Vol.2014-CSEC-65 No.19
Vol.2014-IOT-25 No.19
2014/5/23

ナイーブベイズを用いたDrive-by-Download攻撃予測の評価

安達 貴志[1]　面 和成[1]

概要：近年，Web サイトを閲覧したユーザにマルウェアをダウンロードさせる攻撃 Drive-by-Download による被害が増加している．この攻撃は，Web ブラウザやプラグインの脆弱性を悪用することにより，強制的にマルウェアをダウンロードさせる．対策として様々な取り組みが行われているが，攻撃の高度化により現段階ではあまり有効な手段がない．本研究では，Drive-by-Download 攻撃の過程において悪用される脆弱性に焦点を当てた攻撃予測を行う．悪用される脆弱性には，相互に影響を与える，特定の種類が用いられる等の傾向があるため，似た性質の脆弱性をグルーピングして攻撃予測を行う．我々の知る限り，本提案手法はグルーピングを攻撃予測に用いた初めての手法である．脆弱性のグルーピングには K-means++ 法，また攻撃予測にはナイーブベイズの 2 種類の機械学習を用いる．その結果，この脆弱性のグルーピングにより，攻撃予測の精度を向上させることができた．検証には，D3M データセットの 2010 年から 2013 年を用いた．

キーワード：Drive-by-Download，ナイーブベイズ，マルウェア，攻撃予測

Evaluation of Drive-by-Download attack prediction using Naive Bayes

Takashi Adachi[1]　Kazumasa Omote[1]

Abstract: Recenty, a damage by drive-by-download attacks is increasing, in which after the user browses a compromised website, a malware is automatically downloaded into her/his computer using exploited vulnerabilities. Although there are several countermeasures against drive-by-download attacks, effective one does not have been proposed so far. In this paper, we focus on attack prediciton using exploited vulnerability group. Exploited vulnerabilities can be partially related, hence we classify vulnerabilities into same group by similary charactersitics and predict attacks from such vulnerability groups. To the best of our knowlege, our proposed method is the first one to predict attacks using such grouping method. We use two kinds of machine learning algorithms, K-means++ for grouping vulnerabilities and Naive Bayes for prediting attacks. The accuracy of prediction are improved from the results of our evaluation. We use D3M dataset from 2010 to 2013.

Keywords: Drive-by-Download, Malware, Attack prediction, Naive Bayes

1. はじめに

近年，スマートフォン等の携帯型端末の普及に伴い，サイバー攻撃による被害が増加している．その中でもマルウェアを強制的にダウンロードさせる Drive-by-Download 攻撃が増加しており，2013 年の検知数ランキング [1] では，トップ 10 のうち 7 つがこの攻撃によるものである．この攻撃は，Web ブラウザやプラグインの脆弱性を悪用してマルウェアをダウンロードさせる．そのため，セキュリティパッチを当て脆弱性をなくすことが最も有効な予防手段であるが，[2] によると 80%のユーザがパッチを当てないため，他の方法を模索する必要がある．一般的に，マルウェアが 1 度コンピュータに侵入すると駆除が難しく，また侵

[1] 北陸先端科学技術大学院大学，石川県能美市旭台 1-1
Japan Advanced Institute of Science and Technologh， 1-1 Asahidai，Nomi，Ishikawa,923-1292 Japan

入されたことによるシステム全体への影響を調査する必要があり，非常に煩雑な処理が必要となるため，マルウェアの侵入を事前に防ぐ必要がある．

本研究では，Drive-by-Download 攻撃において，必ず脆弱性が悪用されるという点に注目し，この脆弱性を用いて攻撃予測を行う．悪用される脆弱性には，相互に影響を与える [3]，特定の種類が用いられる等の傾向がある．そのため，似た性質の脆弱性を同一グループに属すると考え，それらのクラスの関係から攻撃の予測を行う．我々の知る限り，本提案手法はグルーピングを攻撃予測に用いた初めての手法である．グルーピングには K-means++アルゴリズム，また攻撃予測にはナイーブベイズの 2 種類の機械学習を用いた．その結果，この脆弱性のグルーピングにより，攻撃予測精度を向上させることができた．

