Drive-by Download 攻撃対策フレームワークにおける Web アクセスログを用いた Web リンク構造の解析による悪性サイト検出手法の提案

松中 隆志 †1　　窪田 歩 †1　　星澤 裕二 †2

†1 KDDI 研究所
埼玉県ふじみ野市大原 2–1–15
{ta-matsunaka, kubota}@kddilabs.jp

†2 セキュアブレイン
東京都千代田区麹町 2–6–7 麹町 RK ビル 4F
yuji_hoshizawa@securebrain.co.jp

あらまし　本稿では著者らが提案する Drive-by Download 攻撃対策フレームワーク (FCDBD: Framework for Countering Drive-By Download) で収集される Web アクセスログを用いた Web ページのリンク構造に着目した悪性サイトの検出手法について提案する．本稿では，FCDBD フレームワークの紹介，ログの収集から解析までの手順の説明，およびリンク構造に着目した提案手法の説明と D3M データセットを用いた当該手法の評価結果について記載する．

An Approach to Detect Drive-by Download by Analyzing Web Link Structureswith Web Access Logs on the Framework for Counting Drive-By Download

Takashi MATSUNAKA†1　　Ayumu KUBOTA†1　　Yuji HOSHIZAWA†2

†1 KDDI R&D Laboratories, Inc.
2–1–15 Ohara, Fujimino-shi, Saitama, JAPAN
{ta-matsunaka, kubota}@kddilabs.jp

†2 SecureBrain Corporation
Kojimachi RK Bldg. 4F, 2–6–7 Kojimachi, Chiyoda-ku, Tokyo, JAPAN
yuji_hoshizawa@securebrain.co.jp

Abstract　We propose an approach to detect and prevent Drive-by Download based on the characteristics of web page transition by using web access logs obtained from the framework for countering Drive-by Download (FCDBD). In this paper, we first explain about the FCDBD framework and how FCDBD framework analyzes web access data and detects Drive-by Download. Then, we explain about our approach utilizing characteristics of web page transition on malicious websites and show the evaluation results of our approach by using D3M dataset.

1　はじめに

Drive-by Download は，Web 上における主要な脅威の一つである．この攻撃は，Web を利用してマルウェアを拡散する攻撃であり，ユーザは，攻撃が仕掛けられた Web ページにアクセスするだけでマルウェアに感染させられてしまう．図 1 に Drive-by Download の典型的な攻撃フローを示す．攻撃者はユーザ環境 (OS，ブラウザ，プラグインなど) の脆弱性を攻撃するサイト (Exploit サイト)，マルウェアを配布するサイト (Distribution サイト)，Exploit サイトまでユーザをリダイレクトさせるサイト (Intermediate

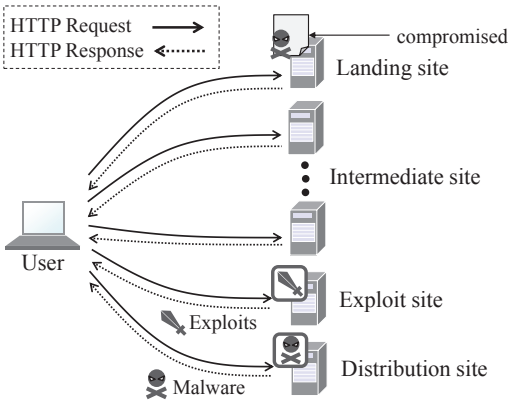

図 1: Drive-by Download の典型的なフロー

サイト) を用意し，さらに Intermediate サイトへユーザをリダイレクトさせるサイト (Landing サイト) を準備する．Landing サイトは多くの場合正規のサイトが改ざんされて，Intermediate サイト，Exploit サイトへユーザをリダイレクトさせるスクリプト (JavaScript，PHP など) を埋め込まれることにより Landing サイトとなる．ユーザが Landing サイトにアクセスすると，Intermediate サイトを経て Exploit サイトまでリダイレクトされる．Exploit サイトでは，ユーザの環境に合わせて攻撃コードを配信する．攻撃が成功すると，ユーザは Distribution サイトからマルウェアを自動的にダウンロードされられ，最終的にマルウェアに感染する．Provos らの報告 [4] によると，Drive-by Download に係る悪性サイト (Exploit サイト，Distribution サイト) は生存期間が短く，発見や解析が非常に困難である．

