

2015年度 修士論文

HTTP 遷移の特徴分析による  
Web 感染型マルウェアの早期検出

提出日：2016年2月1日

指導：後藤滋樹教授

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻  
学籍番号：5114F036-1

小崎頌太

# 目次

<b>第 1 章</b>	<b>序論</b>	<b>4</b>
1.1	研究の背景	4
1.2	研究の目的	5
1.3	論文の構成	5
<b>第 2 章</b>	<b>HTTP</b>	<b>7</b>
2.1	HTTP の概要	7
2.2	HTTP 要求	7
2.2.1	GET ヘッダ	7
2.2.2	Referer (リファラ)	8
2.2.3	User-Agent	8
2.2.4	Accept	8
2.3	HTTP 応答	9
2.3.1	HTTP ヘッダ	9
2.3.2	HTTP ステータスコード	9
2.3.3	Content-Type	10
<b>第 3 章</b>	<b>Drive-by-Download 攻撃</b>	<b>11</b>
3.1	Drive-by-Download 攻撃	11
3.2	Blackhole	12
3.3	Redkit	13
3.4	難読化された JavaScript	13
<b>第 4 章</b>	<b>関連研究</b>	<b>14</b>
4.1	検知を目指した不正リダイレクトの分析	14
4.2	不正リダイレクトの検出による悪性 Web サイト検知システム	14
4.3	Detecting Malicious HTTP Redirections by User Browsing Activity	16

---

<b>第 5 章</b>	<b>提案手法</b>	<b>17</b>
5.1	提案手法の概要	17
5.2	先行研究	17
5.2.1	概要	17
5.2.2	アクセス遷移	17
5.2.3	次のペアの推定	18
5.3	先行研究と本手法の相違点	18
5.4	想定する解析システム	20
5.5	本手法の手順	21
5.6	機械学習で用いる特徴量	21
5.6.1	各特徴量と根拠	21
5.6.2	累積悪性度	22
<b>第 6 章</b>	<b>性能評価</b>	<b>23</b>
6.1	評価に用いるデータ	23
6.2	実験の概要	24
6.3	実験 1 検知精度の測定	24
6.4	実験 2 特徴量の調整	25
6.5	実験 3 検知に要した遷移数の測定	25
6.6	考察	26
<b>第 7 章</b>	<b>結論</b>	<b>30</b>
7.1	まとめ	30
7.2	今後の課題	30
7.2.1	次のペア推定	30
7.2.2	特徴量効率化の手法	31
	謝辞	32
	参考文献	33

# 図一覧

2.1	GET 要求ヘッダ	8
2.2	HTTP 応答ヘッダの例	9
3.1	Drive-by-Download 攻撃	12
3.2	難読化されたコード	13
4.1	分類に使用する遷移の種類	15
5.1	アクセス遷移	18
5.2	次のペア推定	19
5.3	手法 (-D)	19
5.4	システム図	20
5.5	本手法の概要	21
5.6	特徴量	22
5.7	決定木の共通部分	22
6.1	データセットから除外する条件	23
6.2	検知できた遷移数と検体の割合 (先行研究)	27
6.3	検知できた遷移数と検体の割合 (本手法)	28
6.4	検知できた遷移数と検体の割合 (特徴量調整後)	29

# 表一覧

2.1	HTTP ステータスコード	10
4.1	危険なダウンロード	15
5.1	手法 (-D) の適応前後	20
6.1	使用するデータセットの収集日と検体数	24
6.2	本手法と先行研究の性能比較	25
6.3	特徴量の調整	25
6.4	特徴量の調整結果	26
6.5	マルウェアの検出に要した遷移数	26

# 第 1 章

## 序論

### 1.1 研究の背景

改ざんされた Web サイトから攻撃サイトに誘導し、マルウェアに感染させる Web 感染型マルウェアが依然として猛威を振るっている [1]。この背景に、iframe、JavaScript が抱える脆弱性の悪用によるマルウェアの高度化・巧妙化がある [2]。Web 感染型マルウェアに感染させる攻撃手法は、Drive-by-Download 攻撃 (第 3 章) と呼ばれる。Drive-by-Download 攻撃では、ユーザに気付かれないようにマルウェアをダウンロードする。これは、Web ブラウザやプラグインの脆弱性を利用するため、マルウェアに感染した場合には、個人情報の漏洩や Web ページの改ざんなどの、さまざまな問題が、ユーザの気付かない間に発生する。

