Vol.2013-CSEC-62 No.52
Vol.2013-SPT-6 No.52
2013/7/19

ナイーブベイズを用いた攻撃予測に関する評価・考察

面 和成*

概要：不正アクセスからサーバを守る侵入検知システム（IDS）は有効な対策の 1 つであると考えられている．しかしながら，IDS は，ある攻撃シナリオに対して，膨大なアラートを発生させる場合がある．例えば，目的の攻撃に紐付けされる複数の関連アラートが好例である．そのため，攻撃を把握・予測する際に，管理者が IDS の関連アラートに対してどのように対処すればよいかが難しい．本稿では，機械学習手法の一つであるナイーブベイズを用いて関連アラートの相関を導出することで攻撃セッションを学習し，マルウェアダウンロードの予測可能性について評価・考察を行う．本評価では，CCC DATAset に対して具体的なマルウェアダウンロードの予測確率を算出し，マルウェアダウンロードが実験的に 98.3%の確率で予測可能であったことを示す．したがって，未知のマルウェアも含め，深い専門性がなくとも，IDS の関連アラートからマルウェアダウンロードを高い確率で予測できることが分かった．

1. はじめに

侵入検知システム（IDS）とは，ネットワーク上のサーバなどへの不正なアクセスの兆候を検知し，ネットワーク管理者に通報する機能を持つソフトウェア，またはハードウェアである．IDS は，不正アクセスからサーバを守るための有効な対策の 1 つであると考えられており，Snort に代表されるように IDS は広く普及している．IDS は，不正アクセスや攻撃命令の通信パターンをシグネチャ（ルール）として事前に定義することにより怪しいイベントを判別する．ただし，アラート同士の相関については考慮されておらず，個々のイベントは独立で処理される．

一方，官公庁や企業のウェブサイト，サーバへのサイバー攻撃が急増する中，高度なネットワーク技術を持つ専門家（ホワイトハッカーとよばれている）の不足が懸念されている．そのため，ある攻撃シナリオに対して IDS が膨大なアラートを発生させた場合，高度な技術者でない限り，その適切な対処は難しいであろう．ここで，攻撃シナリオとは，攻撃を実行する際に攻撃者が行うイベントの組み合わせである [1]．専門家は，過去に起こった攻撃に関する知識と経験から，この関連アラートが何を意味するのかを知ることができるかもしれない．しかしながら，攻撃を把握・予測する際に，多くの管理者が IDS の膨大な関連アラートに対してどのように対処すればよいかが難しいという現状がある．

攻撃には，攻撃シナリオがあり，目的の攻撃の前に事前処理や権限取得等を行うことが知られている．事前処理では目的のサーバに脆弱性がないかの調査が考えられ，権限取得では root 権限等を取得するなどが考えられる．本稿では，マルウェアダウンロードを目的の攻撃と考え，これが起こらなければよしとする．したがって，本研究の目的は，事前処理や権限取得の段階でマルウェアダウンロードを予測することである．

本稿では，機械学習手法の一つであるナイーブベイズを用いて関連アラートの相関を導出することで攻撃セッションを学習し，マルウェアダウンロードの予測可能性について評価・考察する．攻撃セッションとは，1 つの攻撃シナリオにおいて目的とする攻撃に関連する IDS のアラート列である．つまり，この学習によってマルウェアダウンロードに関する攻撃シナリオが自動的に作成できることを意味する．その結果，深い専門性がなくてもマルウェアダウンロードを予測できることが期待される．さらに，本評価では，CCC DATAset[2] に対して具体的なマルウェアダウンロードの予測確率を算出し，マルウェアダウンロードが実験的に 98.3%の確率で予測可能であったことを示す．このとき，IDS がアラートを出力する度にマルウェアダウンロードの予測確率をリアルタイムで出力する．したがって，未知のマルウェアも含め，深い専門性がなくとも，IDS の関連アラートからマルウェアダウンロードを高い確率で予測できることが分かった．

本稿の構成を示す．2 章では関連研究を概観し，3 章では準備として攻撃セッション及びナイーブベイズを定義し，4 章では CCC DATASet の内容について述べる．次に，5 章ではナイーブベイズ用いた提案手法を説明し，6 章では