このような Drive-by Download の被害をくい止めるためには，悪性サイトの出現，消滅など状況の変化に追随して当該サイトの発見，検出を行う必要がある．悪性サイトを発見，検出する手法の一つとして，Web クローラ (honeyclient) を用いた Web サイトの巡回 (クローリング) がある [8]，[9]．honeyclient で効率的に悪性サイトを検知するためには，クローリングの起点となる seed を適切に与える必要がある．また，攻撃者が，自身の悪性サイトの検出を防ぐために，セキュリティ関連企業，研究機関によるクローリングと思われるアクセスに対して正常の Web サイトのようにふるまう (cloaking) ような対策を行うこともあり，クローリングによる悪性サイトの発見，検知は非常に困難である．

悪性サイトを発見，検出する他の手法として，Web ページ間の遷移関係の構造 (リンク構造) に着目して，未知の悪性サイトを検出する方法が提案されている [5], [6], [7]．Stokes ら [5]，Zhang ら [6] の手法は，既知の悪性サイトから当該サイトの参照元をたどり，ハブとなる Intermediate サイトおよび Intermediate サイトからリンクされている未知の悪性サイトを検出するものである．これらの手法では，既知の悪性サイトの情報および Web 上に存在する Web ページのリンク構造に関する情報が必要となる．また，Stringhini ら [7] の手法は，リンク構造上の特徴に加えてユーザのマシンの環境 (OS，ブラウザ，プラグインなど) も加味することで，膨大な Web アクセスログから Drive-by Download に係る悪性サイトを検出するものである．この手法においても機械学習による特徴の抽出のために，膨大な Web アクセスログが必要となる．

以上の状況を鑑み，著者らは Drive-by Download の早期発見，検出および防御を目的としたフレームワーク (FCDBD: Framework for Countering Drive-By Download) を提案，実装した [1]，[2]．このフレームワークでは，ユーザが使用するブラウザおよび Web プロキシに観測センサを設置することで広域な観測網を構築し，ユーザの Web アクセスに関する情報を提供してもらうことで Web 上の Drive-by Download に係る脅威をリアルタイムに把握する．そして検出された脅威の情報を観測センサに適宜フィードバックすることで，ユーザが攻撃の被害にあうのを未然に防ぐ．観測センサを広域に設置することで cloaking による検知逃れを防ぎ，また観測センサより収集される膨大な Web アクセスログなど情報をもとに Web ページのリンク構造の解析などを有効活用した悪性サイトの早期発見が期待できる．

本稿では，まず FCDBD フレームワークによる Web アクセスログにもとづく悪性サイトの検出方法について全体的な流れを説明する．次に，FCDBD にて行う解析事例として，著者ら

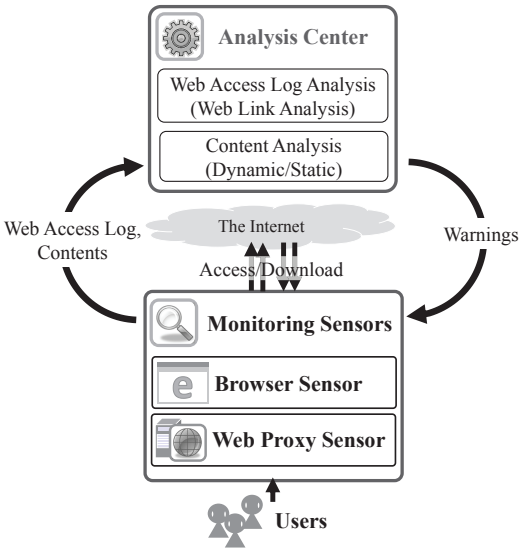

図 2: FCDBD の構成

が考案した Web アクセスログを利用した Web ページのリンク構造の解析にもとづく悪性サイトの検出手法，(1) ページ遷移の挙動にもとづく Distribution サイト検出手法，(2)Web ページの遷移元/遷移先のサイト数に着目した Exploit サイト検出手法について述べる．また上記 2 つの提案手法について，D3M(Drive-by Download Data by Marionette) データセット [3] を用いて評価した結果について述べる．