本稿の構成を示す．2 章では，本研究と関連する論文についての紹介を行う．3 章では，攻撃予測を行うために必要な準備ついて説明する．4 章では，使用する機械学習について説明する．5 章では，攻撃予測を行うための手法を提案する．6 章では，手法の手順や性能評価を示し，7 章にて考察を行う．そして，最後に 8 章で本研究のまとめについて述べる．

2. 関連研究

Drive-by-Download 攻撃に関する研究は，大きく 2 つのタイプに分けることができる．1 つ目は，能動的に攻撃を防ぐ方法である．悪性 URL を行うサイトをクローリングし，情報を保持することでその URL へのアクセスを防ぐ方法 [4] がある．しかし，この方法では，インターネットに存在する莫大なサイトを巡回することは不可能に近いため，現実的な手段ではない．2 つ目の方法として，受動的に攻撃を防ぐ方法である．リダイレクトを中心として解析し攻撃の予測を行う方法 [5] やアクセス先のページコンテンツを動的に解析することで防ぐ方法 [6]，また機械学習を用いて悪性 URL の IP アドレスを学習し，それを用いた悪性サイトの判断 [7] などの研究が行われている．しかしながら，コードの難読化等による攻撃の高度化により，あまり良い精度は得られていない．

一方，複数の IDS アラートの相関を用いた攻撃の予測手法に関する研究が報告されている [8–12]．特に，[11,12] では CCC DATASet に対してマルウェアのダウンロードの予測を行っており，ここでは経年変化についても議論されている．また，[13] では CCC DATAset に対して経年変化の検証を行っており，マルウェア感染検知に有効な特徴量が年々低下している傾向にあることが報告されている．しかしながら，このように経年変化に関する研究がいくつかなされているものの，著者らの知る限り，Drive-by-Download 攻撃予測において経年変化を考慮した手法は現在のところ見当たらない．

表 1　CVE-2009-0927

Factor	評価
Score	10.0
AccessVector	NETWORK
AccessComplexity	LOW
Authentication	NONE
Confidentiality	NONE
Integrity	COMPLETE
Availability	COMPLETE

3. 準備

3.1 D3M データセット [14]

D3M データセットとは，NTT セキュアプラフォーム研究所の高対話型の Web クライアントハニーポットで収集した Web 感染型マルウェアの観測データ群である．データ群には，マルウェア検体とサンドボックスでマルウェアを実行した際の通信データ，悪性 URL へアクセスした際のフルキャプチャーデータの 3 つが収録されている．

3.2 Wepawet [15]

Wepawet は，カルフォルニア大学にて開発されたオンラインのマルウェアアンパックツールである．このツールでは，悪性の可能性がある URL や PDF ファイル，Flash ファイルを指定することで，それらが悪性かどうかを判断し，悪性であれば悪用されている脆弱性識別子番号や JavaScript，シェルコードやダウンロードさせられる実行ファイル名を表示する．また，このツールでは同一 URL の過去の解析結果も得ることができるため，解析時点において悪性とされていない URL の解析結果も得ることができる．

3.3 脆弱性情報データベース

脆弱性情報データベースとは，脆弱性情報をデータベース化し，一般向けに公開しているもののことである．脆弱性情報データベースは，Common Vulnerabilities and Exposures（CVE），National Vulnerability Database（NVD）[16] 等がある．これらのデータベースには，脆弱性の危険度やシステムへの影響，また攻撃時の認証の可否等をの詳細な情報が記載されている．**表 1** は，NVD に記載されている CVE-2009-0927 の情報であり，**表 2** が公開されている脆弱性情報の一部である．表 1 の脆弱性は，脅威度が最高の 10.0 の評価がされており，またネットワーク上から脆弱性の悪用が可能であり，機密性・完全性・可用性のいずれに対しても多大なる影響を与えることができるとされている．このように，脆弱性情報データベースには，脆弱性の詳細な情報が記載されている．

