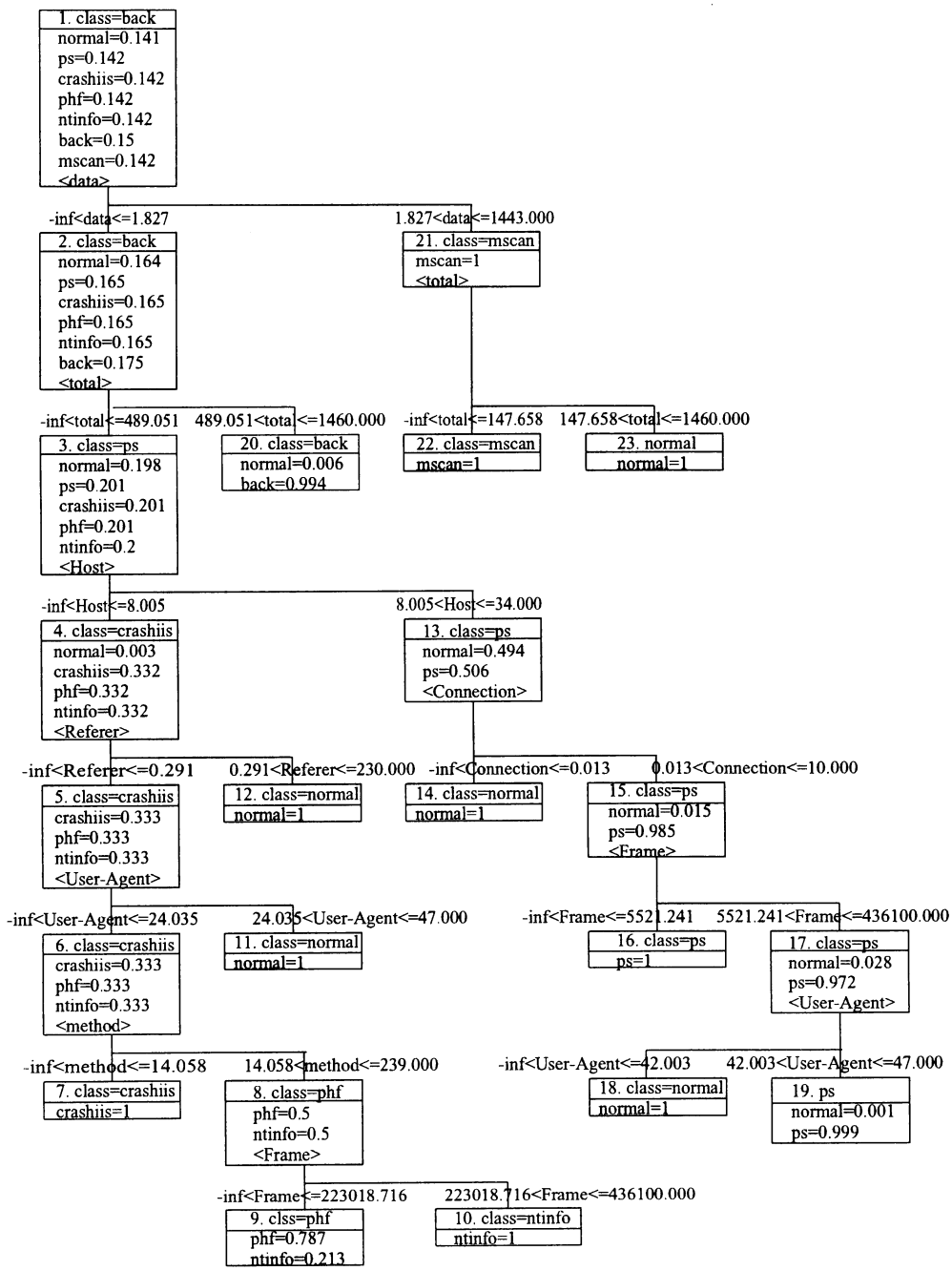


図 A·1 決 定 木

Fig. A·1 Decision Tree.