Vol.2014-DPS-158 No.35
Vol.2014-CSEC-64 No.35
2014/3/7

Exploit Kit 検知用シグネチャの動的解析に基づく自動作成

柴原健一[†]　笠間貴弘[†‡]　神薗雅紀[‡*]
吉岡克成[†]　松本勉[†]

Web ブラウザやプラグインの脆弱性を狙って攻撃し，ユーザに気付かれずにマルウェアに感染させる Drive by Download 攻撃が流行している．この流行の背景にはエクスプロイト攻撃やリダイレクトを行う Web サイトを容易に構築可能な BlackHole Exploit kit や Phoenix Exploit kit といったツールの流布があると考えられる．そこで本稿ではサンドボックス内で Exploit kit により悪性 Web サイトを構築し，様々な種類のクライアントからアクセスすることにより得られる通信の内容を観測し，Exploit kit 検知用のシグネチャを自動作成する手法を提案する．

Automatic Generation of Exploit Kit Signature Based on Sandbox Analysis

KENICHI SHIBAHARA[†]　TAKAHIRO KASAMA[†‡]　MASAKI KAMIZONO[‡]
KATSUNARI YOSHIOKA[†]　TSUTOMU MATSUMOTO[†]

Drive by Download attacks which exploit browser and/or browser's plug-in are prevailing. Exploit Kits are heavily used to construct malicious web sites for Drive by Download attacks. In this report, we propose a technique to automatically generate a signature to detect Exploit Kits by actually creating malicious web sites in a sandbox using the Exploit Kits and accessing them with various browsers to monitor their characteristic traffic.

1. はじめに

近年，Web サイトを閲覧したユーザに強制的にマルウェアをダウンロードさせる Drive by Download 攻撃（以下 DBD 攻撃とする）の被害が増えている．正規の Web ページが改ざんされて DBD 攻撃に利用されることが多く，アクセス数の多い大企業の Web ページが改ざんされる事件が発生している．DBD 攻撃は，主にユーザの Web ブラウザやプラグインの脆弱性を狙ってクライアントマシンに侵入しマルウェア等を送り込む攻撃であり，Web ページ閲覧の裏で行われるため，ユーザが被害に気づきにくい点が特徴である．

この DBD 攻撃の流行の背景として Exploit Kit の存在が挙げられる．Exploit Kit は，様々な種類の脆弱性を突く攻撃コードや，リダイレクト，ユーザプロファイルやブラウザの種類に応じて応答を変える機能を有しており，悪性 Web サイトを構築するためのツールキットである．ネットワークや脆弱性に関する知識が浅い者でも簡単に利用することができることから，DBD 攻撃の実施をより容易にしている．Exploit Kit には様々な種類があり，それぞれにおいて攻撃の方法や挙動には特徴があることから，これに基づき検知を行う方法が文献[9]で提案されている．しかし文献[9]では検知を行うための Exploit Kit の特徴を手動で抽出しており，更新が頻繁な Exploit Kit への対応にはコストが掛かる．そこで本稿ではサンドボックス内に検知対象の Exploit Kit により悪性 Web サイトを構築し，これに対して様々な種類のブラウザを用いてアクセスを行い，その通信の中から URL の遷移パターンの特徴を抽出し，Exploit Kit 検知用シグネチャを自動作成する方法を提案する．

第 2 章で背景や関連研究について説明を行い，第 3 章で本稿において提案する手法の説明をする．第 4 章で本手法の評価実験を行い結果についてまとめる．第 5 章で考察を行い，第 6 章でまとめる．