Web 感染型マルウェアの感染を防ぐための技術としては、ブラックリストによるフィルタリングや、シグネチャによるパターンマッチングが利用されている。これらの技術は既知の攻撃に対しては非常に有効に働き、ブラックリストに登録されている IP アドレス、ドメインからの攻撃や、シグネチャと一致するマルウェアからの攻撃ならば完全に防御することができる。しかし、これらの技術は原理的に未知の攻撃を防ぐことはできないという短所がある。未知の攻撃を防ぐための技術としては、マルウェアのプログラム構造と挙動を元に検知を行うヒューリスティック法がある。この技術を用いることで、未知の攻撃によるマルウェア感染行為でも防ぐことができる。しかし、この技術には誤検知が多いという短所がある [3]。よってヒューリスティック法では、未知の攻撃を完全に防ぐことは困難である。さらにダウンロードされたプログラムを解析しなければいけないというコストもかかる。

## 1.2 研究の目的

本研究は，HTTP 通信のアクセス遷移を解析することにより，Web 感染型マルウェアの検知が可能であることを示す．具体的には，HTTP 要求における URL の変化を解析し，Web 感染型マルウェアの自動ダウンロードと，正常な実行ファイル<sup>1</sup>の手動ダウンロードを識別することで，Web 感染型マルウェアを検知する手法を提案する．

本手法の特徴は，HTTP 通信のリダイレクトに注目することで，悪性ファイルをダウンロードする前の通信データのみ使用し，検出することができる．さらに，ダウンロードされるプログラムを解析しないので，少ない解析コストで，未知のプログラムに対する攻撃の検知が可能である．

## 1.3 論文の構成

本論文は以下の章により構成される．

### 第 1 章 序論

本論文の概要を述べる．

### 第 2 章 HTTP

HTTP (Hypertext Transfer Protocol) の要求と応答について説明する．

### 第 3 章 Drive-by-Download 攻撃

Web ブラウザの脆弱性を利用して，マルウェアを強制的にダウンロード，インストールする Drive-by-Download 攻撃について解説する．

### 第 4 章 関連研究

本論文に関連する研究を紹介する．

### 第 5 章 提案手法

本論文の提案手法を説明する．

### 第 6 章 性能評価

提案手法の性能評価と考察を行う．

---

<sup>1</sup>実行ファイルのうち，不正もしくは悪質な動作を行わないもの．

## 第 7 章 結論

本論文の結論を述べ、残された課題を示す。

## 第 2 章

# HTTP

### 2.1 HTTP の概要

HTTP (Hypertext Transfer Protocol) は、HTML (HyperText Markup Language) 文書や画像をサーバとクライアントの間でやり取りするためのプロトコルである。

HTTP の通信は、クライアントからの HTTP 要求と、サーバからの HTTP 応答という二種類のメッセージの送受信によって行われる。クライアントは URL の入力、ブラウザの画面上の操作により、サーバに HTTP 要求 (第 2.2 節) を送信する Web サイト内での操作により、サーバに HTTP 要求 (第 2.2 節) を送信する。HTTP 要求を受け取ったサーバは、まずその要求を受け付けるか拒否するかを判断する。要求を受け付ける場合は、ステータスコード (第 2.3.2 節) に続いて要求の処理結果が HTTP 応答 (第 2.3 節) として送信される。

