2W-06

Drive-by Download 攻撃における HTTP 通信の特徴を用いた検知手法の提案

榎本　尚代†　　田中　英彦†

情報セキュリティ大学院大学†

1．はじめに

近年の Drive-by Download 攻撃（以下 DbD 攻撃）は，その手法が高度化してきており，難読化やクローキングといった攻撃検知を回避する技術が用いられている．また DbD 攻撃で使われる Exploit Kit と呼ばれる攻撃ツールは，攻撃コードの追加が容易にできるため，修正プログラム公開前の脆弱性がいち早く追加されて攻撃が行われる傾向にある．

2015 年には Adobe Flash Player の脆弱性が多数公表されたが，16 件の脆弱性が DbD 攻撃に悪用されたことが確認されており，そのうちの 9 件についてはゼロデイ脆弱性として悪用された[1]．これらのことから，従来の検知手法の主流であったパターンマッチングによるウイルス対策製品やブラックリスト方式でのアクセス制御では，攻撃の検知が困難になってきている．よって新たな検知手法が必要であると考える．

DbD 攻撃に関する既存研究では，HTTP ヘッダ内に記述される情報を利用して検知する手法が提案されている．

本稿では，HTTP ヘッダフィールド名の出現回数を特徴とし，機械学習を用いて悪性/良性通信を判別する手法を提案する．

2．DbD 攻撃について

2.1 DbD 攻撃の概要

DbD 攻撃とは，攻撃者によって改ざんされた正規 Web サイトや不正に配信された Web 広告を閲覧することでマルウェアに感染させる攻撃手法のことである．DbD 攻撃の典型的な仕組みについて図 1 に示す．

攻撃者は，サイトにアクセスしたユーザを攻撃用のサイトに遷移させるため，HTML の iframe や JavaScript を使って不正コードを埋め込み，正規 Web サイトを改ざんする（図 1,①）．ユーザが改ざんされた Web サイトにアクセスする（図 1,②）と，複数の中継サイトに接続がリダイレクトされる（図 1,③）．中継サイトではユーザ端末からブラウザやプラグインの環境情報を取得し，Adobe Flash Player，JRE，Internet Explorer などの脆弱性を検出する．攻撃コード配布サイトにて脆弱性に対応した不正なスクリプトやプログラムがダウンロード（図 1,④）され，ユーザ端末のシステム権限が奪われる．その後ダウンローダが送り込まれ，マルウェア配布サイトからマルウェアがダウンロードされる（図 1,⑤）．マルウェアに感染したユーザ端末は攻撃者によって不正に操作され，情報漏洩などが発生する（図 1,⑥）．

2.2 関連研究

HTTP 通信の特徴を用いた攻撃検知の研究には，ペイロード情報を利用した検知手法，ヘッダ情報を利用した検知手法などがある．ペイロード情報を利用した検知手法として，ペイロードに記述されるリンクの構造を解析して検知する研究[2]がある．ヘッダ情報を利用した検知手法として，酒井ら[3]は，攻撃時の通信には X-Powered-by ヘッダ内に PHP のバージョン情報が含まれる割合が高いという調査結果を利用した検知手法を提案している．また進藤ら[4]は，攻撃の進行に伴うダウンロードファイルタイプの遷移に着目し，Content-Type ヘッダの情報を特徴として検証を行っている．

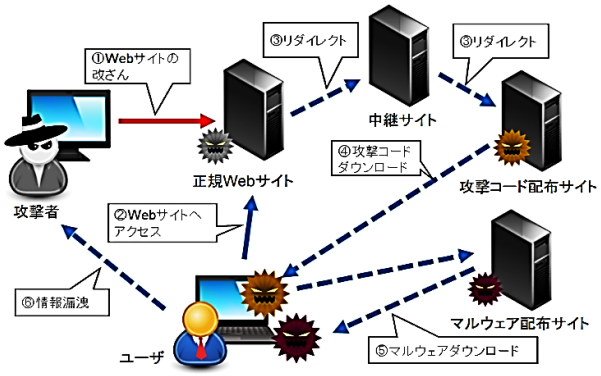

図 1　DbD 攻撃の仕組み

3．提案手法

3.1 着想

DbD 攻撃の一連の通信（図 1,②～⑤）に含まれるヘッダ情報から，一つのヘッダ情報で悪性通信と判定するのは困難である．そのため，関連研究においても複数のヘッダ情報を利用して判定精度を向上させるための様々な検証が行われている．そこで本稿では，一連の通信に含まれるヘッダフィールド名の出現傾向から，悪性/良性通信の分類が可能か検証を行うこととした．