表 2　用いる脆弱性情報

Factor	取りうる属性	説明
Score	0.0-10.0	脅威度
AccessVector	Local，AdjacentNetwork，Network	攻撃元区分
AccessComplexity	High，Medium，Low	攻撃条件の複雑さ
Authentication	Multiple，Single，None	攻撃前の認証要否
Confidentiality	None，Partial，Complete	機密性への影響
Integrity	None，Partial，Complete	完全性の影響
Availability	None，Patrial，Complete	可用性への影響

4. 機械学習アルゴリズム

本稿にて使用する機械学習アルゴリズムについて述べる．

4.1 ナイーブベイズ

ナイーブベイズは，特徴ベクトル間の独立性の仮定とベイズの定理を適用した手法である．クラス変数 y と独立特徴ベクトル $(x_1, ..., x_n)$ の関連は以下の通りになる．

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|n)}{P(x_1, ..., x_n)} \quad (1)$$

式（1）の分母は，クラス y に依存しておらず，また定数であるため，分母は考慮しなくてもよい．さらに，特徴ベクトルは独立していると仮定しているため，分子は以下のように簡単化できる．

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y)$$

ゆえに，式（1）は，以下の通りに変形される．

$$P(y|x_1, ..., x_n) = P(y)\prod_{i=1}^{n} P(x_i|y) \quad (2)$$

式（2）において，尤もらしさ（尤度）を最大化する（最大事後確率）ことで，特徴ベクトルからクラスの推定を行うことができる．

分類モデルとしてガウシアンモデルとベルヌーイモデルがある．入力ベクトルが 2 値の場合には，ガウシアンモデルよりベルヌーイモデルの方が精度が良くなる．本研究では，事前実験の結果から以下に示すベルヌーイモデルを用いる．

$$\prod_{i=1}^{n} P(x_i|y) = P(i|y)x_i(1 - P(i|y))(1 - x_i) \quad (3)$$

4.2 K-means++

教師なし学習クラスタリングアルゴリズムである．以下の手順でクラスタを作成し，クラスタリングを行う．

(1) 各データにランダムなクラスタを割り当てる
(2) 各クラスタの重心を計算する
(3) 各々のデータと各クラスタ間の距離を計算し，最も近い中心のクラスタに再割り当てを行う
(4) 全てのデータに対するクラスタ割り当てが変化しなくなるまで処理を繰り返す．変化しなければ処理を終了する

しかしながら，K-means 法は NP 困難であり，またクラスタとすべきでないものにもクラスタ割り当てを行う問題がある．そこで，開発されたのが K-means++法である．この方法では，データをランダムにクラスタに割り当てるプロセスが以下のように改良されている．

(1) 最初のクラスタ中心点をランダムに選択する
(2) 各々のデータと最近傍中心との距離を算出する
(3) 最も近い中心点までの距離が最大になるデータを次の中心点として採用する
(4) k 個のクラスタ中心が選ばれるまで（2）と（3）を繰り返す
(5) 選ばれたクラスタ中心を初期値として K-means 法を行う

5. 提案手法

一般的に，マルウェアが 1 度コンピュータに侵入すると駆除が難しく，また侵入されたことによるシステム全体への影響を調査する必要があり，非常に煩雑な処理が必要となるため，マルウェアの侵入を事前に防ぐ必要がある．Drive-by-Download 攻撃において，マルウェアはシステムの脆弱性を悪用して侵入してくることから必ず脆弱性を悪用すると考えられるため，脆弱性に焦点を当てた攻撃予測を行う．悪用される脆弱性には，関連や特定の傾向が用いられることが多い．そこで，同じ性質の脆弱性を同一のグループとして扱うことで，攻撃予測の精度を向上させることができると考えられる．よって，悪用された脆弱性をグルーピングし，攻撃予測を行う．我々の知る限り，本提案手法はグルーピングを攻撃予測に用いた初めての手法である．そして，脆弱性のグルーピングには教師なし学習のクラスタリングアルゴリズムである K-means++法を用い，また攻撃予測としてナイーブベイズの 2 種類の機械学習を使用する．