*北陸先端科学技術大学院大学 (JAIST)

© 2013 Information Processing Society of Japan

マルウェアダウンロードの予測に関する実験及び評価について述べ，7 章でナイーブベイズによる評価の有効性について考察する．最後に，8 章で本研究についてまとめる．

2. 関連研究

攻撃予測に関して数多くの研究がなされているが，ここでは IDS アラートの相関を利用する研究を中心に述べる．IDS アラートの相関を利用する研究には，(a) IDS が出力するアラート数を減らすための研究，(b) 攻撃シナリオを構築するためにアラート間の関係を発見する研究，の大きく 2 つがある．前者は，アラートを集約したりクラスタリングしたりするものであり，アラート自体に意味があり，管理者がそのアラートを認知しやすくするものである．これに対して後者は，アラート間の確率的相関関係を導出し，目的の攻撃を含む攻撃シナリオを構築する研究であり [1], [3]，本研究はこれの発展型に属す．以後，上記 (b) の関連研究について述べる．

IDS のアラートを学習して攻撃を予測する手法 (Nexat) が，Cipriano らによって 2011 年に提案された [4]．ここでは，IDS のアラートをベースに攻撃セッションを生成し，機械学習を用いてアラート間の確率的推移を導出することによって，次のアラートが何であるかを予測する手法である．また，森らは，SCIS2013 において，この Nexat を CCC DATAset[2] に適用し，Nexat がマルウェアダウンロードの予測に有効であることを実験的に示した [5]．

一方，機械学習手法の一つであるベイジアンネットワークが注目されている．実際，これまでにベイジアンネットワークが数多くの侵入検知に適用されてきた [6], [7], [8]．特に [8] は，著者の知る限り，ナイーブベイズを用いて攻撃シナリオを考慮した最初の論文である．ナイーブベイズは，ベイジアンネットワークにおいて最もシンプルなグラフ構造を有しているものである．

他に，CCC DATAset に対してアラートの相関を考慮した研究がある [9]．これは，IDS のアラートを基に，Bot の攻撃開始からマルウェア感染までの通信を分析し，その間の IDS のルールを抽出し，最適なルールの組み合わせについて検証したものである．具体的には，CCC DATASet の攻撃通信データを Snort に適応し，IP アドレスやポート番号を用いてマルウェア感染に関する検知モデルを構築し，そのルールを抽出している．

3. 準備

3.1 攻撃セッション

攻撃セッションとは，1 つの攻撃シナリオにおいて，目的とする攻撃に関連する IDS のアラート列である．我々は，[4] で定義されている攻撃セッションを用いる．

Definition1 (攻撃セッション [4]) 攻撃セッションとは，以下の条件を満たすアラート列のことである．次のうち少なくとも一つを満たせば，攻撃セッション S にアラート A を追加する．

- アラート A の宛先が攻撃セッション S 内にある過去のアラートの宛先と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの送信元と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの宛先と一致

さらに，アラート A が攻撃セッション S の最後のアラートから w 秒以内の時間ウィンドウ内に存在する．

目的の攻撃を $\mathcal{O}$ としたとき，攻撃セッションは $S = \{A_1, A_2, \ldots, A_m, \mathcal{O}\}$ と表される．つまり，アラート列 $\{A_1, A_2, \ldots, A_m\}$ は $\mathcal{O}$ の前兆となる関連アラート列と捉えることができる．もちろん，$\mathcal{O}$ が $\mathcal{O}$ の関連アラートになる場合もある．

3.2 ナイーブベイズ

ベイジアンネットワークは，有向非巡回グラフ（DAG）で構成され，頂点がイベントを表し，辺がイベント間の確率的な関係を表すといった図式で示すことができる．すなわち，ベイジアンネットワークは，原因と結果の確率的な推移をグラフィカルに表現する理論といえる．また，親ノードから見て各ノードへの推移が条件付き確率分布によって定量的に与えられる．ナイーブベイズは，非常に単純なベイジアンネットワークであり，観測されない親ノードとそれに対して複数の子ノードを持つ DAG を構成し，親から見て子ノード間が独立であるという強い仮定を持つが，分類精度は高いと言われている．