### 2.2 HTTP 要求

#### 2.2.1 GET ヘッダ

クライアントは、URL に対するリソースを送信するようサーバに GET 要求を送信する。図 2.1 に HTML ファイルに対する GET 要求のヘッダを示す。

図 2.1 のパケットは GET 要求 (Request Method: GET) であり、`www.xxx.zzz/index.html` という URL に対する `index.html` というリソースを取得しようとしている。またその他に、リファラ (Referer, 第 2.2.2 節) や、クライアントの名前 (User-Agent, 第 2.2.3 節)、希望する HTTP データの圧縮方法 (Accept-Encoding) のような GET 要求の付加的な情報も含んでいる。

```
GET /index.html HTTP/1.1
Accept: */*
Referer: http://hoge.com/index.html
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0; Windows NT 5.1;)
Host: www.xxx.zzz
Connection: Keep-Alive
```

図 2.1: GET 要求ヘッダ

### 2.2.2 Referer (リファラ)

GET 要求のヘッダには Referer (リファラ) と呼ばれるフィールドが存在する。Referer とは、別のサイトに移動した時に、参照元サイトの URL が入るフィールドである。Referer に記載されている情報により、どこから訪問してきたのか、また、サイト内でどのようなコンテンツを参照したのかがわかる。すなわち、自分が参照した URL が時間的に後のセッション内の GET 要求ヘッダ内に存在した場合、ページの遷移を特定することができる。図 2.1 では、Referer に `http://hoge.com/index.html` がセットされているから、`http://hoge.com/index.html` から `http://www.xxx.zzz/index.html` が参照されたということになる。

### 2.2.3 User-Agent

User-Agent は、HTTP 要求を送信したクライアントの名前を表すフィールドである。サーバはこのフィールドの値により、クライアントの種類に応じて処理や統計調査を行うことが可能となる。また、多くの Web ブラウザでは User-Agent に Mozilla という文字が含まれている [4]。これは、Netscape Navigator という、シェアが非常に高かった Web ブラウザが、User-Agent に Mozilla を指定していた名残である。なお、現在広く利用されている PC 用 Web ブラウザのうち、User-Agent に Mozilla が含まれないものは、Opera のみとなっている [4]。

### 2.2.4 Accept

Accept は、HTTP 要求を送信したクライアントが受け入れることができるファイルの形式 (MIME タイプ) を表すフィールドである。サーバが HTTP 応答として送信するファイル形式

が、HTTP 要求の `Accept` に含まれない形式の場合には、サーバは HTTP ステータスコード (第 2.3.2 節) として 406 を返す。また、図 2.1 のように、`Accept` に `*/*` が指定されている場合には、クライアントはどんな形式のファイルでも受け入れ可能であることを意味している。このフィールドは HTTP において必須ではない。しかし、クライアントが受け入れ可能なファイルを表すという重要性から、ほとんどのブラウザでは HTTP 要求の中にこのフィールドを含めている。

## 2.3 HTTP 応答

### 2.3.1 HTTP ヘッダ

クライアントから送信された HTTP 要求に対し、サーバは HTTP 応答を送信する。図 2.2 に HTTP 応答のヘッダの一例を示す。図 2.2 は、図 2.1 の HTTP 要求に対する HTTP 応答のヘッダであり、1 行目にプロトコルのバージョン、HTTP ステータスコード (第 2.3.2 節) が記述され、メッセージが生成された日付 (`Date`)、ファイルの長さ (`Content-Length`) やファイルの種類 (`Content-Type`、第 2.3.3 節) といった付加的な情報も記述されている。

```
HTTP/1.1 200 OK
Server: Apache
Last-Modified: Mon, 14 Oct 2013 13:00:27 GMT
Accept-Ranges: bytes
Content-Length: 3296
Content-Type: application/zip
Cache-Control: max-age=205
Expires: Mon, 14 Oct 2013 13:08:35 GMT
Date: Mon, 14 Oct 2013 13:05:10 GMT
Connection: keep-alive
```

図 2.2: HTTP 応答ヘッダの例

### 2.3.2 HTTP ステータスコード

HTTP 応答のヘッダには、サーバが付加する HTTP ステータスコードと呼ばれるものが存在する。ステータスコードは 3 桁の数字からなり、クライアントにリクエストが成功したかど