5.1 データ抽出フェーズ

5.1.1 脆弱性識別子番号の抽出

まず最初に，データセットから実際に Drive-by-Download 攻撃において悪用された脆弱性識別子番号の抽出を行う．

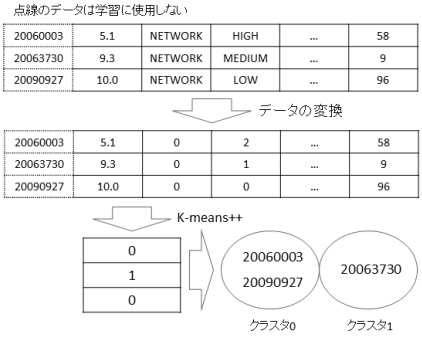

図 1 K-means++(クラスタ数 2) の例

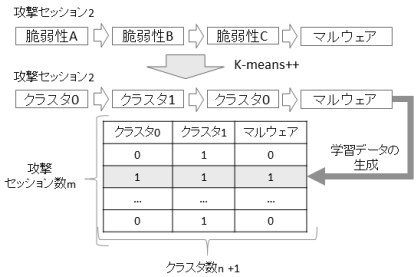

図 2 クラスタ数 2 における学習データの生成例

データセットには，悪性 URL リストが同梱されており，この URL を Wepawet を用いて解析を行う．

Wepawet は解析対象が悪性かどうかを解析し，悪性でかつ悪用されている脆弱性の識別子番号が分かれば，それを表示する．このツールはオンラインベースのため，解析結果は HTML ファイルとして得ることができる．そのため，HTML ファイルから脆弱性識別子番号を抽出する．ただし，十分な解析ができないファイルや URL も存在する．

5.1.2 脆弱性情報の抽出

5.1.1 節で抽出した識別子番号をもとに，脆弱性情報の詳細を抽出する．本研究では，NVD [16] の情報を用いる．NVD は，公開されている全脆弱性の情報を xml ファイルとして配布しているため，そのファイルから識別子をインデックスとして情報の抽出を行う．今回は，R 言語を用いて情報の抽出を行った．

抽出した識別子番号とその詳細情報を関連付けて 1 つのデータとする．本研究では，3.3 節にて述べた 7 つの指標に加え，悪用された回数の情報を付加して 1 つのデータとする．回数情報は，全 URL の解析結果から使用回数を求める．そのため，**表 3** に示す 9 次元のベクトルとなる．

5.2 学習フェーズ

5.2.1 第 1 段階

5.1.2 節にて抽出・作成したデータセットに対し，K-means++法を用いてグルーピングを行う．脆弱性は A→B→C のように 1 度の攻撃で複数悪用されることが多い．そこで，学習データを生成する際に，A→B，B→C のような脆弱性の組み合わせも考えることができる．詳細は 7.2 節にて述べるが，予測精度が下がるため，今回は組み合わせを考慮しない．

第 1 段階の学習アルゴリズム及びその実行結果の例を Algorithm1 と図 1 にそれぞれ示す．最初に，クラスタ数（ClusterCardinalNumber）や K-means++法にて使用する初期値（ClusterInit）を決める．次に，グルーピングに必要のない脆弱性識別情報を除去（ExceptCVEID）し，各々の脆弱性が悪用された回数をデータに追加（AddDetectionCardinalNumber）する．データは，属性情報が文字列のため，アルゴリズムに適用することができない．そこで，文字列を数値に変換（StringToInteger）する．そして，変換したデータとクラスタ数，初期値を K-means++（KmeansEx）に引数として与える．脆弱性情報にもとづいてそれぞれのデータがどのクラスタに属するかを決定し，同一クラスタに属しているものが似た性質を持つ脆弱性となる．そのため，出力として各々の脆弱性が属するクラスタ番号（CVEID.ClusterOrdinalNumber）を得ることができる．