ナイーブベイズは分類問題を扱うのに適している．実際，データベース内の対象がどこに分類されるかは，親ノードを隠れ変数と考えることによって行われる．このとき，子ノードはこの対象を記述する様々な属性（特徴変数）を表す．一度このネットワークが定量化されると，次の式で表されるベイズ規則に従って，どんな新しい対象も分類可能となる．

$$P(c_i|A) = \frac{P(A|c_i)P(c_i)}{P(A)} \quad (1)$$

ここで，c_i はクラス i（class i）を表し，A は行動（action）を表す．つまり，ある対象が，得られた結果（行動）A に基づいてクラス c_i に分類されるかどうかが判定される．

4. CCC DATASet[2]

4.1 概要

CCC DATASet とは，サイバークリーンセンターのサーバ型ハニーポットで収集しているボット観測データである．今回使用する CCC DATAset 2010, 2011 は，(1) マルウェア検体，(2) 攻撃通信データ，(3) 攻撃元データの 3 つから構成されたボット観測データ群である．本研究では，

© 2013 Information Processing Society of Japan

(2) 攻撃通信データ及び (3) 攻撃元データを用いて，マルウェアのダウンロードの予測について評価する．

4.2 攻撃通信データ

攻撃通信データは，研究用データセットを提供するための観測装置で取得した通信のフルキャプチャデータである．具体的には，CCC DATAset 2010, 2011 では，2 台のハニーポット（honey001, honey002）のパケットキャプチャログである．具体的な収集期間は，2010 年 3 月 5 日〜11 日，2010 年 8 月 18 日〜31 日，2011 年 1 月 18 日〜31 日である．今回，この期間の攻撃通信データを学習に用いることにした．

4.3 攻撃元データ

攻撃元データは，観測装置で取得したマルウェア取得時のログデータ（マルウェア検体の取得時刻，送信元 IP アドレス，送信元ポート番号，宛先 IP アドレス，宛先ポート番号，TCP または UDP，マルウェア検体のハッシュ値（SHA1），ウイルス名称，ファイル名）である．これは，ウィルススキャナによって出力されたログである．CCC DATAset 2010 は 2009 年 5 月から 12 ヶ月間に 92 台のハニーポットによって収集され，CCC DATAset 2011 では 2010 年 5 月から 9 ヶ月間に 72 台のハニーポットにより収集された．

5. 提案手法

一般的に攻撃者は，事前定義された攻撃シナリオによって攻撃を行うと考えることができる．本提案手法は，行動に関連付けられる条件を予め決める必要がなく，専門家の知識を多く必要としない．つまり，本方式では，目的の攻撃（マルウェアダウンロード）にたどり着くための攻撃シナリオが自動的に構築される．このマルウェアダウンロードの有無は，ナイーブベイズのクラスによって表される．

5.1 前処理

攻撃シナリオを表すナイーブベイズを学習させるためには，マルウェアダウンロードのイベントが観測アラート列に含まれるように IDS のルールを前処理しなければならない．なぜなら，IDS は，マルウェアダウンロードそのもののアラートを出力しないことが多いためである．この処理によって，IDS がマルウェアダウンロードの観測アラート列を出力する．観測アラート列は時系列で並べられたアラート列であり，これによって，マルウェアダウンロードを含む攻撃シナリオが学習される．

5.2 データ抽出フェーズ

データ抽出フェーズでは，まず初めにルールに関する前処理を行った IDS に対して，攻撃通信データを適用する．

表 1 観測アラートマトリックス

攻撃セッション	A_1	A_2	$\cdots$	A_{11}	$A_{12}(=\mathcal{O})$	$\mathcal{O}$
AS_1	0	1	$\cdots$	1	0	1
AS_2	0	1	$\cdots$	1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
AS_n	1	0	$\cdots$	0	0	1