2 FCDBD: Framework for Countering Drive-By Download

FCDBD (Framework for Countering Drive-By Download) は，Drive-by Download に係る悪性サイトの早期発見，検出および防御を目的としたフレームワークである [1]，[2]．図 2 に FCDBD フレームワークの構成を示す．FCDBD フレームワークはユーザ側に配置される観測センサと，観測センサから提供された情報を解析する解析センタからなる．

観測センサ: 観測センサは，ユーザの Web ブラウザ (ブラウザセンサ) および Web プロキシサーバ (Web プロキシセンサ) に設置され，ユーザの Web アクセスに関する情報を観測し，得られた情報を解析センタに送信する．ブラウザセンサは Web ブラウザのプラグインソフトウェアとして実装され，表 1 に記載した内容を含む Web アクセスログを解析センタに送信する．その際，個々のブラウザセンサは ID で識別されるが，この ID は Web ブラウザが起動されるごとにランダムに変更されるため，解析センタ側で同一ユーザの Web アクセスログを不用意に追跡できない．ブラウザセンサはまた，解析センタ側で悪性が疑われると判断されたサイトのコンテンツをセンタに送信する．

表 1: Web アクセスログの主な内容

・観測センサの ID
・ユーザがアクセスした URL
・ダウンロードしたコンテンツのハッシュ値
・Web ページ遷移時のマウスイベントの有無
・HTTP Request/Response ヘッダ

解析センタ: 解析センタは，観測センサから送信された情報を後述する手順で解析し，悪性と思われるサイトを検出する．そして，観測センサに検出された悪性サイトの情報を送信し，観測センサを利用するユーザが悪性サイトにアクセスするのを防ぐ．

解析センタで行う解析として，Web アクセスログにもとづく各 Web ページのリンク構造の解析，コンテンツ情報にもとづくコンテンツ解析がある．Web ページのリンク構造の解析では，蓄積された Web アクセスログをもとに，例えば，各 Web ページの遷移先ホストの変化を観測し，不明なホストへの遷移が観測されるような変化がみられた場合に改ざんなどによる悪性化を疑う手法 [2]，各 Web ページの遷移先/遷移元サイトの数から Exploit サイトとみられるサイトを抽出する手法 (3 節) を用いる．

また，観測センサからの Web アクセスログの送信に対してリアルタイムに解析を行い，即時的に判定結果を観測センサに返答する処理も行う．これには，Web ページのアクセスから引き起こされる一連のページ遷移の挙動を監視し，Landing ページへのアクセスからマルウェアのダウンロードに至るまでの Web ページの遷移に関する挙動と似た特徴を示した場合に，ダウンロードされた実行形式ファイルを悪性とみな

す手法 (3 節) を用いる．またリンク構造の解析は，コンテンツ解析を行うコンテンツを選定する際にも用いられる．

コンテンツ解析では，動的解析と静的解析 [11] を用いる．実行形式のファイル，PDF には動的解析，JavaScript には静的解析を用いてコンテンツの悪性を判定する．

Web アクセスログの収集から解析までの流れ: 図 3 に解析センタでの処理手順を示す．

(F1) まず，解析センタは，観測センサからログが送信されるログごとに，当該ログに記載された情報に対して悪性サイトへのアクセスを示すかどうかを即時的に判定し，判定結果を当該ログに対する返答として観測センサに送信する．この判定では，過去の解析で悪性と判定された URL およびコンテンツのハッシュ値との照合，Malware Domain List[10] などの外部機関によって公表されているブラックリストとの照合，および 3 節の Web ページ遷移の挙動にもとづく判定手法 (手法 1) にもとづいて判定される．