うか知らせたり，追加の処理が必要であることを示すときに利用される．HTTP ステータスコードの一覧を以下の表 2.1 に示す．ここで，例えば図 2.2 における HTTP ステータスコードは 200 OK であるから，正常に受信されリクエストが成功した例である．

表 2.1: HTTP ステータスコード

1xx	通知のためのステータスコード (e.g. 100 Continue 処理の継続中)
2xx	成功を表すステータスコード (e.g. 200 OK 成功)
3xx	リダイレクトを表すステータスコード (e.g. 301 Moved Permanently リソースが Location ヘッダに示された場所に移動)
4xx	クライアント側でのエラーを表すステータスコード (e.g. 401 Unauthorized 認証されていない)
5xx	サーバ側でのエラーを表すステータスコード (e.g. 500 Internal Server Error サーバ内部のエラー)

### 2.3.3 Content-Type

Content-Type は，サーバが送信するデータの形式を表すフィールドである．クライアントは，このフィールドを参照し，受信データの形式を知ることによって，適切な処理を行うことができる．このフィールドでは，ファイル形式は「タイプ名/サブタイプ名」という構造の MIME タイプで表記される．例えば，サーバから JPEG 形式の画像ファイルを受信した場合，Content-Type には image/jpeg が記載される．なお，MIME タイプは 1 種類のファイル形式に対し，1 種類の MIME タイプが対応するわけではなく，1 つのファイル形式に対し，複数の MIME タイプが存在することがある．具体的には，Javascript ファイルには，application/javascript や text/javascript という MIME タイプがある．

## 第 3 章

# Drive-by-Download 攻撃

### 3.1 Drive-by-Download 攻撃

Drive-by-Download 攻撃とは、Web ブラウザやプラグインの脆弱性を利用し、ユーザの気付かない間にマルウェアをダウンロード、インストールさせる攻撃のことである。Web ブラウザやプラグインの脆弱性には iframe や難読化された JavaScript が使用され、ユーザが Web サイトを閲覧しただけで攻撃を受けてしまう。このためユーザが感染を防ごうと対策することは困難である。この攻撃によって感染するマルウェアは、Web 感染型マルウェアと呼ばれる [2]。

Drive-by-Download 攻撃では、図 3.1 のように複数のサイトを移動することで攻撃を複雑化し、難読化が施されたスクリプトによって HTTP リダイレクトを発生させて悪性サイトへ誘導することが多い。誘導された先のサイトには Exploit kit と呼ばれるツールが設置されている事が多く、Oracle Java や Acrobat/Reader、Adobe Flash、Java Runtime Environment の脆弱性を利用しマルウェアに感染させようとする [1, 2, 5]。また、攻撃の踏み台や入り口としても利用される。Web ページは、データベースの脆弱性をついた SQL インジェクションや、不正に ID、PASSWORD を入手しアクセスする不正アクセスによって悪性スクリプトが挿入され、攻撃に利用される。また、Twitter などのソーシャルネットワークサービス (SNS) や短縮 URL といったサービスを利用することで、より巧妙に、ユーザから気付かれないよう攻撃の入り口サイトへ誘導するような手口も増えている [6]。