次に，IDS から得られた観測アラート列に対して，定義 1 に従って，マルウェアダウンロードを目的の攻撃とする攻撃セッションを構築する．攻撃セッション構築の際に時間ウィンドウを設けるが，マルウェアダウンロードのアラートの時刻によっては時間ウィンドウが重なることに注意する．

5.3 学習フェーズ

目的の攻撃であるマルウェアダウンロードに対して，ナイーブベイズを構築する．そこで，全攻撃セッションに対して，どのアラートが観測され，マルウェアダウンロードがあったのかどうかを整理するために，表 1 のような観測アラートマトリックスを生成する．攻撃セッション $\{AS_1, \ldots, AS_n\}$ は，定義 1 に従い，時間が最近のものから並べられていることに注意する．

表 1 において，一番右の列は目的の攻撃を表す列であり，その他 $\{A_1, \ldots, A_{12}\}$ は目的の攻撃に対する関連アラート列を表す．1 は攻撃セッション内にそのアラートが存在していることを表し，0 は存在していないことを表す．例えば，攻撃セッション AS_1 では，目的の攻撃としてマルウェアダウンロードが観測されており，その関連アラートとして A_2 と A_{11} が時間ウィンドウ内に観測されている．ここには自身のアラートしか含まれない単一アラートで構成される攻撃セッションも含まれ，観測された全ての攻撃セッションが整理される．

5.4 予測フェーズ

マルウェアダウンロードの予測を定量的に評価するために，予測確率を算出する．本節では，直近の w 時間内の観測アラート列から攻撃の目的であるマルウェアダウンロードをどのように予測するかを示す．ナイーブベイズでは，予測確率が次のベイズ公式で算出される．

$$P(c_1|A_j) = \frac{P(A_j|c_1)P(c_1)}{P(A_j)} \tag{2}$$

さらに，観測アラートが複数ある場合は次の式で表される．

$$P(c_1|A_1, \cdots, A_{12}) = \frac{P(A_j|c_1)P(c_1)}{P(A_j)} \tag{3}$$

ここで，ナイーブベイズによる具体的な計算方法を述べる．式 (1) において，$c_0(i=0)$ をマルウェアダウンロード無し，$c_1(i=1)$ をマルウェアダウンロード有りとする．そこで，マルウェアダウンロードが起こる事前確率 $P(c_1)$ は

© 2013 Information Processing Society of Japan

表 2　観測アラート

	アラート内容	SID
A_1	SHELLCODE x86 OS agnostic fnstenv geteip dword xor decoder	17322
A_2	NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrPathCanonicalize overflow attempt	7209
A_3	SPECIFIC-THREATS msblast attempt	9422
A_4	NETBIOS DCERPC NCACN-IP-TCP ISystemActivator RemoteCreateInstance attempt	9607
A_5	NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrpPathCanonicalize path canonicalization stack ...	14782
A_6	SPECIFIC-THREATS sasser attempt	9419
A_7	NETBIOS DCERPC NCACN-IP-TCP lsass DsRolerUpgradeDownlevelServer overflow attempt	2508
A_8	NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation overflow attempt	8694
A_9	SHELLCODE x86 OS agnostic xor dword decoder	17344
A_{10}	FILE-IDENTIFY Portable Executable binary file magic detected	15306
A_{11}	SPECIFIC-THREATS lovegate attempt	9423
$A_{12}(=\mathcal{O})$	Malware download	

下記となる．

$$P(c_1) = \frac{\#c_1 + 1}{n + 2} \tag{4}$$

ここで，$\#c_1$ は表 1 におけるマルウェアダウンロードが目的の攻撃になっている行数（$\mathcal{O}$ が 1 となっている行数）を表している．

次に条件付確率 $P(A_j|c_1)$ は下記のとおりである．

$$P(A_j|c_1) = \frac{\#(A_j \cap c_1)}{\#c_1 + 2} \tag{5}$$

この確率も表 1 から算出される．

本提案手法は，IDS と組み合わせることによって，リアルタイムでアラートを出力することを想定する．つまり，IDS が新しいアラートを出力する度にマルウェアダウンロードの予測確率が計算される．そこで，マルウェアダウンロードかどうかを判断するために，予測確率に閾値を設定する．もし新しい予測確率がこの閾値を超えたなら，マルウェアダウンロードが起こるということを管理者等に通知する．逆に，予測確率がこの閾値を超えない限り，マルウェアダウンロードが起こらないと判断し，次のアラートを待つ．

