Vol.2016-DPS-166 No.25
Vol.2016-CSEC-72 No.25
2016/3/4

エクスプロイトキットで利用される文字列特徴を用いた悪性 URL 検出手法の提案

佐藤祐磨[†1]　中村嘉隆[†2]　高橋修[†2]

概要：近年，Drive-by Download 攻撃の被害が増えている．Drive-by Download 攻撃は特定のサイトに訪れたユーザにマルウェアをダウンロード，実行させる攻撃である．Drive-by Download 攻撃においてエクスプロイトキットが利用される攻撃が見られる．そこで URL のパス・クエリ部のパターンや特徴を基に，エクスプロイトキットで利用される悪性 URL の検出手法を提案する．

キーワード：Drive-by Download 攻撃，エクスプロイトキット，URL

A Proposal for Malicious URLs Detection based on Features of Strings Used in Exploit Kits

YUMA SATO[†1]　YOSHITAKA NAKAMURA[†2]　OSAMU TAKAHASHI[†2]

1. はじめに

近年，Web の普及に伴い，Drive-by Download 攻撃が巧妙化している．Drive-by Download 攻撃は Web 上を介して行われるサイバー攻撃であり，Web を利用するユーザの PC にマルウェアをダウンロードさせる攻撃である．図 1 に IBM TOKYO SOC レポート [1] で報告されている Drive-by Download 攻撃の検知件数を示す．2015 年上半期においては，2740 件の攻撃が観測されており，2013 年からのどの半期においても，800 件以上の攻撃が観測されている．

Drive-by Download 攻撃 [2] は，攻撃者が正規の Web サイトの改ざん，または，不正広告を表示によって，その Web サイトを閲覧したユーザを攻撃サイトに誘導し，マルウェアに感染させる攻撃である．一般にこの攻撃では，Web を利用するユーザの使用しているソフトウェアの脆弱性を突いてマルウェアがユーザ端末にダウンロードされる．マルウェアのダウンロードは，秘密裏で行われるため，攻撃中にユーザが気付くことは難しい．このような Drive-by Download 攻撃には，スクリプトコードが利用される．Drive-by Download 攻撃に利用されるスクリプトコードは，攻撃者によって難読化されていることが多く，企業に設置されている IPS (Intrusion Prevention System) に記憶された攻撃パターンに合致しないように，また，第三者である攻撃解析者による攻撃解析を困難にするために，何らかの細工が施されている事が多い．このように Drive-by Download 攻撃は近年巧妙化する傾向にある．

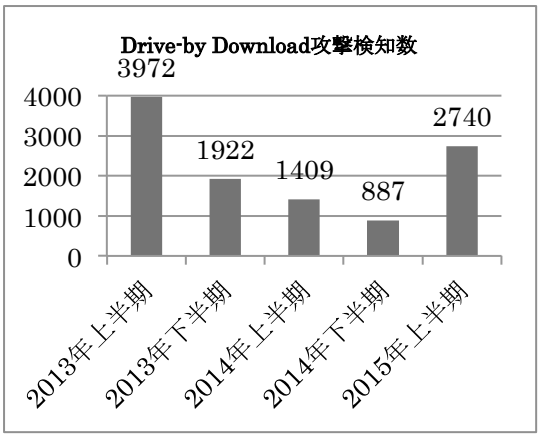

図 1　Drive-by Download 攻撃検知数

Drive-by Download 攻撃の典型的なフローは図 2 のようになっている．攻撃者は，正規 Web サイトの Web ページから攻撃者が用意する攻撃 Web サイトへのリダイレクトを目的として，正規の Web サイトを改ざんする．改ざんされた Web ページにアクセスしたユーザは，このリダイレクトにより，攻撃者が用意した攻撃サイトへ誘導される．一般に，攻撃の検出を困難にするために，このリダイレクトは複数用意されていることが多い．リダイレクトが複数ある理由は，攻撃者が攻撃検出を回避あるいは困難にするためと考えらえる．攻撃サイトでは，ユーザの使用する OS，ブラウザ，ブラウザのアドオンの脆弱性を突く攻撃が行われ，ユーザの端末の制御が攻撃者に奪われる．その後，ユーザはマルウェア配布サイトへ誘導され，悪意あるソフトウェアをダウンロードさせられる．

†1 公立はこだて未来大学大学院システム情報科学研究科
Graduate School of Systems Information Science, Future University Hakodate
†2 公立はこだて未来大学システム情報科学部
School of Systems Information Science, Future University Hakodate