2. 背景・関連研究

2.1 Drive by Download 攻撃

DBD 攻撃の流れを図 1 に示す．

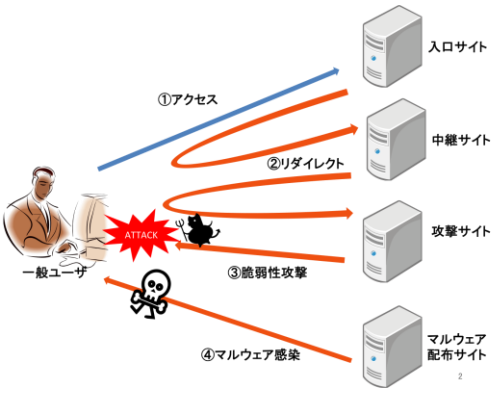

図 1　Drive by Download 攻撃の流れ

† 横浜国立大学
Yokohama National University
‡ 独立行政法人情報通信研究機構
National Institute of Information and Communications Texchnology
* 株式会社セキュアブレイン
SecureBrain Corporation

DBD 攻撃は主に入口サイト，中継サイト，攻撃サイト，マルウェア配布サイトの４つのサイトで構築されている．

入口サイトは DBD 攻撃の始点となるサイトであり，ユーザがアクセスすると中継サイトへのリダイレクトが行われる．入口サイトは，正規のサイトが改ざんされて利用される場合と，攻撃者が用意する場合が考えられる．正規のサイトを改ざんする方法としては，Web サイトの脆弱性を利用したり，Web サイト管理者から FTP のアカウントを盗取したりする方法が挙げられる．正規のサイトを狙う理由としては標的となるユーザを誘導する手間を省くことができることが挙げられる．正規のサイトが改ざんされる被害が多数報告されている[11]．攻撃者が入口サイトを用意する場合は，不正メールや掲示板，SNS サイトなどを通じて当該サイトに被害者をおびき寄せるといった方法が採られる．入口サイトでのリダイレクトは IFRAME タグや Javascript 等で行われ，その後，中継サイトにもリダイレクトされ，最終的には攻撃サイトにアクセスする．中継サイトは複数存在する場合が多く，解析を困難にする目的があると思われる．中継サイトでは TDS（Traffic Distribution System）が使われることもある[2]．終点となる攻撃サイトでは，ユーザの環境（OS や Web ブラウザ，Adobe Flash Player，Java Runtime Environment 等）に合わせて脆弱性を突く攻撃を行う．その結果，攻撃サイトまたはマルウェア配布サイトから自動的にマルウェアがダウンロードされ実行される．

2.2 Exploit Kit

Exploit Kit とは前節で説明した DBD 攻撃の基盤の作成や，脆弱性（Java Runtime Environment，Adobe Flash Player，各種 Web ブラウザ等）を狙った攻撃をするツールキットである．Exploit Kit は主に図１に見られる中継サイト，攻撃サイト，マルウェア配布サイトを構築する．Exploit Kit を使うことで，複数のサーバ間を中継するリダイレクトや，ユーザに合わせた攻撃方法の変更などを容易に行える．一般的に Exploit Kit はアンダーグラウンドのマーケットで非合法に売買されており，様々な種類が存在する．多くの Exploit Kit は GUI を採用しており，収集したユーザの統計情報が表示されていたり，各種設定も変更できたり，視覚的でわかりやすい操作が可能である．

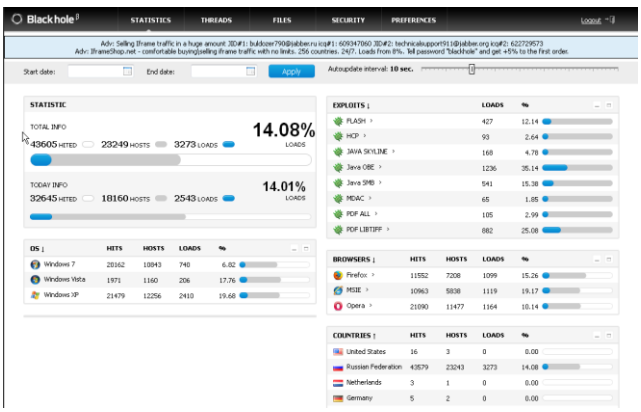

図 2　BlackHole Exploit Kit の管理画面[3]

また，オンラインでのアップデートが可能な Exploit Kit も存在し，新たに見つかった脆弱性への対応が可能である．contagio malware dump[4]というサイトがまとめた Exploit Kit のアップデート状況の一部を表 1 に示す[5].