(F2) 次に解析センタは，当該ログが示すコンテンツを実際に収集するかどうかを判断する．当該コンテンツを収集すると判断した場合，解析センタは観測センサに対して当該コンテンツの送信を要求する．この判断も各ブラックリストとの照合，およびリンク構造の特徴によって，疑わしいコンテンツを選定する．ここで利用するリンク構造の特徴としては，3 節に記載の Web ページの遷移の挙動の他に，Web ページからの遷移先ホストの変化 [2] などを用いることで見逃しを防ぐ．

(F3) 観測センサから収集されたコンテンツは，動的解析/静的解析 [11] により詳細に解析される．その結果，悪性と判定されたコンテンツの情報 (URL，ハッシュ値) はブラックリストとして解析センタ内に格納される．良性と判定されたコンテンツで，当該コンテンツの情報がブラックリストに記載されていた場合は，当該コンテンツの情報をブラックリストから削除することで以降の誤検知を防ぐ．また，蓄積された Web アクセスログをもとに，3 節に記載の Web ページのリンク構造上の特徴にもとづく判定手法 (手法 2) を用いて悪性と思われるサイトを抽出する．抽出された URL は，適宜解析センタ内に格納され，Web クローラによる巡回，コンテンツの収集および解析を行う対象とする．

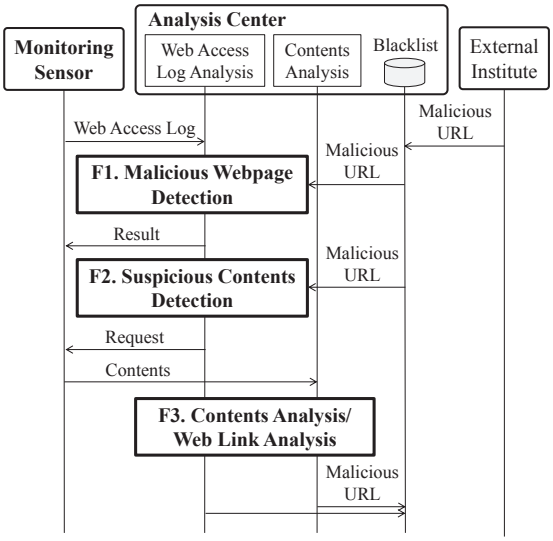

図 3: FCDBD による悪性サイト検出手順

3 Web リンク構造の解析にもとづく悪性サイト検出手法

本節では解析センタで行う解析の事例として，著者らが提案する Web リンク構造の解析にもとづく悪性サイトの検出手法について述べる．本稿で対象とするリンク構造は，ある Web ページにアクセスすることによって自動的に (ユーザによるマウスクリックなどの操作がなく) 引き起こされるページ遷移を表したものを指す．例えば，`iframe`，`img` タグによるコンテンツの参照，`meta` タグなどによる Web ページのリダイレクトなどが該当する．図 4 に実線で例示する．また，以降，一連の自動的に引き起こされるページ遷移のもととなる Web ページを Initial ページと表記する．

手法 1: ページ遷移の挙動にもとづくマルウェアのダウンロード検出手法

図 5 に Drive-by Download におけるマルウェアのダウンロードまでのページ遷移の事例を示す．Landing サイトにアクセスしたユーザは，難読化されたスクリプトによって生成される HTML タグによって，Exploit サイトあるいは