6. マルウェアダウンロードの予測実験

6.1 目的

本稿では，CCC DATAset におけるマルウェアダウンロードを目的の攻撃と考え，これが起こらなければよしとする．そこで，CCC DATAset から観測アラートマトリックス（表 1 参照）を生成し，具体的な予測確率を算出し，その正答率や誤検知率を評価する．

6.2 実験内容

本実験では，CCCDATAset2010, 2011 のデータのうち，2 台（honey001, honey002）の攻撃通信データ及び攻撃元データを用いて，マルウェアダウンロードの予測評価を行う．予測実験を行うために，提案手法を Python で実装し，IDS としては Snort を用いた．

学習期間は次の 3 つである．

(1) 2011/1/18–24
(2) 2010/8/25–31
(3) 2010/3/5–11

まず初めに，上記期間の攻撃通信データを Snort に適用し，それぞれの観測アラート列を出力する．上記 3 期間の観測アラートは，全部で 12 種類であった（表 2 参照）．次に，これら 3 期間の学習データに対して，2011 年 1 月 25 日と 2011 年 1 月 31 日の 2 日間のマルウェアダウンロードの予測確率を時系列で算出する．

予測は，Snort のアラートのみで行い，このアラートにはマルウェアダウンロードのアラートは含まれない．予測が成功したかどうかは，攻撃元データと突き合わせることで確認する．

6.3 実験手順

6.3.1 前処理

攻撃通信データ（pcap ファイル）を IDS である Snort（2.9.3.0 Ruleset）に入力して観測アラート列を作成する．しかしながら，Snort はマルウェアダウンロードそのもののアラートを出力しない場合が多い．そこで，Snort がマルウェアダウンロードのアラートを出力するように，[9] と同様の方法でローカルルールを追加した．ただし，このローカルルールはマルウェアダウンロードだけでなく，正常なダウンロードに対してもアラートを出力するため，マルウェアダウンロードに対するアラートのみとなるように誤検知の除去を行った．具体的には，アラート時刻，送信元 IP アドレス，送信元ポート番号，宛先 IP アドレス，宛先ポート番号でマッチングさせ，Snort がローカルルールによってマルウェアダウンロードを検知したアラートのみを抽出した．

6.3.2 データ抽出フェーズ

ルールに関する前処理を行った Snort に対して攻撃通信

© 2013 Information Processing Society of Japan

表 3　マルウェアダウンロードの平均予測成功確率

学習期間	予測日	時間ウィンドウ w (秒)	honey001	honey002
(1)	1/25	1	**84.2%**	**88.1%**
		5	72.8%	76.2%
		10	67.4%	71.1%
	1/31	1	**90.6%**	**90.6%**
		5	77.9%	77.9%
		10	72.5%	72.5%
(2)	1/25	1	**70.8%**	**74.3%**
		5	55.9%	58.8%
		10	45.2%	47.6%
	1/31	1	**77.3%**	**77.3%**
		5	60.2%	60.2%
		10	48.0%	48.0%
(3)	1/25	1	**83.0%**	**84.6%**
		5	75.4%	76.6%
		10	65.8%	67.0%
	1/31	1	**89.2%**	**89.2%**
		5	80.3%	80.3%
		10	69.7%	69.7%

データを適用し，観測アラート列を作成し，攻撃セッションを構築する．なお，攻撃セッションは，定義 1 に従い，Cipriano らの方式と全く同じ方法で構築する．本実験では，目的の攻撃がマルウェアダウンロードである場合のみの攻撃セッションを用いる．

6.3.3　学習フェーズ

学習フェーズでは，表 1 の観測アラートマトリックスを作成することが必要である．この表から事前確率 $P(c_1)$，尤度 $P(A_j|c_i)$，アラートの発生確率 $P(A_j)$ を算出する．具体的な観測アラートの内容は表 2 である．