ⓒ2016 Information Processing Society of Japan

Vol.2016-DPS-166 No.25
Vol.2016-CSEC-72 No.25
2016/3/4

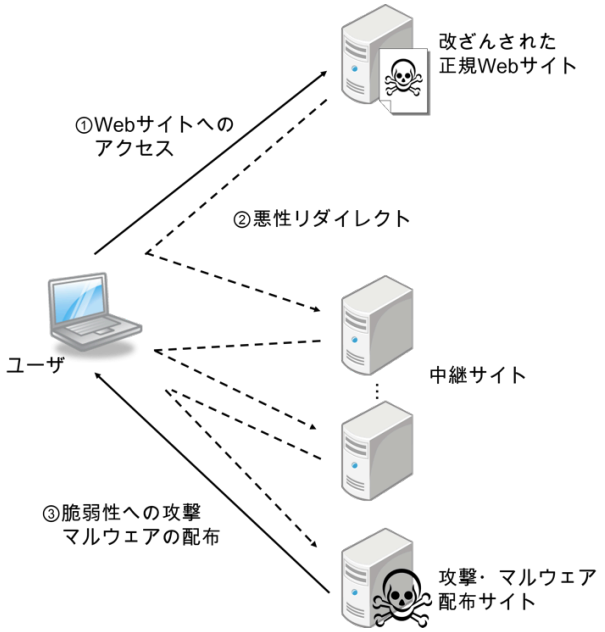

図 2　Drive-by Download 攻撃のフロー

近年では，Drive-by Download 攻撃にはエクスプロイトキットが用いられることが多くなってきている [3]．エクスプロイトキットは，様々な脆弱性を突く攻撃が可能なエクスプロイトコードをパッケージ化したツールキットである．新たに発見された脆弱性に対する攻撃コードが随時追加されており，GUI ベースで操作・管理が可能なエクスプロイトキットが登場するなど，専門知識のない人間にも容易に効果的な攻撃を行うことが可能になっている．

Drive-by Download 攻撃では，フィンガースプリクティングというユーザの使用する環境情報を判別する手法が利用されている．エクスプロイトキットの攻撃コードには，複数の攻撃コードが含まれていて，環境情報に合った脆弱性を突く攻撃を行うことが可能であるため，ユーザが使用するブラウザや，ソフトウェアに 1 つでも脆弱性が存在すれば，脆弱性を突く攻撃が成功する．それにより，ユーザ端末にマルウェアがダウンロードされる．

Drive-by Download 攻撃で利用されるエクスプロイトキットには，複数種類があり，エクスプロイトキット毎に特徴がある．攻撃に用いられるエクスプロイトキットを判別することができれば，Drive-by Download 攻撃のパターンを検出することができ，攻撃を防ぐことができる可能性がある．そこで，本稿では，URL のパス・クエリ部のパターンや特徴を基に，エクスプロイトキットで利用される悪性 URL の検出手法を提案する．

2. 関連研究

Drive-by Download 攻撃の検出について様々な手法が提案されている．

Google Safe ブラウジングなど，ブラックリスト型の手法は，Drive-by Download 攻撃の悪性 Web サイトの攻撃を防ぐために一般的に使用されている．ブラックリストの作成は，基本的に 3 つの処理によって行われる．まず，Web クローラを用いて，Web ページに含まれるリンクを辿ることで，悪性と考えられる URL を収集する．次にその URL の中から悪性と考えられる URL に対して，解析者が実際に Web ページにアクセスし，HTML，JavaScript のコードを評価し，点数付けする．この点数を用いて，Web ページが悪性であるかどうか判断し，ブラックリストを作成する．効率良く Web 空間をクロールすることで，コストを抑えてブラックリストを作成する手法が提案されている [4]．

しかし，Drive-by Download 攻撃で利用されるドメイン名は，頻繁に変更が行われる．悪性 Web サイトのドメイン名は，1 カ月以内で 4 割，6 カ月で 8 割が利用されなくなるという調査がある [5, 6]．攻撃者が，攻撃サイトのドメイン名を変更した場合，ブラックリスト型の手法では，全ての攻撃サイトを網羅することが困難であり，さらにブラックリストの更新・管理にコストがかかるという問題があるため，悪性 Web サイトの完全な検出に利用するには限界がある．