Gong Da / GonDad	Redkit 2.2	x2o (Redkit Light)	Fiesta (=Neosploit)	Cool Styxy	DotkaChef
CVE-2011-3544	CVE-2013-2551	CVE-2013-2465	CVE-2010-0188	CVE-2010-0188	CVE-2012-5692
CVE-2012-0507	CVE-2013-2471		CVE-2013-0074/3896	CVE-2011-3402	CVE-2013-1493
CVE-2012-1723	CVE-2013-1493		CVE-2013-0431	CVE-2013-0431	CVE-2013-2423
CVE-2012-1889	CVE-2013-2460		CVE-2013-0634	CVE-2013-1493	
CVE-2012-4681			CVE-2013-2551	CVE-2013-2423	
CVE-2012-5076					
CVE-2013-0422					
CVE-2013-0634					
CVE-2013-2465					

表 1　Exploit Kit アップデートにより追加された脆弱性

GongDa，Redkit，Styx など最近流行している Exploit Kit が並んでいる．例えば CVE-2013-2551 は攻撃者が作成した Web サイトを通して任意のコードを実行できる Internet Explorer（ver 6~10）の脆弱性である[6]．Exploit Kit によっては検知を回避する機能が搭載されている．例えば，同じ IP アドレスからの複数回のアクセスの遮断，Referer 情報を基に想定されるリダイレクト元以外からの Exploit Kit へのアクセスを遮断する機能や，匿名通信路である Tor を使ったアクセスを遮断するといった機能が存在する．

2.3 関連研究

DBD 攻撃の検知手法はこれまでにいくつも提案されている．例えば公開 URL ブラックリスト[7]を使用する方法があるが，悪性サイトの存在期間は短い場合も多く，このような悪性 URL の有効期間は短い．そこで論文[8]では，悪性 URL 群の特徴に着目し木構造を作成しフィルタリングを行う方法を提案している．また，論文[9]では時間的に変化する悪性 URL 群をハニーポットで効率的に巡回する方法を提案している．

通信内容による検知では，論文[10]において DBD 攻撃の多くに PHP を用いられていることから通信の HTTP ヘッダの情報を基に検知を行っている．論文[11]では具体的に Exploit Kit を対象とした検知手法が提案されている．既知の各種 Exploit Kit を解析し，URL 及び引数を対象に特徴を抽出して検知ルールを作成している．この検知ルールは論文[12]で提案された大規模な DBD 攻撃対策フレームワーク上で適用することを想定している．

Vol.2014-DPS-158 No.35
Vol.2014-CSEC-64 No.35
2014/3/7

3. Exploit Kit 検知用シグネチャの自動作成

第 2 章 2 節で説明したように Exploit Kit は様々な種類があり，挙動も異なっている．論文[9]ではその特徴から検知ルールを作っており，これらは Web 上で公開されている情報や，Exploit Kit のリソースファイル，ソースを解析して作成されている．これに対して本稿では，Exploit Kit 検知用シグネチャを自動作成する手法を提案する．提案手法では，Exploit Kit の特徴であるリダイレクトによるアクセス先 URL の遷移に着目し，各 URL 内のパスとクエリパラメータを正規表現として表すことでシグネチャとする．提案手法は収集フェーズとシグネチャ作成フェーズで構成される．以下にそれぞれのフェーズを示す．

3.1 収集フェーズ

このフェーズでは Exploit Kit 検知用シグネチャを生成するための通信データを収集する．収集する通信データは Exploit Kit にアクセスする際の HTTP 通信とする．仮想環境上のサーバにシグネチャ生成対象の Exploit Kit を用いて悪性 Web サイトを構築し，同仮想環境内のクライアント側から Web ブラウザを通してアクセスすることで，実際に行われる DBD 攻撃を模擬する．Exploit Kit は一般にクライアント環境に応じて攻撃の挙動を変化させるためクライアント側で使用する Web ブラウザは複数の種類，バージョンを使用して様々な環境での通信を収集する．ここで，Exploit Kit の中には，同一の IP アドレスを用いたクライアントに対してアクセス履歴に応じて応答を変える機能があるものがあり，特に初回通信時のみ攻撃を行うものが存在するため，上記の通信データ収集時は，アクセス毎にクライアントの IP アドレスを変更し，実際の DBD 攻撃時と近い状況で攻撃通信を観測できるようにする．収集した通信はクライアントの IP アドレス毎に分割して DB に保存する．この収集フェーズで行われる実験は外部との接続を断った PC の仮想環境上において行うことで，安全性を保証している．