6.3.4　攻撃予測フェーズ

攻撃予測では，2011 年 1 月 25 日と 2011 年 1 月 31 日の 2 日間攻撃通信データを Snort に適用して出力された観測アラート列に対して行う．ただし，このとき Snort にはローカルルールを含めないため（デフォルトルールのみ），観測アラート列にはマルウェアダウンロードのアラートは含まれないことに注意する．学習フェーズで構築したナイーブベイズを用いて，予測確率 $P(c_1|A_j)$ を算出する．

6.4　実験結果

表 3 は，時間ウィンドウのサイズを 1 秒，5 秒，10 秒と変化させたときの 3 つの学習期間に対する平均予測成功確率を示したものである．この結果より，時間ウィンドウのサイズが 1 秒のときに最も高い予測確率であることが明らかになった．以後，時間ウィンドウサイズを 1 秒としたときの結果を示す．

図 1-4 は，学習期間 (1) に対する 2011/1/25, 31 における honey001 と honey002 の予測結果である．横軸は，1 日 (24 時間) の時間を表し，縦軸は予測確率を表している．グ

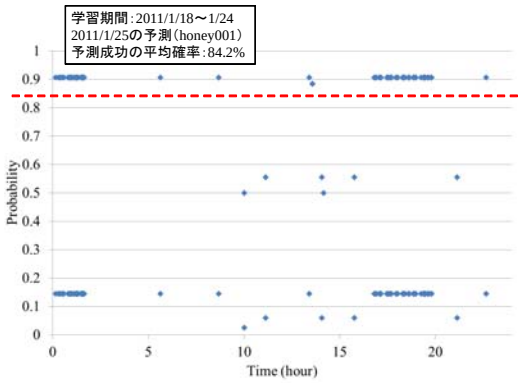


図 1　学習期間 (1) に対する 2011/1/25 の予測確率 (honey001)

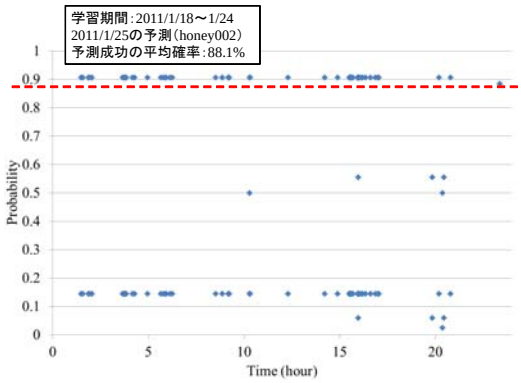


図 2　学習期間 (1) に対する 2011/1/25 の予測確率 (honey002)

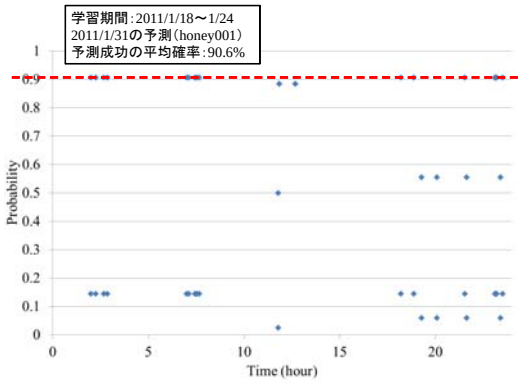


図 3　学習期間 (1) に対する 2011/1/31 の予測確率 (honey001)

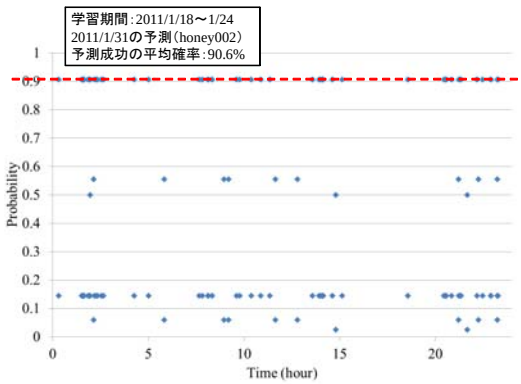


図 4　学習期間 (1) に対する 2011/1/31 の予測確率 (honey002)

