Computer Security Symposium 2013
21 - 23 October 2013

Drive-by-Download 攻撃における通信の定性的特徴とその遷移を捉えた検知方式

北野 美紗 †　　大谷 尚通 †　　宮本 久仁男 †

† 株式会社 NTT データ
{kitanoms,ohtanihs,miyamotokn}@nttdata.co.jp

あらまし 近年, 改ざんされた Web サイトにアクセスしたユーザ端末を, マルウェアに感染させる Drive by Download 攻撃が多発している. これらの攻撃においては,Exploit Kit と呼ばれるツールが多用されており, アクセス時の URL に含まれる特徴的な文字列を利用して, 攻撃に起因するマルウェアの感染を検知することができる. しかし, 文字列の特徴が変化した場合は, これを検知できないため, 本研究では上述した文字列の特徴に依存しない, 感染時に発生する一連の通信に着目した. この感染時の通信の定性的な特徴と感染の進行に伴って特徴が段階的に遷移していくことを利用した検知方式を考案, 実装の上,D3M Dataset を利用した評価を行った.

Detection of Drive-by-Download Attack Using Qualitative Characteristics and Transitions

Misa Kitano†　　Hisamichi Ohtani†　　Kunio Miyamoto†

†NTTDATA Corporation

Abstract In the latest cyber attacks, a number of computers are infected by accessing compromised web sites. It is possible to detect these infection by using blacklists and regular expressions of malicious URLs. But since these patterns are variable, it is not effective to use them for detection. In this paper, we propose the method of detecting these infections using qualitative characteristics and its transitions from communication logs by considering mechanisms of Drive-by-Download attacks and Exploit Kits. We implement and evaluate the methods by using D3M Dataset.

1　はじめに

近年, 業務の妨害, 情報の詐取などを目的として, ユーザ端末をマルウェアに感染させるサイバー攻撃が多発している. メールを利用してマルウェアを被害者端末に送り込む手法に加えて, 最近では, 改ざんした Web サイトへのアクセスを通じて, マルウェアを強制的にダウンロードさせるタイプの攻撃が多くなってきている [1]. このタイプの攻撃では, 一般的に Drive by Download 攻撃（以下,DBD 攻撃）と呼ばれる手法が利用される.

この手法の近年の特徴として, Exploit Kit と呼ばれる攻撃用ツールが多く利用されていることが挙げられる [1]. Exploit Kit では, 攻撃に必要なコードとサーバをまとめたものがパッケージとして提供されており, セキュリティ技術に関する知識・経験の少ない者でも容易に DBD 攻撃を行うことができる. この Exploit Kit の存在が, 攻撃件数の大幅な増加の一因となって

いる.

また, 巧妙な仕掛けによって, 被害者は Web 閲覧を行っている最中に, 気づかないままマルウェアをダウンロードして実行してしまう. このようなマルウェアをウイルス対策ソフトでは検知できないことも多い.

本研究では,proxy サーバ,DNS サーバなどサーバ機器に蓄積されるログ (以降,「通信ログ」と言う) に現れる定性的な特徴と, 感染の進行に伴いそれらの特徴が段階的に遷移することを利用して, Exploit Kit を利用した DBD 攻撃によってマルウェアに感染した端末を検知することを目指す.

2 本研究の提案手法

本章では, 通信ログに現れる特徴を利用して, DBD 攻撃を検知する既存の手法とその問題点について述べた後, 本研究における提案手法について述べる.

2.1 既存の検知手法とその問題点

本節では, 通信ログを利用した DBD 攻撃の検知手法のうち既存のものを, 以下に挙げる 2 種類に分類した.

1. URL BlackList を利用する手法

   予め与えられた攻撃者サーバの URL BlackList をもとに, HoneyPot を利用して巡回を行い不審なダウンロードを検知することでさらなる攻撃者サーバの検出を行う手法が, いくつか提案されている ([2], [3]).