Algorithm 1 Grouping

Input: CVEInformationList
Output: CVEID.ClusterOrdinalNumber

$ClusterCardinalNumber \Leftarrow 10, ClusterInit \Leftarrow 1$
$ClusterData \Leftarrow$ ExceptCVEID($CVEInformationList$)
$ClusterData \Leftarrow$ AddDetectionCardinalNumber($ClusterData$)
$ClusterData \Leftarrow$ StringToInteger($ClusterData$)
$CVEID.ClusterOrdinalNumber \Leftarrow$
KmeansEx($ClusterData, ClusterCardinalNumber, ClusterInit$)
return ($CVEID.ClusterOrdinalNumber$)

5.2.2 第 2 段階

グルーピングした結果と各々の URL の解析結果から学習データを生成し，ナイーブベイズを適用して学習を行う．学習データとは，ナイーブベイズで攻撃予測モデルを構築するためのデータであり，攻撃セッション数 m とクラスタ数 n，マルウェアダウンロードの有無を追加した $m \times (n+1)$ の行列となる．アルゴリズム及び実行結果の例を Algorithm2 と図 2 にそれぞれ示す．

K-means++法によって算出した各々脆弱性のクラスタ番号（CVEID.ClusterOrdinalNumber）及び各々の URL の解析結果（AnalysisResultOfURL）を用いる．ある脆弱性が悪用された場合には，それが所属するクラスタの脆弱性が悪用されたとし，その情報をデータに記載する．ただし，今回はベルヌーイモデルを用いているため，1 つの攻撃にて同じクラスタの脆弱性が悪用されたとしても 2 回悪用されたという情報は学習することができないことに注意する．そして，最終的にマルウェアがダウンロードされたかどうかを表す 2 値情報を学習データに追加する．これを全ての URL の解析結果に対し行い，学習データ（DataForNaiveBayes）を作成する．

表 3　抽出した脆弱性情報の例

CVEID	Score	AccessVector	AccessComplexity	Authentication	Confidentiality	Integrity	Availability	Num
20060003	5.1	NETWORK	HIGH	NONE	PARTIAL	PARTIAL	PARTIAL	58
20063730	9.3	NETWORK	MEDIUM	NONE	COMPLETE	COMPLETE	COMPLETE	9
20090927	10.0	NETWORK	LOW	NONE	COMPLETE	COMPLETE	COMPLETE	96

Algorithm 2 Making Learning Data

Input: CVEID.ClusterOrdinalNumber, AnalysisResultOfURL
Output: DataForNaiveBayes

```
for CVEID in AnalysisResultOfURL.ExploitedCVE do
  if CVEID.ClusterOrdinalNumber not Used then
    DataForNaiveBayes ⇐ true
  else
    DataForNaiveBayes ⇐ false
  end if
end for
return (DataForNaiveBayes)
```

5.3　予測フェーズ

5.2 節に学習したモデルに対し，テストデータを適用して実際に攻撃予測を行う．予測結果は，マルウェアダウンロードが起きる確率と起きない確率の 2 つが結果として得られる．しかし，予測確率が低い場合にはその結果の信憑性が低いと考えられるため，ある一定の閾値未満の確率はマルウェアのダウンロードが起こらないとする．

6.　実験

6.1　目的

5 章にて述べた提案手法の有効性を，D3M データセットを用いて検証を行う．テストデータを用いて予測確率を算出し，6.3 節の 5 つの指標を用いて，評価を行う

6.2　概要

本実験では，以下を学習データ及びテストデータとして用いた．

- 学習データ
  D3M データセットの 2010 年と 2011 年の 2 年分
- テストデータ
  D3M データセットの 2012 年と 2012 年の 2 年分