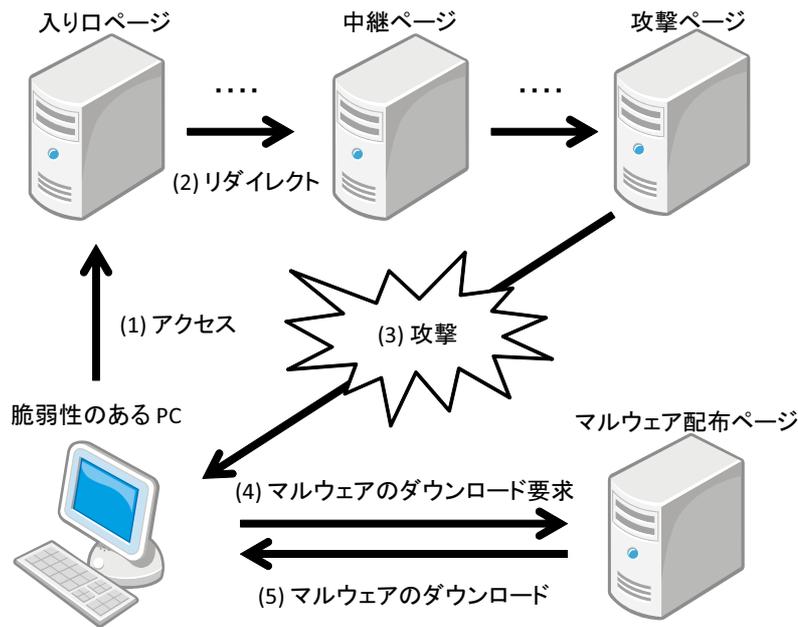


図 3.1: Drive-by-Download 攻撃

## 3.2 Blackhole

Blackhole とは、Web 感染型マルウェア配布用攻撃ツールである [5]。このツールにより Web 感染型マルウェアに感染したユーザ PC は、その PC が管理する Web ページが改ざんされ、攻撃の踏み台として利用されてしまう。このことにより第三者にも被害を与える事になる。このツールは、ブラックリストによるフィルタリングを回避するために、Drive-by-Download 攻撃のリダイレクト先を変更させる機能や、作成されたマルウェアがアンチウイルスソフトにより検知されるかどうかを確かめる機能を保有している [7]。また、Blackhole は管理用インタフェースが充実しており、攻撃者が欲しい機能をマルウェアに簡単に付加することができる。そのため、マルウェア作成の知識がない人でも攻撃が可能である。なお、Blackhole は頻りにプログラムが更新されており、既知の脆弱性を用いる攻撃だけではなく、ゼロデイ攻撃<sup>2</sup>も可能となっている。そのため、Blackhole による攻撃は、OS や Web ブラウザのバージョンが最新であるとしても防ぐことが困難である。

なお 2013 年 10 月に Blackhole の首謀者とされる人物が逮捕されたこともあり、攻撃キットの別の被害率で 8 位となるなど被害は減少している。しかし、Blackhole を改良した攻撃ツールが使用されるなど、現在でも Blackhole 系統の攻撃ツールによる被害は継続している [8, 9]。

<sup>2</sup>脆弱性に対応する修正パッチが提供される前に行われる攻撃のこと。

### 3.3 Redkit

Redkit という攻撃ツールが 2013 年から台頭している。Blackhole と同様に感染したユーザ PC が管理する Web ページを踏み台に利用する。Java, Adobe PDF, Flash の脆弱性を標的にしている Blackhole に対し, Redkit は Java の脆弱性だけを攻撃する。ユーザにダウンロードさせるコンテンツを攻撃サーバから入手し, 踏み台サイトから配信する。これは, 検出を困難にするため、ユーザからは, 踏み台サイトからのダウンロードのように見えるからである [9]。

### 3.4 難読化された JavaScript

難読化された JavaScript に, HTTP リダイレクトを発生させるリンクを埋め込む手法を使用して攻撃が行われる。これは, パターンマッチングによる検知を回避するためである。また, 難読化された JavaScript を動的に生成することで, 可読性を低くする手法も使用される。alert("Hello, World!"); を難読化支援サイト [10] に適用した結果を以下の図 3.2 に示す。

```
eval(function(p,a,c,k,e,r){e=String;if(!".replace(/\\/,String))){while(c--)r[c]=k[c]
——c;k=[function(e){return r[e]};e=function(){return '\\w+'};c=1};
while(c--)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return
p}('0("1, 2!!");',3,3,'alert—Hello—World'.split('—'),0,{}))
```

図 3.2: 難読化されたコード

文字列を評価する eval 関数および, 文字列の操作をする String オブジェクトの replace 関数, split 関数が用いられている。他にも escape 関数による ASCII 変換や, fromCharCode 関数による文字コード変換などが行われている。難読化の手法は, 複数存在し, 難読化手法はコンテンツの種類に応じて変更されるため, シグネチャによる検知は難しい。さらに, 難読化を解除して解析する手法も存在するが, 解析にコストがかかるという短所を持つ。