© 2013 Information Processing Society of Japan

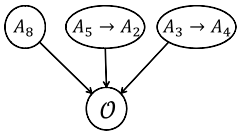


図 5 マルウェアダウンロードに関する攻撃セッション

表 4 関連アラートに対するマルウェアダウンロードの予測確率

関連アラート	学習期間 (1)	学習期間 (2)	学習期間 (3)
A_3	5.86%	4.78%	18.9%
A_5	14.4%	11.0%	18.2%
A_8	49.9%	–	–
$A_5 \to A_2$	90.6%	77.3%	89.2%
$A_3 \to A_4$	55.5%	47.7%	94.0%

ラフ上の波線は，予測成功の平均確率を示している．予測成功の平均確率は，本提案手法が予測した出力のうち，攻撃元データと突き合わせてマルウェアダウンロードを確認できたものの予測確率の平均値である．この予測成功確率は，TPR（True Positive Rate）を指す．この実験結果から分かるとおり，低い予測確率でマルウェアダウンロードがほとんど起こっていないことが分かる．

攻撃元データにおけるマルウェア数は，honey001 (2011/1/25) で 20 検体，honey002 (2011/1/25) で 17 検体，honey001 (2011/1/31) で 7 検体，honey002 (2011/1/31) で 16 検体の合計 60 件体（UNKNOWN を含む 7 種類）であった．また，これらに対して，Snort のアラートの総数は，honey001 (2011/1/25) で 141 回，honey002 (2011/1/25) で 111 回，honey001 (2011/1/31) で 48 回，honey002 (2011/1/31) で 121 回であった．

図 5 は，目的の攻撃をマルウェアダウンロードとした場合に，学習フェーズで学習した具体的な攻撃セッションである．事前確率は，学習期間 (1)(2)(3) でそれぞれ，$P(c_1) = 9.64\%$，$P(c_1) = 5.63\%$，$P(c_1) = 12.1\%$ であった．また，関連アラートに対するマルウェアダウンロードの予測確率は，表 4 を参照されたい．

7. 考察

7.1 マルウェアダウンロードの予測について

今回の実験では，マルウェアダウンロード回数が 2 日間で合計 60 回行われ，そのうち 59 回のマルウェアダウンロードを予測できた (約 98.3%)．また，60 回のマルウェアダウンロードの中で 1 回だけ未知のマルウェアが（攻撃元データで UNKNOWN と表示）含まれていたが，これも 49.9%の予測確率で予測ができていた．実際，今回予測できた未知のマルウェアの攻撃セッションは $A_5 \to A_2 \to A_{10}$ であり，学習フェーズにおいてアラート A_{10} がマルウェアダウンロードとの相関が全くなかった．しかし，その前のアラート列 $A_5 \to A_2$ により，未知のマルウェアを予測できたということが明らかになった．

表 5 マルウェアダウンロードの誤検知率

予測確率の閾値	FNR	FPR
40%	1.67%	36.8%
80%	3.33%	30.4%

ここでマルウェアダウンロードの予測成功確率を整理すると，学習期間 (1) に対して，2011/1/25 は予測確率 0%が 1 回，49.9%が 1 回，55.5%が 1 回，88.4%が 1 回，残りは全て 90.6% であった．一方，2011/1/31 は全て予測確率が 90.6%であった．

また，提案手法の予測確率は，森らが Nexat で算出した予測確率 [5] よりも高くなっており，マルウェアダウンロードの予測には，Nexat よりナイーブベイズが向いていることが明らかになった．