学習データを用いて攻撃予測を行うためのモデルを構築する．そして，テストデータを用いて性能評価を行う．

6.3　性能評価

性能評価には，D3M データセットの 2012 年と 2013 年分のデータを用いる．性能評価の指標には TPR，TNR，FNR，FPR，Accuracy の 5 つを用いる．

TPR（True Positive Rate）は，マルウェアダウンロードが発生すると予測し，実際にダウンロードが発生した確率である．また，TNR（True Negative Rate）はマルウェアダウンロードが発生しないと予測し，実際にダウンロードが発生しなかった確率を表す．マルウェアダウンロードの予測確率が低いにも関わらず，ダウンロードされる確率（False Negative Rate）と予測確率が高いにも関わらず，ダウンロードされなかった確率（False Positive Rate）の 2 つの指標が重要である．その中でも特に FNR は重要であるため，FPR を犠牲にしてでも FNR は低くする必要がある．そこで，予測確率が一定の閾値を超えていない場合は，その予測結果を採用しないことで FNR を下げる．

6.4　手順

6.4.1　学習準備

脆弱性は，Wepawet によって抽出するが，中には脆弱性が悪用されているにも関わらずマルウェアのダウンロードが行われていないものやマルウェアのダウンロードが行われているにもかかわらず脆弱性の抽出ができないものも含まれる．現実世界にて適用する際には，必ずしも脆弱性が検出されるとは言えないため，それらに対し特別な処理はせずに学習データに入れる．

また，グルーピングを行うために必要となる脆弱性情報を NVD より抽出を行っておく．

6.4.2　学習

5.2.1 節に記載の第 1 段階の学習により脆弱性のグルーピングを行い，5.2.2 節にて記載した第 2 段階の学習により，ナイーブベイズを適用するためのデータを生成する，グルーピングでは，事前実験の結果が最も良かったクラスタ数 10 で行う．そして，ナイーブベイズを適用し，攻撃予測モデルを構築する．

6.4.3　予測

6.4.2 節にて生成した攻撃予測モデルに対してテストデータを適用し，予測確率の算出を行う．

6.5　結果

表 4 は，通常の攻撃予測とグルーピングを行った場合の攻撃予測精度である．最も重要な値である FNR は，グルーピングによって約 50%下がっていることがわかる．また TPR 及び FNR が向上し，Accuracy も僅かながら上昇しているが，FPR 及び TNR は下がっている．

7.　考察

7.1　脆弱性のグルーピングについて

Drive-by-Download 攻撃は，年によって悪用される脆弱性の種類が大きく変わることがあるため，グルーピングな

表 4　グルーピングの有無

	グルーピングなし	グルーピングあり
TPR	0.59	0.78
FPR	0.00	0.10
FNR	0.41	0.21
TNR	1.00	0.91
Accuracy	0.80	0.85

表 5　組み合わせの有無

	無	有
TPR	0.59	0.40
FPR	0.00	0.15
FNR	0.41	0.60
TNR	1.00	0.85
Accuracy	0.80	0.63

しの通常の攻撃予測では精度が期待できない．しかし，脆弱性自体が異なっていたとしても，攻撃者に好まれる脆弱性の性質はあまり変わらないため，今回の実験のように脆弱性をグルーピングすることで攻撃予測の精度が向上したと考えられる．

7.2 脆弱性の組み合わせについて

アラートの相関関係を用いた攻撃予測 Nexat [17] では，A→B→C→ というアラートの流れにおいて，A→B や B→→C の組み合わせを学習することで予測精度を向上させている．脆弱性の悪用においても同様のことを行うことで予測精度の向上が見込まれるため，組み合わせを用いた追加実験を行った．

しかしながら，組み合わせを用いることによって予測精度が低下するという結果が得られた（**表 5**）．この結果から，攻撃予測においても脆弱性の一連のプロセスを考慮した攻撃予測を行うべきだと考えられる．