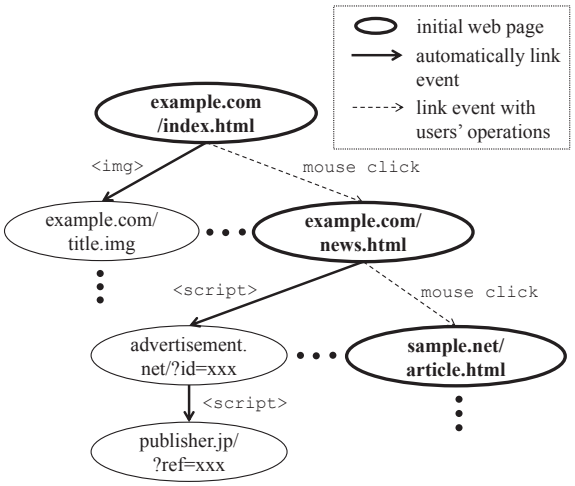

図 4: Web リンク構造の例

Intermediate サイトに遷移させられる．そして Exploit サイトから脆弱性をつくためのコンテンツ (Java アーカイブ，PDF ファイルなど) をダウンロードさせられる．そして攻撃が成功するとマルウェアをダウンロードさせられる．その際，上述のとおり，一連のページ遷移は難読化されたスクリプトおよび脆弱性をつく攻撃によって引き起こされるため，マルウェアのダウンロードに至るページ遷移は，一連のページ遷移の過程においてダウンロードされた HTML ファイル，JavaScript などからは容易に推測できない (特徴 1)．また，マルウェアのダウンロードは攻撃によって引き起こされるため，当該ダウンロードが引き起こされるもととなった参照元のページの URL は `Referer`，`Location` ヘッダなど HTTP ヘッダ上に記載されない (特徴 2)．以上の特徴をもとに，本手法では観測センサから送信される情報をもとに以下の 2 つの条件をいずれも満たすページ遷移によって実行ファイルがダウンロードされた場合に，マルウェアなど悪性コンテンツのダウンロードと判定する．

条件 1: Initial ページから引き起こされる一連のページ遷移に係るすべての HTTP リクエスト/レスポンスヘッダに，対象となる実行ファイルの参照元となる情報 (e.g. `Referer` ヘッダ，`Location` ヘッダ) が存在しない．

条件 2: Initial ページから引き起こされる一連のページ遷移によってダウンロードされるすべての HTML，JavaScript 内に，対象となる実行ファイルへの遷移を示す情報が存在しない．

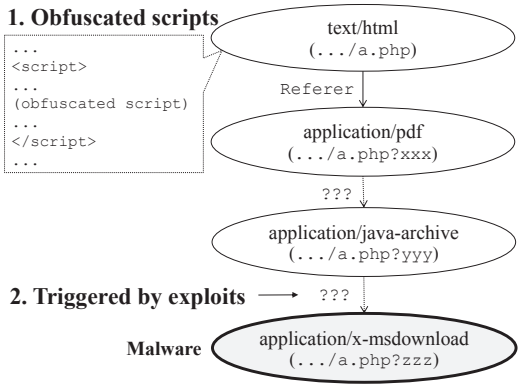

図 5: Drive-by Download のページ遷移例 (1)

手法 2: Web ページの参照元/参照先に着目した悪性サイト検出手法

図 6 に Drive-by Download に係る悪性サイトのページ遷移の他の事例を示す．攻撃者はマルウェアを広域に拡散させるために，複数の Web ページを改ざんし，自身の Exploit サイトへ転送させるスクリプトを埋め込む．そのため，図 6 のように，Exploit サイトのリンク構造を見ると当該サイトは複数のサイトから参照されるような構造になる．そして Exploit サイトではユーザに対して攻撃を仕掛け，最終的にマルウェアをダウンロードさせるため，Exploit サイトからの参照先は単一サイト (Exploit サイト/Distribution サイト) のみとなる．一方，正規の Web ページにおいては，例えば広告ネットワークは，Web ページにアクセスしたユーザを広告主のコンテンツへ誘導するため，Exploit サイトと同様に複数のサイトから参照される．しかし，広告主のサイトは複数であるため，参照先のサイトも複数であると考えられる．

以上の考察より，本手法では (1) 参照元のサイトが複数あり，かつ (2) 参照先のサイトが 1 つしかないような Web ページは悪性であると判定する．その際，広告のバナー，WebBug のようなトラッキング用の gif ファイルによる誤検知を防ぐために，例えば参照先のサイトのコンテンツのサイズが所定値より小さい場合は良性であると判定する．