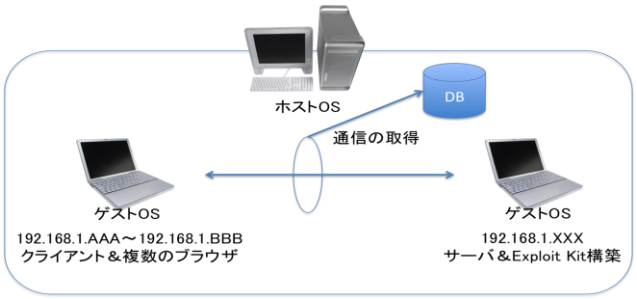

図 3　収集フェーズ概念図

3.2 シグネチャ作成フェーズ

前節の収集フェーズにて収集した通信データから Exploit Kit 検知用シグネチャを作成する．作成するシグネチャは Exploit Kit ごとに作成する．様々な種類やバージョンのクライアントから複数回アクセスを行った通信データを用いることで Exploit Kit の通信の特徴的を出来る限り網羅的に抽出する．4 章の評価実験では，シグネチャは json 形式で出力するよう実装している．

図 4　シグネチャ作成フェーズ概念図

3.3 Exploit Kit 検知用シグネチャ

提案手法では，Exploit Kit の URI にアクセスした際に自動的に発生するリダイレクト先の URL の遷移の特徴により当該 Exploit Kit を検知する．このアクセス先 URL の遷移を表現するため，図 5 のようなグラフをシグネチャとして用いる．図 5 の A と B は被害者であるクライアントの最初のアクセス先 URL に対応し，A-1，A-2 は，それぞれ A から自動的にリダイレクトされた先の URL に対応する．各 Exploit Kit について収集フェーズで観測した全ての通信を解析し，それぞれ 1 つずつ遷移図を作成し，これを各 Exploit Kit のシグネチャとする．シグネチャの各ノードでは 7 つの値を保持している．以下にその値を説明する．

- id（ユニークな id）
- pre_id（遷移元のノードの id．遷移元が存在しない場合は 0）
- next_id（遷移先のノードの id．遷移先が存在しない場合は 0）
- path（アクセス先 URL のパス内のファイル名）
- pal（HTTP リクエスト内のクエリパラメータ群．収集フェーズで観測された全てのパラメータをリスト化したもの）
- re（クエリパラメータの正規表現）
- lastflg（このノードが URL 遷移の終端の場合に 1.）