7.2 誤検知について

本実験の誤検知は次の 2 つが考えられる．

(a) 予測確率が低いにも関わらずマルウェアダウンロードが発生した（False Negative Rate（FNR）が高くなるケース）

(b) 予測確率が高いにも関わらずマルウェアダウンロードが発生しなかった（False Positive Rate（FPR）が高くなるケース）

(a) の誤検知は致命的である．本実験では，2011/1/25 の 14:24:30 に honey001（攻撃元データ）で検知された BKDR_SDBOT.ZP(Rtsecar.exe) が全く予測出来なかった（予測確率がゼロ）．なぜなら，このマルウェアダウンロードでは，直前の前兆となる関連アラートが何もなかったためである．しかしながら，この 1 検体を除く残り 59 検体のマルウェアダウンロードを全て予測できている．表 5 にマルウェアダウンロードの誤検知率を示した．この実験結果より，本実験における FNR が低いことが分かる．さらに，FNR が低いということは，FNR= 1−TPR の関係より，予測成功確率を表す TPR が高いということが分かる．

(b) の誤検知はそれほど重要でないと考える．本提案手法は，Snort と組み合わせることによって，リアルタイムでアラートを出力することを想定している．そのため，たとえマルウェアダウンロードが実際に起こらなかったとしても，実際にマルウェアダウンロードの前兆となる攻撃がなされたことは事実である．したがって，これは誤検知と呼べない場合もあり，FPR が高めでも許容することを考える．実際，本実験結果では FPR が 3 割程度であり，許容範囲とみなしている．

7.3 経年変化について

本実験の学習時期は，(1) 直前の 1 週間，(2) 約 5 ヶ月前，(3) 約 10 ヶ月前の 3 期間である．直前の 1 週間で学習させた場合が最も予測精度が良かったのは当然と考えられ

© 2013 Information Processing Society of Japan

るが，約 10 ヶ月前の学習においても，十分予測可能であることが表 3 より明らかになった．

8. まとめ

本研究では，本提案手法と Snort を組み合わせることによって IDS のアラートの相関を導出し，マルウェアダウンロードがどの程度予測可能であるかを，マルウェア対策のための研究用データセット（CCC DATAset）を用いて評価・考察を行った．その結果，マルウェアダウンロードが 98.3%の確率で予測可能であることが実験的に示された．したがって，未知のマルウェアも含め，深い専門性がなくとも，IDS の関連アラートからマルウェアダウンロードを高い確率で予測できることが分かった．

今後の課題として，予測評価のデータが少ないことが挙げられる．今回の予測実験では，2011/1/25 と 1/31 の 2 日間のデータ（60 検体）を用いたが，出現したマルウェアはわずか 7 種類（未知のマルウェアを含む）であった．今後は，予測可能なマルウェアがどの程度であるのかの検証が必要である．

参考文献

[1] F. Cuppens and A. Miege: *Alert correlation in a cooperative intrusion detection framework*, In Proceedings of the IEEE Symposium of Security and Privacy, 202–215, 2002.
[2] 畑田充弘, 中津留勇, 秋山満昭: マルウェア対策のための研究用データセット 〜 MWS 2011 Datasets 〜, CSS/MWS, 2011.
[3] P. Ning, Y. Cui and D.S. Reeves: *Analyzing intensive intrusion alerts via correlation*, In Proceedings of the RAID, 74–94, 2002.
[4] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel and G. Vigna: *Nexat: a history-based approach to predict attacker actions*, In Proceedings of ACSAC, 383–392, 2011.
[5] 森俊貴, 面和成: Nexat を用いた攻撃予測に関する考察, SCIS2013, 3C4-3, 2013.
[6] C. Krugel, D. Mutz, W.K. Robertson and F. Valeur: *Bayesian event classification for intrusion detection*, In Proceedings of ACSAC, 14–23, 2003.
[7] N.B. Amor, S. Benferhat and Z. Elouedi: *Naive Bayes vs decision trees in intrusion detection systems*, In Proceedings of ACM SAC, 420–424, 2004.
[8] S. Benferhat, T. Kenaza and A. Mokhtari: *A Naive Bayes Approach for Detecting Coordinated Attacks*, In Proceedings of IEEE COMPSAC, 704–709, 2008.
[9] 田中達也, 佐々木良一: Snort ルールを組み合わせるボット通信検知方式の確率と改ざんサイトの自動検知システム DICE の改良, CSS2011, 266–271, 2011.

© 2013 Information Processing Society of Japan