   しかし, 攻撃者サーバの URL は頻繁に変化・新出するため, 既存の BlackList を利用する手法では, 追従しきれない. また HoneyPot からのアクセスを検知した場合には, そのときのみ正常なコンテンツを提示するなどの細工が施された攻撃者サーバも数多く存在する [4]. このような場合, 正確な検知を行うことができない.

2. URL 上の特徴を利用する手法

   攻撃者サーバに利用される URL に現れる文字列の特徴を利用して検知を行う手法も提案されている. Zhang らは, クライアント型 HoneyPot を利用して収集した不正 URL 群を利用して, 正規表現によるシグネチャを生成する方式を提案している [5]. また,Exploit Kit を利用時には, URL やファイル名に現れる, 特徴的な文字列が現れることが多いことが多く指摘されている ([6],[7]). これらの文字列を正規表現によって検知することが可能である.

   また, 既存ツールのバージョンアップや新しいツールの登場により, 対象とする文字列の特徴は変化していく. さらに, 攻撃者が Exploit Kit の設定ファイルを変更するなどして,URL に文字列上の特徴が現れないように操作することも可能である.

   このような事情から, 対象とする文字列の特徴パターン, 全てに追従することは難しい.

2.2 本研究で提案する手法

冒頭でも述べたように, 本研究では,Exploit Kit による DBD 攻撃時に通信ログに現れる特徴を利用して, マルウェアに感染したユーザ端末を特定することを目指す. 本研究の特徴は以下 2 点である.

第一の特徴は,DBD 攻撃を受けたときに現れる定性的な特徴を利用している点である. 本研究では,DBD 攻撃の共通的なメカニズムや, Exploit Kit が狙う脆弱性などに依存した, ある程度固定的な特徴を「定性的な特徴」と定義する. 定性的な特徴は, 前述した URL の文字列上の特徴などとは異なり, Exploit Kit のバージョンアップや, 攻撃者の設定によって変化する可能性が低いと考えられるものである. 具体的に本研究で用いたものとしては, UserAgent[1], ファイルの suffix,Web アプリケーションに渡す引数の有無, 受信バイト数などがある. このような定性的な特徴を利用することで, 多種類の DBD 攻撃を検知可能な汎用性の高い検知方式を目指す.

第二の特徴は, 上述した定性的な特徴の「遷移」を利用して検知を試みている点である. DBD

[1]Web アクセス時に利用されるプログラムによって特定の値が現れる

攻撃によって発生する通信は, 正常な Web アクセスによって発生する通信と類似しており検知することが困難である. 上述した定性的な特徴を利用して検知を試みる場合, 正常な Web アクセスが誤検知として大量に発生する恐れがある. 一方で, 誤検知数を減らすために, 定性的な特徴をより詳細に記述するとその分, 第一の特徴の利点である, 検知方式の汎用性が低下する. そこで, 本研究で提案する方式では,DBD 攻撃における定性的な特徴を利用すると同時にその「遷移」追うことで, 汎用性が高くかつ精度の良い検知方式を提案する.

3 DBD 攻撃の感染ステップ

DBD 攻撃は,Web アクセスを通じてユーザ端末のシステム権限を奪い, マルウェアをダウンロードさせる. 異なった攻撃においても, 攻撃者が利用する手法に表面的な差異はあるもの, その手順は共通している. 被害端末の感染は段階的に進行することが一般的に知られている ([4],[8], [9],[10],[11]).

本章では複数の感染事例を分析し, 感染端末のふるまいに着目し Exploit Kit による DBD 攻撃の感染の進行を, 本研究で目的とする検知方式に合わせて, 図 1 に示す 5 つのステップに分解した.

図 1: Exploit Kit による DBD 攻撃に共通するステップ

1. redirect ステップ
   正常 Web サイトを管理するサーバのパスワードを詐取するなどの方法によって, 攻撃者は正常 Web サイトを乗っ取る. そしてコンテンツを改ざんし, iframe や JavaScript などの不正コードを埋め込む. このような不正コードによって, 正常 web サイトにアクセスしたブラウザは一旦別のサーバを経由してから, 攻撃者サーバへリダイレクトされる. このステップを,「redirect ステップ」と定義する.