各ノードでこれらの値を保持していることで，アクセス先遷移を表現できる．シグネチャの例を図 6 に示す．

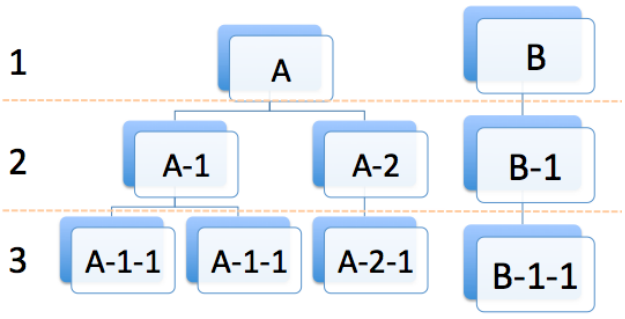

図 5　シグネチャ構成図

Vol.2014-DPS-158 No.35
Vol.2014-CSEC-64 No.35
2014/3/7

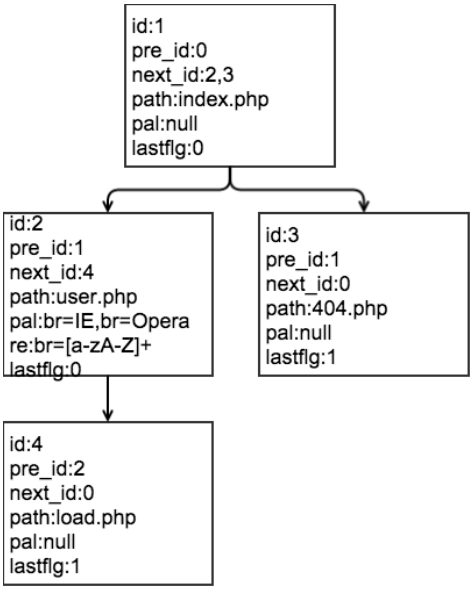

図 6　シグネチャの例

この例では id が 1 のノードが，最初のアクセス先であり，要求ファイル名は index.php であることが分かる．また，id が 2 のノードと id が 3 のノードに遷移していることから，自動的に user.php ファイルの要求に遷移する場合と，404.php ファイルの要求に遷移する場合があることを示している．さらに，id が 2 のノードの pal は br=IE，br=Opera となっており，アクセス時にクエリパラメータとして上記の 2 つが観測されたことを示している．観測されたパラメータ群を正規表現化したものが「re:br=[a-zA-Z]+」である．正規表現生成法は様々なものが存在するが，以下では，4 章の評価実験時に用いた方法を説明する．

- 正規表現について

シグネチャ内で使用している正規表現の作成方法を示す．まず pal の文字列を「変数名=値」と定義する．この形式でない場合はその文字列全てを値とする．正規表現は「(変数名）=（値の正規表現パターン)」という形となる．変数名がない場合は（値の正規表現パターン）のみである．

pal の全ての文字列に対して変数名により分類し，その変数名に対応する値の重複をなくす．変数名が無い場合は仮の変数名をつくり同様に処理する．ひとつの変数名に対して，次のフラグを用意する．

① 文字数フラグ（文字数が等しいか）
② [0-9]フラグ（文字列が数字のみで構成されているか）
③ [a-f]フラグ（文字列が a~f のみで構成されているか）
④ [a-zA-Z]フラグ（文字列が大小英字のみで構成されているか）
⑤ [0-9a-f]フラグ（文字列が数字と a~f のみで構成されているか）
⑥ [0-9a-zA-Z]フラグ（文字列が数字と大小英字のみで構成されているか）

これら全てのフラグの初期状態は True である．変数名に対応した全ての値に対して上記 6 個のフラグに対して解析を行い，フラグの内容に対応していない場合フラグを False にする．解析後①から順にチェックを行う．①のフラグが True の場合は，正規表現パターンに長さが追加され，文字列の中で固定文字が存在した場合それをパターンとする．②～⑥のフラグは順にチェックし最初に True となっているフラグの内容を正規表現のパターンとする．該当するものがない場合，任意の文字に合致するオールマイティなパターンとする．

例として次の 3 つのデータがある場合，

1. val=abcXXdef
2. val=ghiXXjkl
3. val=mnoXXpqr

この場合作成される正規表現は「num=[a-z]{3}XX[a-z]{3}」となる．

3.4 検査対象の通信データとのマッチング方法

生成したシグネチャと検査対象の通信データをマッチングする際は，まず，通信データから HTTP 通信のアクセス先の遷移を抽出し，シグネチャと一致するかどうかを比較する．具体的には，入口のノードから順番に遷移を比較していき，遷移数が閾値回数以上であり，かつ遷移先のノードの lastflag が 1 となった場合に，シグネチャとマッチしたと判断し，当該 Exploit Kit にアクセスしたと判断する．なお，4 章の実験では閾値回数は 1 回とした．

4. 評価実験

本章では，実際の Exploit Kit を対象に検知用シグネチャを自動作成し，当該 Exploit Kit を用いて作成した悪性サイトを当該シグネチャにより検知できるかの評価実験を行う．また，作成したシグネチャによる誤検知がないかを正常通信時のトラヒックとのマッチングにより検証する．