## 第 4 章

### 関連研究

#### 4.1 検知を目指した不正リダイレクトの分析

寺田ら [11] は、悪性通信を収集したデータセットにおける、危険なダウンロードに至る URL の判別を行っている。危険なダウンロードとは、具体的には表 4.1 に該当するものである。

具体的な手法は、まず HTTP リダイレクトを図 4.1 に示す 6 種類に分類している。なお図 4.1 における巡回 URL リストとは、データセットが提供している、アクセスしただけで攻撃を受ける URL の一覧のことを指す。図 4.1 は URL の情報がリダイレクトする前の HTTP 応答に存在するか、存在するとき、HTTP 応答のどの部分に存在するかにより分類している。この分類を用いて、悪性通信のデータセットにおいてどの遷移が多いのか統計的に調査している。その結果、分類 (6) すなわち「組の中に次の組を指す URL が無い」場合が危険なダウンロードを含む HTTP 通信に多いことを示している。さらに機械学習により (6) 「HTTP 応答の中にリダイレクト先を指す URL が無い」という遷移が、危険なダウンロードを識別する上で有効な指標となるかを検証している。

この手法はリダイレクト先がわかっているデータセットの中での検知手法なので、複数の HTTP 通信が混在する実ネットワークには適応できないという短所がある。

#### 4.2 不正リダイレクトの検出による悪性 Web サイト検知システム

安藤ら [12] は、Drive-by-Download 攻撃の不正リダイレクトに注目し、ホームページのリンクの深さと広がりという概念を用いて、攻撃の感染活動に関わる異常な通信を検知している。

表 4.1: 危険なダウンロード

ファイル	Content-type
pdf	application/pdf
swf	application/x-shockwave-flash
BIN	application/octet-stream application/x-msdownload application/x-download application/x-msdos-program

- (1) 巡回 URL リストに載っている URL が HTTP 応答に指定されている
- (2) HTTP 応答のステータスが 3xx である
- (3) HTTP 応答に URL が記載されている
- (4) リダイレクト先とホストが同じで、HTTP 応答に URL が記載されている
- (5) リダイレクト先と同じホスト
- (6) HTTP 応答の中にリダイレクト先を指す URL が無い

図 4.1: 分類に使用する遷移の種類

この手法は、ブラウザ上でのユーザの操作と、ブラウザと Web サーバでやり取りされる通信データを観測し、リダイレクトのつながりをリンクの深さと広がりを用いて表すことで検知する。リンクの深さとは、ユーザが指定した URL からダウンロードした HTML を第 0 層として、そこから呼び出されるファイルを第 1 層 (JavaScript ファイルや、CSS ファイル)、さらにそこから呼び出される画像などのファイルを第 2 層.....、と表現する。リンクの広がりとは、異なるドメインへの呼び出しが起こった場合に広がりが増える、と表現する。ユーザ起因の URL アクセスでなく、深さと広がりが増える場合はリダイレクトが複数発生していると推定されるため、異常な通信とみなして、検知している。

この手法は各ユーザの操作を観測しなければいけないので、検知システムに導入コストがかかってしまうという短所が存在する。

### 4.3 Detecting Malicious HTTP Redirections by User Browsing Activity

Mekky ら [13] は、ISP で観測された通信データを解析し特徴を抽出、IDS のアラートを正解データとして、悪性なりダイレクトと良性なりダイレクトとの分類を行っている。使用している特徴としては、ドメインの変更数、リダイレクトの数に関するものである。

この手法はドメイン名の判別に時間がかかるという短所があり、実際リアルタイムで検知するのは困難である。

# 第 5 章

## 提案手法

### 5.1 提案手法の概要

本章では，HTTP アクセス遷移を解析し，正常な通信と Web 感染型マルウェアによる通信とを識別する方法を述べる．通信データとしては，悪性ファイルをダウンロードする前の部分のみ使用する．この方法により，Web 感染型マルウェアに感染する前に検知可能であり，リアルタイムに検知する事が可能となる．