また，ブラックリストを使用しない攻撃検出手法として，攻撃スクリプトコードを分類し，頻出文字の特徴を観測することで，Drive-by Download 攻撃を検出する手法 [7] や，ユーザの Web 通信ログを監視し，Drive-by Download 攻撃で使用される悪性サイトを検出・通報するフレームワークを使用した手法や HTTP ヘッダの解析をして攻撃検出する手法 [8] がある．

3. 提案手法

3.1 基本的なアイディア

Drive-by Download 攻撃で利用される URL には良性 Web ページの URL よりも平均文字列長が長いという特徴があり，エクスプロイトキットを用いた攻撃で生成される URL は正規表現で検出できる特徴があることが知られている．また，利用したエクスプロイトキットごとにも特徴があることが知られている[9]．

このようなエクスプロイトキットで生成される URL のパス・クエリを利用して，悪性 URL を検出する．ドメイン名が頻繁に変更されても，使用されるエクスプロイトキットが同じであれば，パス・クエリの特徴に変化はなく，ドメイン名が短期間で変更されても，URL パス・クエリを元に悪性 URL を検出することができると考えられる．そこで，悪性 URL に多く見られる特徴を元に，URL パス・クエリをベクトル化して表現する．このベクトルを利用して決定木を作成する．作成した決定木によって，エクスプロイトキットで生成された URL パス・クエリと良性 URL パス・クエリの分岐が作成される．これを用いて悪性 URL を検出

ⓒ2016 Information Processing Society of Japan

する．

3.2 URL パス・クエリ

URL パス・クエリは URL に含まれるパスと URL に含まれるクエリを “?” で結合したものと定義する．例として，URL “http://www.example.com/dir/file.html?key=value” において，パスは，“/dir/file.html”であり，クエリは，“key=value” となる．つまり，この URL の URL パス・クエリは，“ /dir/file.html?key=value” となる．

3.3 決定木

決定木は，入力データがあらかじめに定められたカテゴリのどれに当てはまるのかを分類するための装置である．決定木である木構造の分類器を作成することを決定木の学習という．決定木は，葉ノード，中間ノード，根ノード，から構成され，葉ノードは，分類結果となるカテゴリを表し，葉ノードを除いた中間ノード，根ノードは，入力データに対するテストを表す．根ノードから葉ノードまで，テストを繰り返すことで，入力データがどのカテゴリに当てはまるか決定することができる．

Weka (Waikato Environment for Knowledge Analysis) は，ニュージーランドのワイカト大学で開発されたオープンソースの機会学習ツールである [10] ．Weka は，データの前処理，データの可視化などができる．様々な機会学習ツールが含まれている．

本稿では，決定木の作成に Quinlan が考案した決定木学習アルゴリズム C4.5 [11] を Weka に実装した J4.8 アルゴリズムを使用する．

3.4 悪性 URL の検出手法

3.4.1 検出手順

良性 Web ページにアクセスした通信データから通信で発生した URL を抽出する．また，エクスプロイトキットの攻撃通信データから通信で発生した URL を抽出する．これらの抽出した良性，悪性 URL から URL パス・クエリのみを抽出し，ベクトル化する．このベクトルを用いてとして決定木を作成する．アルゴリズム C4.5 を使用して，悪性 URL を検出する決定木を作成する．決定木の分岐を用いて，URL が悪性であるかどうかを判別する．

3.4.2 URL パス・クエリのベクトル化

抽出した URL パス・クエリは，表 1 の項目でベクトルに変換する．

(1) , (2) については，L. Xu らが，良性 URL 長は，平均 18.23，悪性 URL 長は，平均 25.11 と悪性 URL は自動で動的な長いランダムな URL が利用されると述べている [12] . また，悪性 URL に含まれる特殊記号の数が平均 3.36，良性 URL に含まれる特殊記号の数が平均 2.93 で悪性 URL に特殊記号が，良性 URL に比べて多く含まれると述べている．これらの特徴から，悪性 URL パス・クエリも良性 URL パス･クエリよりも長く，特殊記号の数が多いと考えられる．

(3) については，J. Ma らが，パス文字列の平均・最長文字列長を悪性判定要素としている． (1) からクエリ長も長くなると考えられる．また， (4) について，J. Ma らは，URL に含まれる数字の数を悪性 URL の判定要素にしている [13] ．URL パス・クエリにも有効的であると考えられる．