対象とする Exploit Kit は，入手が可能であり，サンドボックス内で動作させることが可能であった以下の 8 種類とした．

- adpack-2
- armitage
- Eleonore Exploit Kit
- fiesta-2
- g-pack
- ice-pack-3
- my-poly-sploit
- Seo Sploit Pack

Vol.2014-DPS-158 No.35
Vol.2014-CSEC-64 No.35
2014/3/7

4.1 シグネチャの作成

収集フェーズで使用するクライアント側の OS は Windows XP SP2 とし, 使用するブラウザおよびバージョンは, Internet Explorer (ver. 6, 8), Firefox (ver. 1.0.1, 1.5, 3.5), Opera (ver. 9.0, 10.0) の 7 種類である. それぞれのブラウザ, バージョンにおいて IP アドレスを変更しつつ 100 回アクセスを行い, 計 700 回分の通信からシグネチャを自動作成する.

例として Seo Sploit Pack を対象に作成したシグネチャを示す. まず, Internet Explorer 6 で 100 回アクセスした際の遷移の様子をグラフ化したものを図 7 に示す.

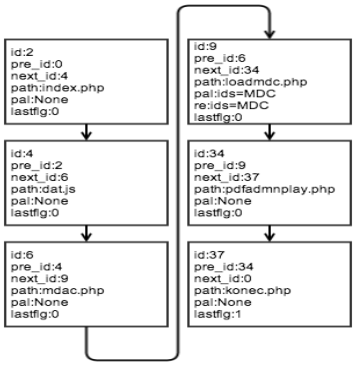

図 7 Internet Explorer 6 でアクセスした際の Seo Sploit Pack のアクセス先遷移

全体で 5 回の自動リダイレクトが発生し, 遷移中の分岐は無く, クエリパラメータにも多様性がないため, 非常に特徴的であり検知が容易な通信パターンといえる.

次に上記の 7 種類のブラウザおよびバージョンでの通信を利用して作成したシグネチャを示す. 図 7 のシグネチャと比べると, ノードの数, 遷移経路も増加して複雑になっている. ノードの中には同じ path のノードが多数存在するが, 遷移数 (遷移順) が異なれば別ノードとして扱う. このように全ての Exploit Kit においてシグネチャを作成する.