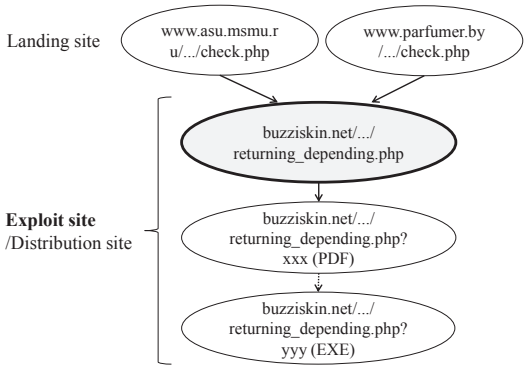

図 6: Drive-by Download のページ遷移例 (2)

4 評価

3 節で述べた各手法について，D3M データセットおよびブラウザセンサで収集した正規 Web サイトのアクセスデータを用いて評価した．

データセットについて

D3M データセット: D3M (Drive-by Download Data by Marionette) データセット [3] は，マルウェア対策研究人材育成ワークショップ (MWS: Anti Malware Engineering Workshop) でマルウェア対策技術の技術者の育成および研究促進を目的として配布されているデータセットである．D3M データセットには，Marionette[9] と呼ばれる honeyclient を用いて収集された Drive-by Download に係る悪性サイトへのアクセス時のトラヒックおよびマルウェアへの感染によって引き起こされたトラヒックのキャプチャデータが収録されている．

本稿では，悪性サイトの Web アクセスデータのサンプルとして D3M データセット 2013, 2014 を用いた．手法 1 の評価においては，実行形式のファイル (`Content-Type` ヘッダの値が `application/x-msdownload`, `application/java-archive`, `application/pdf`) がダウンロードされていた 33URL を抽出し，そのアクセスデータを評価用データとして用いた．手法 2 の評価においては，手法 2 の検出対象である Exploit サイトの最初のページと思われる URL を 28URL 抽出して評価に用いた．

正規 Web サイトのアクセスデータ: 本稿ではまた，正規サイトの Web アクセスデータのサンプルとして，ブラウザセンサを用いて 2,479URL にアクセスした際のログデータを用いた．Web ブラウザは Internet Explorer 9 を用いた．データの収集は 2014 年 2 月 3 日から 3 月 14 日まで実施し，当該期間内でアクセスされた Web ページはの総数は 135,703URL であった．

表 2: 評価結果: 手法 1

	# URLs	Cond.1	Cond.2	Cond.1∧2
Leg.	75	32	0	0
Mal.	33	33	25	25
FP.	—	42.7%	0%	0%
FN.	—	0%	24.2%	24.2%

また，手法 1 の評価のために，上記のアクセスデータとは別に実行ファイルなど (`.exe`, `.dll`, `.pdf`, `.swf`, `.msi`, `.zip`) を自動的にダウンロードさせるサイト 75 件に対してブラウザセンサを用いて Web アクセスログを収集した．

評価結果

手法 1: 表 2 に手法 1 の評価結果を示す．表内の # URLs はサンプルとして用いたサイト (URL) の数，Cond.1，Cond.2 はそれぞれ上述の条件 1，条件 2 を満たすサイトの数，Cond.1∧Cond.2 は条件 1，条件 2 をいずれも満たす，すなわち手法 1 により悪性と判定されたサイトの数を示す．Leg.，Mal. はそれぞれ正規サイト，悪性サイトの評価結果，FP.，FN. はそれぞれ false positives，false negatives の割合を示す．表 2 より，正規サイトにおいて条件 2 を満たすサイトが 0 であるため，false positives は 0%となっている．一方，悪性サイトにおいて条件 2 を満たさないサイトが 8URL 存在していたため，false negatives が 24.2%と高くなっている．この 8URL のうちの 7URL においては，いずれも `applet` タグによって明示的に攻撃用の Java アーカイブ (`Moon.jar`) のダウンロードが示されており，なおかつマルウェアと思われる実行形式のファイルのダウンロードがなされていなかった．その他の 1URL においては，PDF ファイル (`pdf.php`) のダウンロードが明示的に示されているが，以降にマルウェアと思われる実行形式のファイルのダウンロードがなされていなかった．このことから，検知されなかった 8URL はいずれもマルウェアと思われるファイルのダ