3.2 提案手法

ヘッダフィールド名の出現回数を特徴量として，教師あり学習の一つであるサポートベクターマシン（SVM）を用いて悪性/良性通信を分類する手法を提案する．

4．実験

4.1 実験に用いるデータ

悪性の通信データは，DbD 攻撃の一連の通信データが収録されている D3M(Drive-by Download Data by Marionette)2010～2015[5]から，1306 件のアクセス先 URL の通信データを利用した．また良性の通信データについては，Alexa[6]の Web アクセス数ランキングをもとに Web クローリングを行い，294 件のアクセス先 URL の通信データを利用した．

Python を用いて収集した通信データのパケットをストリーム化して通信内容の再構築を行った．パケットを再構築した後，HTTP メソッド及び HTTP ヘッダ（リクエストヘッダ，レスポンスヘッダ，一般ヘッダ，エンティティヘッダ）を抽出する処理を行った．

A proposal of detection method using characteristics of HTTP traffic for the Drive-by Download attack
Hisayo ENOMOTO† ,Hidehiko TANAKA†
†Institute of Information Security

4.2 分類に用いるヘッダ情報

(1) 全てのヘッダフィールド名

抽出した全てのヘッダについてそれぞれのアクセス先URL毎に，どのヘッダフィールド名が何回出現したかを集計し，特徴とした．

また，下記のヘッダ情報についてはヘッダフィールド名に加えヘッダの値も合わせて抽出し，出現回数を集計して特徴として利用した．図 2 にヘッダの集計リストの例を示す．

(2) Content-Type ヘッダ

送信するコンテンツのデータ形式が記述されるエンティティヘッダである．DbD 攻撃では，PDF ファイル，SWF ファイル，Java ファイル，実行ファイルの 4 種類のファイルダウンロードが危険であるとされているため，このヘッダ内に記述される MIME タイプ毎に集計を行った．

(3) X-Powered-by ヘッダ

Web サーバアプリケーションのバージョン情報が記述されるレスポンスヘッダである．先に述べた酒井らの調査結果から，良悪性の分類に有効であると判断し，このヘッダに記載される値を含めて出現回数を集計した．

(4) Content-Disposition ヘッダ

コンテンツのダウンロード時に使用されるレスポンスヘッダである．このヘッダに記述される Disposition-Type によってダウンロード時の挙動が異なる．悪性通信の場合は強制ダウンロードを表す inline の割合が高いと考えたため，Disposition-Type 毎に集計した．

	ヘッダ ＼ アクセス先URL	URL 1	URL 2	…	URL N
(1)	GET	28	15		68
	Referer	24	9		60
	Content-Length	26	12		64
	Location	2	0		0
	⋮	⋮	⋮		⋮
(2)	Content-Type: text/html	2	7		3
	Content-Type: application/pdf	1	2	…	1
	⋮	⋮	⋮		⋮
(3)	X-Powered-By: ASP.NET	0	5		0
	X-Powered-By: PHP	6	5		6
	⋮	⋮	⋮		⋮
(4)	Content-Disposition: inline	1	1		4
	Content-Disposition: attachment	1	0		0
					特徴量 (出現回数)

図 2　ヘッダ集計リストの例

4.3 実験方法

実験は，4.2 で述べた(1)の特徴を利用した場合，(1)の特徴に加え(2)～(4)の特徴を複数組み合わせた場合，(1)～(4)の特徴を利用した場合の 5 パターンについて行った．表 1 に実施した実験パターンを示す．