### 5.2 先行研究

#### 5.2.1 概要

筆者は，寺田ら [11] の手法をもとに，HTTP アクセス遷移を解析して，正常な通信と，Web 感染型マルウェアの自動的なダウンロードを識別する方法を提案した．具体的には，まず HTTP 要求とそれに対する応答を一つの「ペア」として扱う．あるペアから次のペアを推定し，機械学習を用いて HTTP 通信を分類する手法である．この手法が識別に対して有効である事を示した [14] ．

#### 5.2.2 アクセス遷移

HTTP 通信は複数のペアから構成される場合がある．図 5.1 に示すように，あるペアとその次のペアの URL が異なる場合をアクセス遷移と呼ぶ．

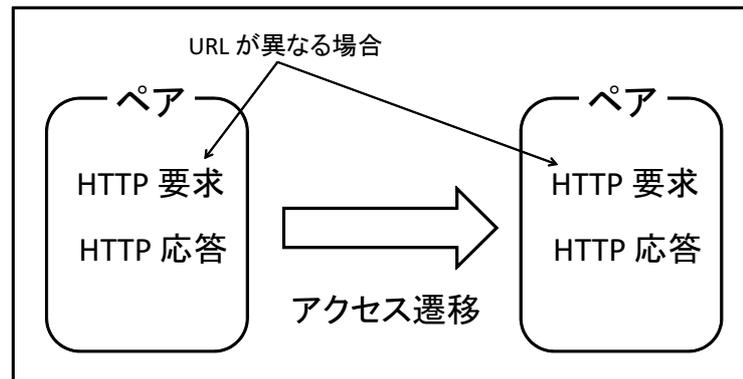


図 5.1: アクセス遷移

### 5.2.3 次のペアの推定

寺田らの手法における短所である，複数の HTTP 通信が混在したネットワークでは検知できない事 (4.1) への改良策として，以下のペアの推定を行う．

HTTP 応答において URL が難読化されておらず，その内容が読み取れる場合には，その URL を用いて次のペアを構成する．具体的には解析の対象となっているペアの HTTP 要求よりも時間的に後に出現するペアの中から，当該の URL を持つペアを検索して次のペアとする．なお本手法では，次のペアを指す URL が明示されていない場合には次のように推定する．

- ・ HTTP 要求内の Host header を手掛かりとして，そのホスト名を含む URL を持つペアであり，現在までの解析においては，他のペアからの遷移先になっていない独立したペアを次の遷移先と見なす．

## 5.3 先行研究と本手法の相違点

本手法は先行研究 (第 5.2 節) をもとに，より高精度な検出法を提案する．悪質な通信を高精度に検出するために，図 5.3 のような悪性ファイルをダウンロードする前の部分のみ使用する事を考える．この手法を以後「手法 (-D)」と呼ぶ．手法 (-D) により，Web 感染型マルウェアに感染する前に検知しないと誤検知と判定されるので，より高精度な検知ができると考える．

先行研究に手法 (-D) を適応する前と後の Accuracy を表 5.1 に示す．表 5.1 より TPR の低下がみられる．つまり，先行研究では，悪性ファイルをダウンロードする段階での検知が少なからずみられるという事がわかる．