8. まとめ

本研究では，脆弱性及びそのグルーピングを用いた Drive-by-Download 攻撃の予測手法を提案し，D3M データセットを用いて手法の有効性の検証を行った．表 4 の実験結果より，攻撃予測の精度向上にはグルーピングを用いることが有効な手段であることが明らかになった．

しかし，依然として FNR が高く，また TPR も十分な数値だとは言えない．そこで，今後の課題として，グルーピングに用いる脆弱性情報の取捨選択やデータの精査，また別の機械学習アルゴリズムを用いた攻撃予測などが考えられる．

参考文献

[1] 本城信輔：マカフィー，2013 年におけるサイバー脅威の分析結果を発表，マカフィー株式会社（オンライン），入手先 ⟨http://www.mcafee.com/japan/security/monthly/PC201312.asp⟩（参照 2014-4-10）．

[2] D.Danchev: Research:80Flash/Acrobat, ZDNet (online), available from ⟨http://www.zdnet.com/blog/security/research-80-of-web-users-running-unpatched-versions-of-flashacrobat/4097⟩ (accessed 2013-10-10).

[3] Frigault, M. and Wang, L.: Measuring Network Security Using Bayesian Network-Based Attack Graphs, *32nd Annual IEEE International Computer Software and Applications Conference*, pp. 698–703 (2008).

[4] Invernizzi, L. and Comparetti, P. M.: EVILSEED:A Guided Approach to Finding Malicious Web Pages, *IEEE Symposium on Security and Privacy* (2012).

[5] 寺田剛陽，古川忠延，東角芳樹，鳥居悟：検知を目指した不正リダイレクトの分析，*CSS2010*, pp. 765–770 (2010).

[6] Priya, M.，Sandhya, L.，Ciza, T. : A Static Approach to Detect Drive-by-download Attacks on Webpages, *International Conference on Control Communication and Computing* (2013).

[7] Chiba, D.，Tobe, K.，Mori, T.，Goto, S. : Detection Malicious Websites by Learning IP Address Features, *IEEE/IPSJ 12th International Symposium on Applications and the Internet* (2012).

[8] Cuppens.F, M.: Alert Correlation in a Cooperative Intrusion-Detection Framework, *IEEE Symposium on Security and Privacy* (2002).

[9] Herve Debar, A. W.: Aggregation and Correlation of Intrusion-Detection Alerts, *4th International Sysmposium on Recent Advances in Intrusion Detection* (2001).

[10] Ning.P, Cui.Y, R.: Constructing Attack Scenarios through Correlation of Intrusion Alerts, *International Symposium on the Recent Advances in Intrusion Detection* (2002).

[11] 森俊貴，面和成：ネットワーク通信アラートを利用した攻撃予測に関する評価・考察，*SCIS 2014* (2014).

[12] 面和成：ナイーブベイズを用いた攻撃予測に関する評価・考察，*CSEC-62* (2013).

[13] 川本研治，市田達也，市野将嗣，畑田充弘，小松尚久：マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察，*CSS 2011* (2011).

[14] 神薗雅紀，畑田充弘，寺田真敏，秋山満昭，笠間貴弘，村上純一：マルウェア対策のための研究用データセット 〜MWS Datasets 2013〜，*MWS2013* (2013).

[15] WepawetTeam: Wepawet, The Regents of the University of California (online), available from ⟨http://wepawet.iseclab.org/⟩ (accessed 2013-10-20).

[16] NIST: National Vulnerability Database, NIST (online), available from ⟨http://nvd.nist.gov/⟩ (accessed 2013-10-20).

[17] Cipriano, C.，Zand, A.，Houmansadr, A.，Kruegel, C.，Vigna, C. : Nexat: A histry-Based Approach to Predict Attacker Actions, *AC-SAC 2011*, pp. 383–392 (2011).