ウンロードまで至っていなかったケースであると考えられる．

手法 2: 表 3，表 4 に手法 2 の評価結果を示す．表 3 は D3M データセットより抽出した 28URL に対する参照元/参照先サイトの数を解析した結果を示す．表では，28URL の解析結果を参照元/参照先の数に応じて 4 つのカテゴリに分類している．*#fan-in*，*#fan-out* はそれぞれ Web ページの参照元/参照先のサイト数を示す．参照元/参照先のサイト数は IP アドレスをもとにカウントした．

カテゴリ I に分類された 4URL は手法 2 で検出されるサイトを示す．カテゴリ II に分類された 11URL は，参照元のサイトが少ないため条件を満たしていない．しかし，FCDBD フレームワークによって多くの Web アクセスデータが収集されることで，参照元サイトが増え検出されるようになると期待できる．カテゴリ III に分類された 12URL は参照先サイトが 0 のため，条件を満たしていない．この 12URL はいずれも `applet` タグによって脆弱性をつく Java アーカイブファイルをダウンロードさせていた．この場合は，`Referer` ヘッダがつかないため参照先を解析することができない．しかしながら，参照先に係る情報がないケースにおいては，手法 1 の検出対象となっているため，脆弱性をつく Java アーカイブファイル，もしくは攻撃の後にダウンロードされるマルウェアのダウンロードに係る挙動は，手法 1 によって検知することができると考えられる．カテゴリ IV の 1URL については，参照先が 2 つ以上存在したため条件を満たしていない．このサイトの参照先は同じ FQDN で複数の異なる IP アドレスを利用していた．参照元/参照先の数をカウントする基準については今後の課題である．

表 4 は正規サイトのデータセットを解析した結果を示す．表 4 より，条件に合致する URL(= false positives) は 1.9%ほどであった．条件に合致した Web ページの中には，参照先サイトの参照先から複数サイトに遷移しているケースが見られた．このことから数ホップ先までの参照先を加味することで，誤検知が減少できると考えられる．

表 3: 評価結果: 手法 2 (悪性サイト)

category	*#fan-in*	*#fan-out*	#sites
I	≥ 2	1	4
II	< 2	1	11
III	—	0	12
IV	—	≥ 2	1

表 4: 評価結果: 手法 2 (正規サイト)

Total	$\#\ fan\text{-}in \geq 2 \wedge \#\ fan\text{-}out = 1$
135,703	2,524 (1.9%)

考察

各手法の制限，改善策: 手法 1 においては，HTML 上に明示的にマルウェアなど悪性コンテンツのダウンロードが記載されているような場合には検知できない．この場合 Web ブラウザは，一般的にダイアログを表示して当該ファイルの保存先など処理方法をユーザに問い合わせる．そのため，ユーザは当該ダイアログで保存，実行しない旨を選択することにより，マルウェアへの感染を防ぐことができる．また，Exploit サイトでの攻撃の後，Web ブラウザ経由以外の方法でマルウェアをダウンロードさせるような場合は，本手法では検知することができない．この場合は，例えばセキュリティ対策ソフトウェアのファイアウォール機能で未知のアプリケーション (プロセス) からの通信を遮断するような設定を行うことでダウンロードを防ぐことができる．さらに，Exploit サイトでの攻撃により，攻撃後のマルウェアのダウンロードが観測センサで観測できないケースが考えられるが，今のところそのようなケースは確認されていない．

手法 2 においては，false positives を減らすための対策として，3 節，4 節で述べたように，参照先の Web ページのコンテンツの種類，サイズによるカウントする参照先の選定，および数ホップ先の参照先を加味した参照先のカウントなどを行うなどの改善策が考えられる．