2. pre-exploit ステップ
   被害端末が攻撃者サーバにリダイレクトされてくると, 攻撃者サーバは被害端末の脆弱性を調査し,exploit を実行する準備を行う. 本研究ではこのステップを,「pre-exploit ステップ」と定義する. pre-exploit ステップにおいては, 被害端末の OS, ブラウザ, ブラウザプラグイン, Java Runtime Environment(以下,JRE) などのバージョンが検出するためのコードがダウンロードされる. コード中では,JavaScript の PluginDetect ライブラリが利用されることが多い [12]. 感染端末から送られてきた HTTP リクエストに含まれる, UserAgent ヘッダを利用して, 攻撃者サーバ側で OS, ブラウザ,JRE のバージョンの検出を行うものもある [11]. このような機能を利用して, 攻撃を実行するのに最適な脆弱性を選択した後に, 脆弱性ごとに用意された exploit 用の Web ページへアクセスさせる.

3. exploit ステップ
   pre-exploit によって, 被害端末が exploit 用の web ページへアクセスすると, 脆弱性を利用して, 被害端末の権限を奪うことが行われる. このステップを「exploit ステップ」と定義する. 利用される脆弱性としては, 主に JRE,Adobe Reader/Acrobat,Adobe Flash Player, Internet Explorer(以下 IE) などがある. 利用する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・実行され, buffer overflow などが引き起こされ, 端末のシステム権限が奪われる.

4. pre-download ステップ
   exploit ステップで被害端末の権限を奪った後, 攻撃者サーバは, 被害端末にインターネット上から特定のファイルをダウンロードするような指示が書かれたコードを送り込む. このコードは一般的にダウンローダと呼ばれ, 次ステップの malware download が引き起こされる. これを pre-download ステップと定義する.

5. malware download ステップ

downloader により, マルウェア本体がダウンロードがされるステップを malware download と定義する. Zeus,Citadel など感染端末の情報を窃取するような危険性の高いマルウェアがダウンロードされることが多く, このステップまで攻撃が進行した場合, 非常にセキュリティ上のリスクが高いといえる.

4 通信ログに現れる特徴の遷移

本章では,3 章で述べた 5 段階のステップについて, 通信ログに現れる定性的な特徴の遷移 (以下, 定性的な遷移とする) を説明する. 4.1 節では,DBD 攻撃に共通的に現れる定性的な遷移について, 4.2 節では,Exploit Kit ごとに固有の定性的な遷移について述べる.

4.1 DBD 攻撃に共通する定性的な遷移

本節では,3 章で述べた各ステップにおいて, DBD 攻撃に共通的に現れる特徴の遷移を抽出した. 全体を通して, 主にドメイン名,HTTP ヘッダの UserAgent などに現れる特徴が遷移すること.

1. redirect ステップ
   通信ログにはステータスコードがリダイレクトを示す 300 番台が記録される. redirect ステップで利用されるサーバは中継専用に利用され, 以降はほとんどの場合, 登場しない.

2. pre-exploit ステップ
   攻撃者サーバに誘導されるので, ドメインが遷移する. 殆どの場合, 以降,pre-download ステップまで, 複数ドメインをまたがらず単一ドメイン上で攻撃が発生する.

3. exploit ステップ
   exploit ステップで現れる特徴は, 利用される脆弱性によって異なる. 表 1 で 2013 年 4 月時点で, 各 Exploit Kit が悪用している脆弱性の内訳を示す ([13],[16]). 大半が JRE の脆弱性である. 以下で JRE の脆弱性を狙った Exploit (以下 Java Exploit) の定性的特徴について述べる.