(5) , (6) , (7) については， (1) によって，文字列長が増えると悪性と考えられるため，キーの数が多くなればなるほど，文字列が多くなると考えられる．また，文字列の数も多くなると考えられる．

Drive-by Download 攻撃では，リダイレクトが利用されるため，クエリにリダイレクト先 URL が含まれる場合があると考えられる．そこで (8) の “http” を含むかどうかを成分要素に追加する．また (9) についても同様に，リダイレクト先の Web サイトの場所がクエリに含まれる場合があると考えられる．

表 1　ベクトルの成分

番号	ベクトルの成分要素	値
(1)	URL パス・クエリ文字列長	整数
(2)	特殊記号が含まれる数	整数
(3)	パス文字列長	整数
(4)	数字が含まれる数	整数
(5)	クエリ文字列長	整数
(6)	文字列が含まれる数	整数
(7)	クエリに含まれるキーの数	整数
(8)	“http” が含まれる	1, 0
(9)	IP アドレスが含まれる	1, 0

4. 実験・評価方法

4.1 実験方法

実験は，URL パス・クエリのみのテキストファイルに抽出し，そのテキストファイルを使用する．本実験はユーザの Web ブラウザの操作を考慮せず，URL クエリ・パスのみを対象とする．

実験には，決定木の作成をする必要がある．決定木は実験データを用い，エクスプロイトキットの種類と良性を分類する決定木を作成する．決定木の作成は，アルゴリズム C4.5 を用いる．実験では，Weka の J48 を使用する．

実験データに対し，作成した決定木を用いて交差検証を行うことで，提案手法の評価を行う．

4.2 実験データ

ここでは，本提案手法の評価を行うために使用する実験データについて説明する．

実験には，悪性データと良性データを使用する．このデータは決定木を作成するために利用するデータである．

悪性データは，Malware-Traffic-Analysis.net にある 2013 年から 2015 年に含まれる PCAP 形式のファイルを利用する

ⓒ2016 Information Processing Society of Japan

[14] ．本実験では，この PCAP 形式のファイルから HTTP 通信のリクエスト URL を抽出し，抽出した URL パス・クエリを悪性データとして使用する．URL クエリ・パスの件数は，7212 件である．エクスプロイトキットの分類は，ファイル名を利用し，分類する．悪性データから抽出したエクスプロイトキットの種類を以下に示す．

- Angler Exploit Kit
- Blackhole Exploit Kit
- Cool Exploit Kit
- Dotkachef Exploit Kit
- Fiesta Exploit Kit
- Flashpack Exploit Kit
- Goon Exploit Kit
- Hello Exploit Kit
- KaiXin Exploit Kit
- Magnitude Exploit Kit
- Neutrino Exploit Kit
- Nuclear Exploit Kit
- Rig Exploit Kit
- Styx Exploit Kit
- Sweet Orange Exploit Kit

良性データは，DMOZ を利用する [15] ．DMOZ は世界中のボランティアエディタによって構築・管理されている世界最大の Web ディレクトリである．DMOZ にインデックスされている Web ページの URL をランダムに 300 件抽出する．ランダムに抽出した 300 件の URL にアクセスし，発生するリクエスト URL の URL パス・クエリの中からランダムで 2368 件の URL パス・クエリを本実験で使用する良性データとする．

4.3 評価方法

本実験では，真陽性率 (TP) ，偽陰性率 (FN) ，真陰性率 (TN) ，偽陽性率 (FP) ，全体検出率の 5 つの評価項目によって評価する．

悪性データは，Web ページごとに評価を行う．任意の Web ページへの 1 リクエストに含まれる URL パス・クエリを 1 件とする．真陽性は，マルウェアの URL パス・クエリを悪性としてみなすことと定義する．検出した悪性 URL パス・クエリと，エクスプロイトキットの種類が異なる場合でも，悪性 URL パス・クエリを悪性とみなしたことと判断する．真陽性率は，マルウェアのダウンロードが発生したセッション全件において，真陽性の件数の割合である．真陽性率は，計算式 (1) で評価する．

$$\text{真陽性率} = \frac{\text{悪性 URL パス・クエリを悪性とみなした数}}{\text{悪性通信データの URL の数}} \quad (1)$$

偽陰性は，悪性データの URL パス・クエリを良性とみなすことと定義する．偽陰性率は，悪性データ全件において，偽陰性の件数の割合である．偽陰性率は，計算式 (2) で評価する．

$$\text{偽陰性率} = \frac{\text{悪性 URL パス・クエリを良性とみなした数}}{\text{悪性通信データの URL の数}} \quad (2)$$