FCDBD での各手法の適用性: 手法 1 は，今回の評価では false positives が 0%であり，また実際にマルウェアのダウンロードに至ったケースにおいては確実に検出できている．そのため，2 節の解析手順 (F1) で手法 1 を用いることでユーザによるマルウェアのダウンロードを即時的に

防ぐ効果が期待できる．

手法 2 は，2 節の解析手順 (F3) において，蓄積された Web アクセスログを定期的に解析する際に適用する．false positives が 1.9%であったことから，手法 2 の条件に合致するような URL に対してさらに Web クローラによるクローリング，コンテンツの収集，解析を行い悪性を確認するような運用が望ましいと考えられる．

5 まとめ

本稿では，著者らが提案する Drive-by Download の早期発見と防御を目的としたフレームワーク FCDBD の紹介と，本フレームワークで行う解析事例として観測センサより提供された情報をもとに Web ページのリンク構造を解析し，悪性と思われるサイトを抽出する手法の提案と評価を行った．

今後，実際にユーザに観測センサを配布してフレームワークを運用し，当該フレームワークの評価を行う予定である．

謝辞

本研究成果は，独立行政法人情報通信研究機構 ((以下，NICT) 理事長: 坂内正夫，本部: 東京都小金井市) の委託研究「ドライブ・バイ・ダウンロード攻撃対策フレームワークの研究開発」により得られたものである．ここに深謝する．

参考文献

[1] 笠間貴弘, 井上大介, 衛藤将史, 中里純二, 中尾康二,「ドライブ・バイ・ダウンロード攻撃対策フレームワークの提案」, コンピュータセキュリティシンポジウム 2011(CSS2011), 2011.

[2] T. Matsunaka, J. Urakawa and A. Kubota, *Detecting and Preventing Drive-by Download Attack via Participative Monitoring of the Web*, Proc. of 8th Asia Joint Conference on Information Security (AsiaJCIS2013), 2013.

[3] 秋山満昭, 神薗雅紀, 松木隆宏, 畑田充弘,「マルウェア対策のための研究用データセット〜MWS Datasets 2014〜」, 情報処理学会 研究報告 コンピュータセキュリティ(CSEC) Vol. 2014-CSEC-66, No. 19, pp. 1–7, 2014.

[4] N. Provos, P. Mavrommatis, M. A. Rajab and F. Monrose, *All Your iFRAMEs Point to Us*, Proc. the 17th USENIX Security Symposium, pp. 1–15, 2008.

[5] J. W. Stokes, R. Andersen, C. Seifert and K. Chellapilla, *WebCop: Locating Neighborhoods of Malware on the Web*, Proc. 3rd USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET2010), 2010.

[6] J. Zhang, C. Seifert, J. W. Stokes and W. Lee, *ARROW: GenerAting SignatuRes to Detect DRive-By DOWnloads*, Proc. 20th International World Wide Web Conference (WWW2011), 2011.

[7] G. Stringhini, C. Kruegel and G. Vigna, *Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages*, Proc. 20th ACM Conference on Computer and Communications Security (CCS2013), 2013.

[8] Y–M. Wang, D. Beck, X. Jiang, C. Verbowski, S. Chen and S. King, *Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities*, Proc. 13th Annual Network & Distributed System Security Symposium (NDSS2006), 2006.

[9] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki and M. Itoh, *Design and Implementation of High Interaction Client Honeypot for Drive-by-Download Attack*, IEEE Trans. of Communication, Vol. E93–B, No. 5, pp. 1131–1139, May. 2010.

[10] *MalwareDomainList*, `http://www.malwaredomainlist.com/`.

[11] 西田雅太, 星澤裕二, 笠間貴弘, 衛藤将史, 井上大介, 中尾康二,「文字出現頻度をパラメータとした機械学習による悪質な難読化 JavaScript の検出」, 情報処理学会 研究報告 コンピュータセキュリティ(CSEC), Vol. 2014-CSEC-64, No. 21, 2014.