表 1: 各 Exploit Kit が利用する脆弱性の内訳と件数

Kit	JRE	AR[2]	IE	その他
BlackHole	6	1	1	0
RedKit	6	1	0	0
Neutrino	2	0	0	0
Sakura	6	0	2	0
SweetOrange	7	1	1	0
Glazunov[3]	-	0	0	-
Cool	3	2	0	3
Styx	6	1	0	1

   JRE Exploit の検知には, 通信ログの UserAgent を利用する. 一般に,JRE Exploit はブラウザによってダウンロードされた Java プログラムが, 自動的に実行されることで発生する. その後 Java アプリケーションが攻撃者サーバに対して通信を行い, 通信ログ中にも UserAgent が Java として記録される.

   Java 以外の脆弱性を利用する Exploit では, 不正プログラムがブラウザ上で実行されることが多く, UserAgent の値もブラウザから変化せず, この方法では検知することが難しい. このため, 主に次節で解説する各 Exploit Kit の特徴を利用して検知を行う.

4. pre-download ステップ

   pre-download ステップにおいては, 特に受信バイト数に特徴が現れる. ダウンローダのダウンロードが発生するため, redirect〜exploit ステップにおける受信バイト数は, 通常数 100〜数 1,000B 程度であるのに対し, pre-download ステップでは, 数 10KB〜100KB 程度と受信バイト数が遷移することが多い.

5. malware download ステップ

   malware download ステップにおいては, 多

[2]Adobe Reader/Acrobat を利用する脆弱性
[3]脆弱性の内訳についてのデータは得られなかったが, 主に JRE の脆弱性を利用することが知られている.

表 2: pre-exploit ステップにおける各 Exploit Kit の定性的特徴

	ファイルの suffix	引数	URL 例
BlackHole	.php	無	http://[domain]/q.php
RedKit	.html	有	http://[domain]/qawe.html?i=12345
Sakura	.php	無	http://[domain]/check.php
SweetOrange	php	有	http://[domain]/gu.php?vote=71&long=264&(略)
Glazunov	判別不可	無	http://[domain]/2634432140/589
Cool	.php,.html(複数回)	無	http:///[domain]/a.php, /b.html, /c.html, ...
Styx	判別不可	無	http://[domain]/6YciA04F(略)5Ql06rpn/

くの場合ドメインの遷移が発生する. これはマルウェアが格納されているサーバは, ほとんどの場合, Exploit Kit が格納されているものとは別のサーバであるためである.

また, マルウェアのダウンロードはダウンローダよって行われ,UserAgent はブラウザのものとは異なるダウンローダ独自の値が現れる.

具体的には UserAgent 中の,OS バージョン,IE バージョン項目がインストールされているものと異なる場合, マルウェアのダウンロードが発生している可能性があるといえる.

4.2 Exploit Kit 固有の定性的な遷移

4.1 節で述べた,DBD 攻撃に共通的に現れる定性的な遷移に加えて, Exploit Kit それぞれに固有の定性的な遷移もある.

本節では既存文献の情報 ([6],[7],[14], [15]) および, 独自に収集した proxy ログ,pcap データをもとに複数の感染事例について調査を行い, 多くの攻撃で利用されている代表的な 8 種類の Exploit Kit(BlackHole,RedKit,Neutrino,Sakura, SweetOrange,Glazunov, Cool,Styx) を分析した.

2.1 節で述べたように,Exploit Kit ごとに固有の URL の文字列上の特徴は頻繁に変化する.

本研究では,Expoit Kit が主体となって引き起こす pre-exploit〜pre-download ステップにおいて, ツールのバージョンアップや攻撃者の操作でも変化する可能性が少ない定性的な特徴として, ファイルの suffix, web アプリケーションに

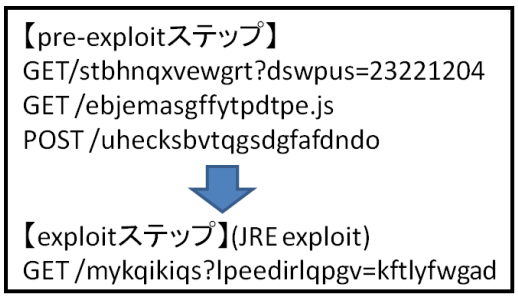

図 2: Neurtino Exploit Kit による pre-exploit〜exploit ステップのアクセス例

渡す引数の有無を選定しその特徴があることがわかった.