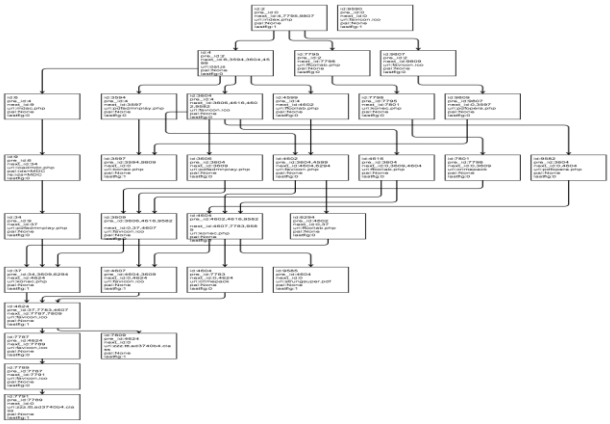

図 8 Seo Sploit Pack のシグネチャ

4.2 検知実験

前節で作成したシグネチャを基に検知実験を行う. シグネチャ作成の際に使用した全てのブラウザ, バージョン毎に再度 100 回分のアクセスを行い, Exploit Kit のシグネチャとマッチングを行うことで検知可否を調べた結果を表 2 に示す. 表の値は「検知数/検知対象数/アクセス数」である. ここで検知対象数とは, 100 回のアクセスのうち, 少なくとも 1 回はリダイレクトによる遷移が起こった場合の数である. 提案手法では, 検知の条件として最少のリダイレクト回数を閾値として設定しているが, 本実験では閾値を 1 としており, リダイレクトが発生しない通信は原理的に検知ができないため, 評価対象外とした. また, 表 2 の各ブラウザ, バージョンにおける検知結果は, 当該ブラウザに関するアクセス先遷移パターンのみを用いて検知した際の結果であり, ALL は全てのブラウザ, バージョンにおける結果を統合したシグネチャによる検知結果である. 表 2 より各ブラウザ, バージョンに対して, いずれも高い検知率となっている. 表中の塗りつぶしがおこなわれている部分は検知率が他に比べ低いが, この原因は検知対象の HTTP リクエストのサイズが大きくパケット 2 つに渡っており, 検知用スクリプトにおいて正しく処理ができずエラー終了していたためである. 表の該当部分の値の括弧内の値はエラーの回数であり, 正しく処理ができていれば高い検知率が期待される. AVG の列は総合したシグネチャにおいて各ブラウザ, バージョン毎を検知した際の平均値を示している. 上記理由において検知率が低くなった ice-pack-3

	Internet Explorer 6	Internet Explorer 8	Firefox 1.0.1	Firefox 1.5	Firefox 3.5	Opera 9.0	Opera 10.0	ALL	AVG
adpack-2	100/100/100	100/100/100	0/0/100	0/0/100	0/0/100	0/0/100	0/0/100	200/200/700	100%
armitage	100/100/100	95/100/100	0/0/100	0/0/100	0/0/100	0/0/100	0/0/100	195/200/700	98%
Eleonore Exploit Kit	100/100/100	100/100/100	100/100/100	100/100/100	10/10/100	0/1/100	1/1/100	412/412/700	100%
fiesta-2	97/100/100	0/0/100	0/0/100	0/0/100	0/0/100	100/100/100	0/0/100	199/200/700	100%
g-pack	100/100/100	100/100/100	0/0/100	0/0/100	0/0/100	0/0/100	0/0/100	200/200/700	100%
ice-pack-3	100/100/100	0/0/100	5(95)/100/100	9(91)/100/100	11(89)/100/100	0/0/100	0/0/100	125(275)/400/700	31%
my-poly-sploit	100/100/100	0/0/100	0/0/100	0/0/100	0/0/100	100/100/100	0/0/100	200/200/700	100%
Seo Sploit Pack	100/100/100	98/100/100	100/100/100	100/100/100	100/100/100	1/1/100	98/100/100	597/601/700	99%

表 2 検知実験-結果

を除けば最低 98%と非常に高い検知率である．

4.3 誤検知実験

各 Exploit Kit のシグネチャの誤検知について評価実験を行う．対象とする通信は 4 人のユーザがそれぞれ異なるマシンを用いて約 3 日間に渡り日常のネットサーフィンを行った際の HTTP 通信(80/TCP のみ)であり，通信量は約 1.5GB である．結果を表 3 に示す．

	HTTP
adpack-2	0
armitage	0
Eleonore Exploit Kit	0
fiesta-2	0
g-pack	0
ice-pack-3	0
my-poly-sploit	0
Sea Sploit Pack	0

表 3　誤検知実験-結果

各 Exploit Kit において誤検知は存在しなかった．いくつかの ExploitKit のシグネチャにおいては，入口となるノードの path は，正規サイトでも使用されている index.php であり，誤検知の可能性があるが検知における最低遷移回数の閾値を 1 回とすることで，誤検知を防ぐことができた．

5. 考察

5.1 Exploit Kit 検知用シグネチャの精度

本手法で自動作成する Exploit Kit 検知用シグネチャの見逃し率はリソースとなる通信の多様性が大きく影響する．今回の実験ではブラウザの種類，バージョンについては一定の多様性をもたせるようにしたが，他にも考慮しなければいけない条件が多々存在する．例えば，一般的に DBD 攻撃で狙われる，JRE や Adobe flash player など各種プラグイン，アプリの有無やバージョンによっても異なる通信が発生する可能性がある．想定される様々なクライアント環境を用意することが検知漏れの減少につながると考える．この技術要件は悪性サイトを検知するためのクライアントハニーポットと同様であるため，今後，クライアントハニーポット技術との連携が重要といえる．