良性データは，Web ページごとに評価を行う．任意の Web ページへの 1 リクエストに含まれる URL パス・クエリを 1 件とする．真陰性は，良性 Web ページの URL パス・クエリを良性とみなすことと定義する．真陰性率は，良性通信データ全リクエストにおいて，真陰性の件数の割合である．真陰性率は，計算式 (3) で評価する．

$$\text{真陰性率} = \frac{\text{良性 URL パス・クエリを良性とみなした数}}{\text{良性データの URL の数}} \quad (3)$$

偽陽性は，良性 Web ページの URL パス・クエリを悪性とみなすことと定義する．偽陽性率は，良性通信データの全リクエストにおいて，偽陽性の件数の割合である．偽陽性率は，計算式 (4) で評価する．

$$\text{偽陽性率} = \frac{\text{良性 URL パス・クエリを悪性とみなした数}}{\text{良性データの URL の数}} \quad (4)$$

全体検出率は，悪性データ全件と良性データのリクエスト全件において，真陽性の件数と真陰性の件数の割合である．全体検出率は (5) の計算式で評価する．

$$\text{全体の検出率} = \frac{\text{真陽性の件数} + \text{真陰性の件数}}{\text{悪性データの件数} + \text{良性データの件数}} \quad (5)$$

ⓒ2016 Information Processing Society of Japan

表 2　エクスプロイトキット毎の検出結果

			分類された結果															
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
エクスプロイトキット	A	Angler	1572	0	0	1	9	3	5	0	2	77	9	100	0	3	0	166
	B	Blackhole	0	25	0	0	0	0	3	0	0	0	0	0	0	0	0	11
	C	Cool	0	0	5	0	0	2	0	0	0	0	0	0	0	0	0	11
	D	DotKachef	3	0	0	36	0	0	3	0	0	0	0	0	0	0	0	12
	E	Fiesta	10	0	0	0	889	2	12	0	0	34	2	37	6	0	0	79
	F	Flashpack	1	0	2	0	3	139	3	0	0	1	15	5	0	0	2	54
	G	Goon	12	0	0	1	9	1	105	0	1	12	2	26	11	0	3	42
	H	Hello	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	3
	I	KaiXin	3	0	0	0	0	0	0	0	3	0	0	1	0	0	0	11
	J	Magnitude	86	0	0	0	27	1	15	0	0	859	6	54	6	2	0	72
	K	Neutrino	25	0	0	0	4	11	3	0	0	11	237	57	0	0	2	58
	L	Nuclear	73	0	0	0	28	2	8	0	0	55	30	1013	5	0	2	98
	M	Rig	5	0	0	1	12	0	9	0	0	7	1	17	304	3	0	28
	N	Styx	5	0	0	0	2	0	0	0	0	6	0	1	2	70	0	49
	O	Sweet Orange	5	0	0	0	3	0	3	0	0	1	0	4	2	0	118	98
良性	P	良性データ	97	3	4	5	53	26	19	1	0	61	37	82	13	13	20	1934

5. 結果・考察

5.1 検出結果

交差検証のクラスごとに分類された精度は，76.36%であった．また，エクスプロイトキット毎の分類結果を表 2 に示す．作成された決定木を用いて，良性と悪性の URL パス･クエリを分類した結果を表 3 に示す. 表 2 の結果より，データ件数の多いエクスプロイトキットの URL パス･クエリを用いた場合に高い検出精度を達成している. その反面，データの件数が少ないエクスプロイトキットの検出精度は，低くなっていることがわかる．

表 3　作成した決定木で分類した検出結果

評価項目	検出率
真陽性率 (TP)	89.02%
偽陰性率 (FN)	10.98%
真陰性率 (TN)	81.67%
偽陽性率 (FP)	18.33%
全体の検知率	87.20%

5.2 考察

提案手法では，エクスプロイトキットで発生する URL パス・クエリに変化がなければ，悪性 Web サイトのドメイン名が頻繁に変更されても，悪性 Web ページの URL を検出できるという長所がある．

エクスプロイトキットのデータが多いものほど，検出精度が高い. 悪性の URL パス・クエリの検出精度を高くするためには，エクスプロイトキットによるデータを多数収集する必要がある．