まず, 一例として, 図 2 で Neutrino によって pre-exploit ステップから exploit ステップにおいて現れる, 特徴の遷移を示す. Neutrino は Exploit Kit の中でも特徴的な通信が発生する. 図 2 に示すように, ファイルの suffix が URL 中に現れないが, exploit が発生すると引数を渡す.POST メソッドによる通信も発生する.

他の Exploit Kit についても,pre-exploit〜pre-download の各ステップにおいて特徴の遷移が発生する. 今回調査した範囲のうち,pre-exploit ステップにおいて, 現れる特徴を表 2 に一覧化した.

表 2 に示すように,Exploit Kit によって, 特徴は異なる. exploit,pre-download ステップにおいても, 同様に特徴が現れ, 攻撃の進行に伴い遷移していく.

5　検知方式

本章では,4 章で解説した特徴を通信ログから検出するための検知方式と, 実際に作成した検知ルールについて解説する.

5.1　前提条件

本検知方式では,proxy サーバログに一般的に含まれている項目である, 日時, 発信元 IP, アクセス先 URL,UserAgent, 受信バイト数, メソッドを利用して検知を試みる.

5.2　検知に利用する遷移のパターン

本節では検知手法に利用する遷移のパターンについて述べる. 4 章で述べた 5 ステップ中における, ステップの遷移としては, 網羅的には 10 種類が存在するがその中でどの遷移を利用すれば, 効果的な検知ができるのかは明らかではない. 今回は, 特徴の捉えやすさを考慮し, 図 3 に示す,3 種類の遷移を利用して検知ルールを作成した.

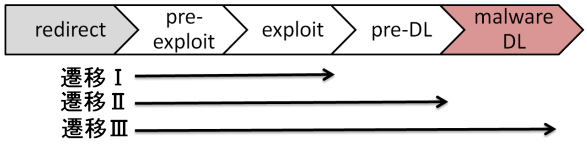

図 3: 検知に利用する遷移のパターン

5.3　作成した検知ルール

5.2 節で述べた検知方式に基づき,3 パターンの遷移を通信ログから抽出する検知ルールを記述した. Exploit Kit の種類や exploit の種類 (JRE Exploit か否か) によって, 通信ログに現れる特徴の遷移が大きく異なることに留意し, 前述した 3 種類の遷移を抽出するルールを作成した. 合計で 32 件の検知ルールを作成した.

6　検証

本章では,5 章で作成した検知ルールを検知数, 誤検知の内容の観点から検証する.

6.1　D3M Dataset を利用した検証

検知数の検証には,D3M Dataset 2012[18],2013[17] を利用する. D3M Dataset2012, 2013 には,Web クライアントハニーポット Marionette を利用して,2012 年 3 月〜2013 年 3 月の間に, 悪性 URL を巡回した際に得られた DBD 攻撃による感染の通信データが pcap 形式で提供されている.

6.1.1　検証用ログの作成

検知数の検証を行うために,D3M Dataset のデータを,5 章で作成した検知ルールが適用可能な, 通信ログ形式に整形した. 検証用ログ作成の手順は以下の通りである.

1. 必要な項目の抽出
   D3M Dataset に含まれる HTTP パケットデータの中から, リクエスト・レスポンスパケットを抽出し, 今回の検知ルールに利用している項目, 日時, アクセス先 URL, メソッド, UserAgent,status コード, 受信バイト数を抽出し, 通信ログ形式に整形した.
2. 感染ログ群の集約
   D3M Dataset には複数の感染事例が含まれる. 1. で抽出した通信ログの発生時間間隔, アクセス先ドメインを利用して, 1 回の感染事象において発生する一連のログを, 感染ログ群として集約した. 結果,1. で抽出した 2,358 件の通信ログから, 計 203 件の感染ログ群を生成することができた. 1 件の感染ログ群の中には, 感染時に発生する一連の遷移が含まれるため, 検証においては, この 203 件の感染ログ群中から, 何件を検知できたかを検証する.