表 1　実験パターン

	実験①	実験②	実験③	実験④	実験⑤
(1)ヘッダフィールド名	○	○	○	○	○
(2)Content-Type	－	○	○	－	○
(3)X-Powered-by	－	○	－	○	○
(4)Content-Disposition	－	－	○	○	○

○：利用する　－：利用しない

4.4 実験結果と考察

各実験パターンについて，SVM により 10 分割交差検証を行い，平均値を算出した．実験結果を表 2 に示す．

どの程度正確に悪性/良性を分類できたかの指標となる正解率は，実験③で 97.56%となり最も高い評価となった．実験①のヘッダフィールド名のみの場合でも 96.65%と，十分に高い正解率であった．

表 2　実験結果

	実験①	実験②	実験③	実験④	実験⑤
正解率	96.65%	97.31%	97.56%	96.77%	97.43%
再現率	90.25%	93.45%	93.71%	88.40%	92.87%
F 値	90.39%	92.68%	93.05%	91.38%	93.38%

ただし，通信データを収集したブラウザ環境などの影響により出現するヘッダ情報に偏りが出ている可能性も考えられる．そこで汎化性能を調べるため，他の悪性データについても検証を行う必要があると考えた．最終的には多くの悪性データを収集して検証を行う予定であるが，本稿では初期の検証として，5 件の悪性データを準備し検証を実施した．

5．検証

5.1 検証方法

Malware-Traffic-Analysis.net[7]から，異なる種類の Exploit Kit を悪用した DbD 攻撃の通信データを 5 件収集し，テストデータとした．4 章で利用したデータを訓練データとして学習させ，テストデータが分類できるかを検証した．

5.2 検証結果と考察

検証結果を表 3 に示す．ヘッダフィールド名のみの出現回数を特徴とした検証①の場合では，5 件中 3 件が悪性であると判定できた．検証結果及び 4.4 の実験結果より，Content-Type と Content-Disposition の値が汎化性能の向上に有効であると思われる．

表 3　検証結果

Exploit Kit 名	検証①	検証②	検証③	検証④	検証⑤
Sweet Orange	○	○	○	○	○
Fiesta	○	○	○	○	○
Magnitude	×	○	○	×	○
Nuclear	×	×	×	×	×
Angler	○	○	○	○	○

6．まとめ

本稿では，ヘッダフィールド名の出現傾向から悪性/良性通信の分類をする手法を提案した．D3M を用いた実験では 96%以上の正解率で分類が可能であった．

今後は，どのヘッダ情報が良悪性の分類に影響しているのか調査を行うとともに，より多くの悪性データを収集して学習させ，汎化性能が向上するか検証を行う．

参考文献

[1] IBM：2015 年下半期 Tokyo SOC 情報分析レポート，http://www-935.ibm.com/services/jp/jait-serservisoc-report/(2016-10-18 参照)

[2] 松中隆志 他：Drive-by Download 攻撃対策フレームワーク実現に向けたリンク構造解析による Web サイトの分析，情報処理学会，vol.2015, CSEC, No.48, pp.1-8, 2015

[3] 酒井裕亮 他：Drive by Download 攻撃に対する HTTP ヘッダ情報に基づく検知手法の提案，情報処理学会，vol.2013, CSEC, No.29, pp.1-6, 2013

[4] 進藤康孝 他：マルウェア感染ステップのファイルタイプ遷移に基づいた Drive-by Download 攻撃検知手法，コンピュータセキュリティシンポジウム，Vol.2014, No.2, pp.575-582, 2014.

[5] 神薗雅紀 他：マルウェア対策のための研究用データセット～MWS Datasets 2015～，情報処理学会，Vol.115, CSEC, No.121, pp.37-44, 2015

[6] Alexa - Actionable Analytics for the Web, http://www.alexa.com/ (2016-10-29 参照)

[7] Malware-Traffic-Analysis.net, http://www.malware-traffic-analysis.net/ (2016-12-29 参照)