しかし，提案手法では，ベクトル化が URL からしか行わないため，良性データと悪性データのベクトルが同一になることが考えられる. それにより，URL パス・クエリが “/” のみの場合などの，良性データと悪性データの両方に含まれる URL パス･クエリなど良性データのベクトルと悪性データのベクトルが同一になる場合には，作成された決定木によって，良性データまたは，悪性データのどちらかに判別されてしまう可能性がある．この問題に対して，ユーザの意図とリダイレクトの回数を考慮し，最終的にダウンロードさせられるマルウェアのみ悪性と判別できる手法が必要であると考えられる. また，攻撃で発生する URL に含まれるファイル名をエクスプロイトキットごとに分類し，正

ⓒ2016 Information Processing Society of Japan

規表現や文字列一致を利用して判別すると真陽性がさらに向上すると思われる.

6. おわりに

本稿では，Drive-by Download 攻撃においてエクスプロイトキットを利用した攻撃で発生する URL パス・クエリの特徴を利用して悪性 URL を検出する手法を提案した. 本手法は，実際の攻撃通信データから抽出した URL パス・クエリの特徴から決定木を作成し, 決定木を元に悪性 URL を検出する. 真陽性率は，89.02%，真陰性率は，81.67%となった.

今後は, 攻撃で発生する URL に含まれるファイル名をエクスプロイトキットごとに細かく分類し，正規表現や文字列一致を利用して，悪性 URL を検出する手法を考案・提案する必要がある.

参考文献

[1] IBM. “Tokyo SOC 情報分析レポート”. http://www-935.ibm.com/services/jp/ja/it-services/soc-report/
[2] 独立行政法人情報処理推進機構. “コンピュータウイルス・不正アクセスの届出状況 [2010 年 11 月分] について”. https://www.ipa.go.jp/files/000016351.pdf
[3] トレンドマイクロ.“2015 年第 3 四半期 セキュリティラウンドアップ”. http://www.trendmicro.co.jp/cloud-content/jp/pdfs/security-intelligence/threat-report/pdf-sr2015q3-20151119.pdf?cm_sp=threat-_-sr2015q2-_-lp-btn
[4] L. Invernizzi. “EvilSeed: A Guided Approach to Finding Malicious Web Page,” Proceedings of the 2012 IEEE Symposium on Security and Privacy, pp.428-442, 2012.
[5] M. Akiyama, T. Yagi and M. Itoh. “Searching structural neighborhood of malicious URLs to improve blacklisting,” Proceedings of the IEEE/IPSJ 11th International Symposium on Applications and the Internet(SAINT2011), pp.1-10, 2011.
[6] 秋山満昭, 八木 毅, 針生剛男. “改ざん Web サイトリダイレクトに基づく悪性 Web サイト 生存期間測定”. 情報処理学会研究報告, Vol. 2014-SPT-8, No.32, pp. 1-6, 2014.
[7] M. Cherukuri, S. Mukkamala and D. Shin. “Similarity Analysis of Shellcodes in Drive-by Download Attack Kits,” Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing(CollaborateCom2012), pp.687-694, 2012.
[8] T. Matsunaka, A. Kubota and T. Kakasama. “An Approach to Detect Drive-by Download by Observing the Web Page Transition Behaviors,” Proceedings of the 9th Asia Joint Conference on Information Security(AsiaJCIS2014), pp.19-25,2014.
[9] IIJ group Security Coordination Team. “IIJ Security Diary: 継続する Web 改ざんと Exploit Kit によるドライブバイダウンロード”. https://sect.iij.ad.jp/d/2013/07/056557.html
[10] “Weka 3 - Data Mining with Open Source Machine Learning Software in Java”. http://www.cs.waikato.ac.nz/ml/weka/index.html
[11] J. R. Quinlan. C4.5: Programs for Machine Leaning. Morgan Kaufmann, 1993.
[12] L. Xu, Z. Zhan, S. Xu, and K. Ye. “Cross-layer detection of malicious websites,” Proceedings of the third ACM conference on Data and application security and privacy(CODASPY'13), pp. 141–152, 2013.
[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. “Beyond blacklists: learning to detect malicious web sites from suspicious urls,” Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining(KDD'09), pp. 1245–1254, 2009.
[14] “Malware-Traffic-Analysis.net”. http://www.malware-traffic-analysis.net/
[15] “DMOZ - the Open Directory Project”. https://www.dmoz.org/

ⓒ2016 Information Processing Society of Japan