6.1.2　D3M を利用した検知数の検証

検知ルールを, 以上の手順を経て作成した検証用ログに適用した検知数の結果を表 3 に示す. 表には検証用ログを検知した検知ルールごとに, 対象とする exploit が利用する Exploit kit, 脆弱性を示し, 検知数を提示している.

また, 表中の各ルールが検知した感染ログ群には, 重複がある. 重複を除外すると, 全体としては 203 件の感染ログ群のうち 127 件を検知することができた. 検知率としては, 約 62.5%となる.

表 3: D3M Dataset を検知したルールの検知数

ルール名	脆弱性	検知数
BlackHole1	JRE	13
BlackHole2	JRE 以外	4
SweetOrange1	JRE	67
SweetOrange2	JRE 以外	26
Sakura1	JRE	9
Sakura2	JRE 以外	23
総検知数 [4]	127/203 件	

6.1.3 検知した感染ログ群の内訳

表に示すように検知した感染ログ群は BlackHole,SweetOrange,Sakura によるものであり, 各 Exploit Kit において JRE の脆弱性を利用するものと, 利用しないものが存在した.

特に BlackHole では URL 上に文字列上の特徴が現れるものとそうでないものが存在する. 今回考案した検知ルールでは, どちらの感染ログ群も検知することができた.

SweetOrange の検知ルールで検知した感染ログ群は, 実際には BlackHole の旧バージョンによる感染ログであった. また,Sakura の検知ルールで検知した感染ログ群を調査すると, 実際には Phoenix Exploit Kit によるものであることがわかった.

以上により,Exploit Kit の定性的な特徴を利用することで, 文字列上の特徴が現れない感染ログ, 異なった Exploit Kit による感染ログも検知できると言える.

また, 検知したルールでは, 遷移 I まで進行した感染ログ群が大半であり, 遷移 II, 遷移 III まで進行している感染ログ群は, そもそも D3M Dataset に含まれておらず, 検知しなかった. これは,D3M Dataset を作成する際に利用する Marionette は, 不正コンテンツを受信した時点で無害化するため, pre-download ステップ以降の感染データを収集しないためと考えられる.

6.1.4 検知しなかった感染ログ群の内訳

検知しなかった 75 件のうち,26 件については, ふるまいが不明であったため, 本研究の対象外とする. 19 件は Exploit Kit を利用せず, 直接マルウェア本体をダウンロードさせているとみられるような単純な手法による感染ログ群であった. これらのログ群を除外すると, 検知率は約 80.8%(127/157 件) となる. 単純な手法による感染は本研究の対象外であるが, 感染後に発生する通信の特徴により検知することができる [19].

また,30 件は,Bleeding Life,Elenore などといった, 検知ルールを作成する際に未想定の Exploit Kit による感染ログ群であった. これらは, 定性的な遷移が今回対象とした Exploit Kit と異なるために検知することができない. また, そのほとんどが本研究の調査時 (2013 年 8 月) の 1 年以上前である 2012 年 8 月以前に収集されたものである. Exploit Kit は, 常に新しいものが登場しており, 既存の定性的特徴のみでは検知できない場合があり, 新しく登場するものついては, キャッチアップしていく必要がある.

6.2 誤検知原因

誤検知の検証として, 作成した検知ルールを, 企業内ネットワークの約 1200 万行の通信ログに対して, 適用し原因の調査を行った.

ほとんどの検知ルールでは, 誤検知は 5 件以下であったが, SweetOrange,Sakura,Styx に対応する検知ルールのうち, JRE Exploit 以外の感染時に遷移 I を捉えるものでは, 約 10 件のドメインに対して誤検知が発生した. 例えば SweetOrange による感染ログでは,php スクリプトへアクセスするが, 掲示板などの web アプリケーションへのアクセスによっても同様のアクセスが発生する. このように, 感染時に発生するログが通常の Web アクセスによって, 発生するログと類似している場合, 誤検知が発生する.