5.2 実インターネット上で動作する悪性サイトからのシグネチャ生成

本稿の収集フェーズではサンドボックス内に自ら Exploit Kit を用いて悪性 Web サイトを構築し，これにアクセスすることでシグネチャ作成用の通信を収集した．しかし，この方法では，所持している Exploit Kit 以外には対応できない．

そこで，実インターネット上で動作している悪性サイトから通信データを収集することで，当該サイトを構築した Exploit Kit に関するシグネチャを作成する方法が考えられる．この際，サンドボックス内では容易にクライアント側の IP アドレスを変更してアクセスを行うことが出来るが，実インターネット上では ISP による動的 IP アドレス割当等を利用するといった工夫が必要である．また，2.3 節で述べたように悪性 URL の有効期間は短く，目的のサイトを見つけることは難しい．ハニーポット等と連携してサイトを巡回しながら通信を取得する方法も考えられる．

5.3 シグネチャの使用方法

今回作成したシグネチャは事前に収集した通信と事後的な比較しかしていないが，リアルタイムでの検知や DBD 攻撃のブロックにおいても使用できると考える．例えば論文[8]のように Web プロキシサーバに本手法で作成したシグネチャを使用した検知機能を付加する，または Web ブラウザのプラグインにおいても同様の機能を付加することができると考えられる．

6. まとめと今後の課題

本稿では，サンドボックス上に自ら Exploit Kit を構築しアクセスを行い，その際取得した通信から Exploit Kit 検知用シグネチャを自動作成する手法を提案した．本手法では通信に表れる遷移順や URL 等からより精度の高いシグネチャを作成できる可能性を示した．

今後の課題として，検知実験で発生したエラーに対応できるように実験環境の改善を行い，また様々な脆弱性に対応したクライアント環境を作成したい．5.2 節で述べたように収集フェーズを拡張し，外部の悪性サイトにアクセスした際の通信を取得する．また，5.3 節で述べたリアルタイム検知を行えるようにプロキシサーバなどに実装を行いたい．

謝辞

本研究の一部は，総務省情報通信分野における研究開発委託／ 国際連携によるサイバー攻撃の予知技術の研究開発／サイバー攻撃情報と マルウェア実体の突合分析技術／類似判定に関する研究開発により行われた．

参考文献

1 日本の大手出版社の Web サイトが Gongda 悪用ツールキットに利用される
http://www.symantec.com/connect/ja/blogs/web-gongda
2 Web ベースのマルウェア拡散経路：トラフィック流通システムの概要
http://www.symantec.com/connect/ja/blogs/web-1
3 最新の状態でないオペレーティングシステムを狙う BlackHole Exploit Kit
http://blog.webroot.co.jp/2011/12/28/%E6%9C%80%E6%96%B0%E3

%81%AE%E7%8A%B6%E6%85%8B%E3%81%A7%E3%81%AA%E3%81%84%E3%82%AA%E3%83%9A%E3%83%AC%E3%83%BC%E3%83%86%E3%82%A3%E3%83%B3%E3%82%B0-%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%82%92%E7%8B%99/
4 contagio malware dump
http://contagiodump.blogspot.jp/
5 An Overview of Exploit Packs（Update 20）Jan 2014
http://contagiodump.blogspot.jp/2010/06/overview-of-exploit-packs-update.html
6 CVE-2013-2511
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2551
7 Malware Domain List
http://www.malwaredomainlist.com/mdl.php
8 秋山満昭，八木毅，伊藤光恭“悪性 URL 群の木構造に着目した URL フィルタリングの粒度決定”ICSS2013
9 千葉大紀，森達哉，後藤滋樹“悪性 Web サイト探索のための優先巡回順序の選定法”CSS2012
10 酒井裕亮，佐々木良一“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”CSEC2013 DPS2013
11 笠間貴弘，神薗雅紀，井上大介“Exploit Kit の特徴を用いた悪性 Web サイトの検知手法の提案”CSS2013
12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”CSS2011