[4] 各ルールごとの検知数から重複を削除した総検知数

7　まとめと今後の課題

本研究では,Exploit Kit を利用した DBD 攻撃が共通のステップを経て進行することに着目した. 8 種類の Exploit Kit について, 通信ログに現れる特徴を調査し, 検知方式を考案し実装した. 実装した検知ルールを検知と誤検知の観点から検証し, いくつかの検知ルールについて有用性があることを確認できた. 本研究の考察を経て得た課題を以下で述べる.

- JRE Exploit 以外の Exploit
  本研究では,JRE Exploit については, 十分な情報を収集し精度の良い検知ルールを作成することができたが, Adobe Reader/Acrobat,Adobe Flash Player, IE などの脆弱性を利用する exploit ステップでのふるまいについて, 十分な情報を収集することができず, いくつかの検知ルールでは誤検知が多い. これらも大きな脅威であるため, 今後調査する.
- 対象とするデータの拡大
  proxy,DNS,Firewall といった通信ログのみならず, IDS のログや HTTP パケットを利用した検知を行いたい. 誤検知の低減や, より多種類の攻撃の検知が期待できる.
- 攻撃の進行度の把握
  攻撃が成功したステップによって, セキュリティリスクは大きく異なり, 対応方法も異なってくる. 攻撃の進行度を定量的に把握する方式について検討を進めたい.

参考文献

[1] ”2012 年下半期 Tokyo SOC Report” 日本 IBM

[2] Stokes, Jack W., et al. ”Webcop: Locating neighborhoods of malware on the web.” USENIX Workshop on Large-Scale Exploits and Emergent Threats. 2010.

[3] Akiyama, Mitsuaki, Takeshi Yagi, and Mitsutaka Itoh. ”Searching structural neighborhood of malicious urls to improve blacklisting.” Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on. IEEE, 2011.

[4] Lu, Long, et al. ”Blade: an attack-agnostic approach for preventing drive-by malware infections.” Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010.

[5] Zhang, Junjie, et al. ”Arrow: Generating signatures to detect drive-by downloads. ” Proceedings of the 20th international conference on World wide web. ACM, 2011.

[6] MalwareSigs
http://www.malwaresigs.com/

[7] Malware don’t need Coffee
http://malware.dontneedcoffee.com/

[8] Egele, Manuel, Engin Kirda, and Christopher Kruegel. ”Mitigating drive-by download attacks: Challenges and open problems. ” iNetSec 2009 Open Research Problems in Network Security. Springer Berlin Heidelberg, 2009. 52-62.

[9] ”Blue Coat Systems 2012 Web Security Report”

[10] Van Lam Le, Ian Welch, Xiaoying Gao, and Peter Komisarczuk. ”Anatomy of Drive-by Download Attack.” (2013).

[11] Vadim Kotov, Fabio Massacci (2013) ”Anatomy of Exploit Kits Preliminary Analysys of Exploit Kits as Software Artefacts” ESSoS 2013, LNCS 7781, pp. 181-196, 2013

[12] Fraser Howard (2012) ”Exploring the Blackhole Exploit Kit” SophosLabs,UK

[13] contagiodump
http://contagiodump.blogspot.jp/

[14] Malware Must Die!
http://malwaremustdie.blogspot.jp/

[15] MALware FORensics SECurity
http://malforsec.blogspot.jp/

[16] Naked Security
http://nakedsecurity.sophos.com/

[17] 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 2013～,MWS2013（2013 年 10 月）

[18] MWS2012 実行委員会, 研究用データセット MWS 2012 Datasets について, http://www.iwsec.org/mws/2012/about.html

[19] 大谷 尚通, 北野 美紗, 重田 真義：企業内ネットワークの通信ログを用いたサイバー攻撃検知システム,CSS2013（2013 年 10 月）