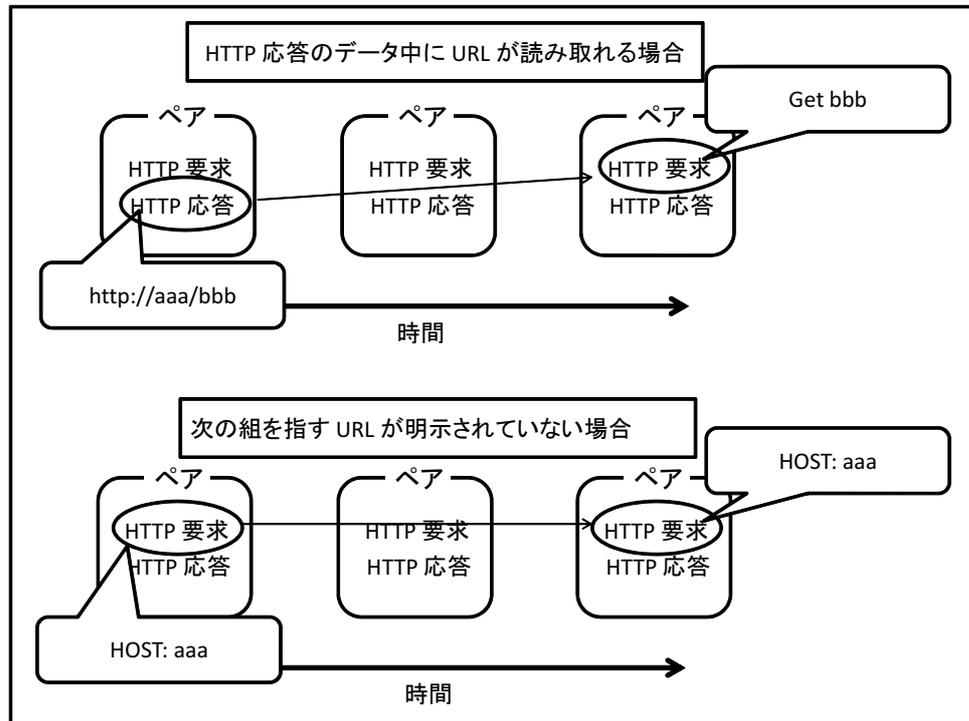


図 5.2: 次のペア推定

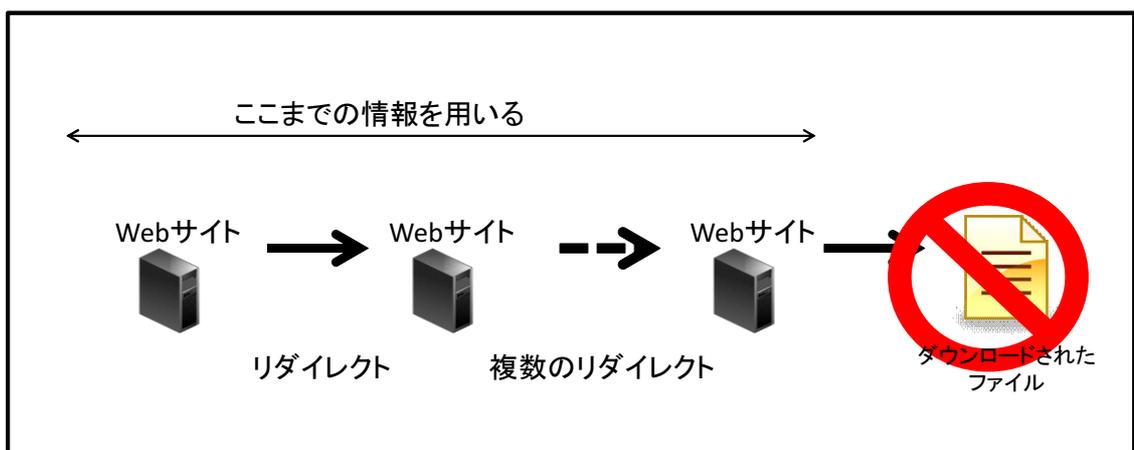


図 5.3: 手法 (-D)

表 5.1: 手法 (-D) の適応前後

評価手法	適応前 [%]	適応後 [%]
Accuracy	98.52	84.93
TPR	98.85	92.86
FPR	1.68	26.33

## 5.4 想定する解析システム

図 5.4 は、本手法の実用化を想定した解析システムの解説図である。解析システムはルータなどのユーザを集約する装置に実装され、「検知」と「分類」により Web 感染型マルウェアからの脅威を防御する。本手法はその中の「検知」部分を担うものである。

ユーザを集約する装置に実装することを想定しているため、ユーザの操作を観測できない、解析が軽量である、といった条件が必要となる。この 2 つの条件より、関連研究 (4 章) の安藤ら [12] の手法、Mekky ら [13] の手法は適応できない。

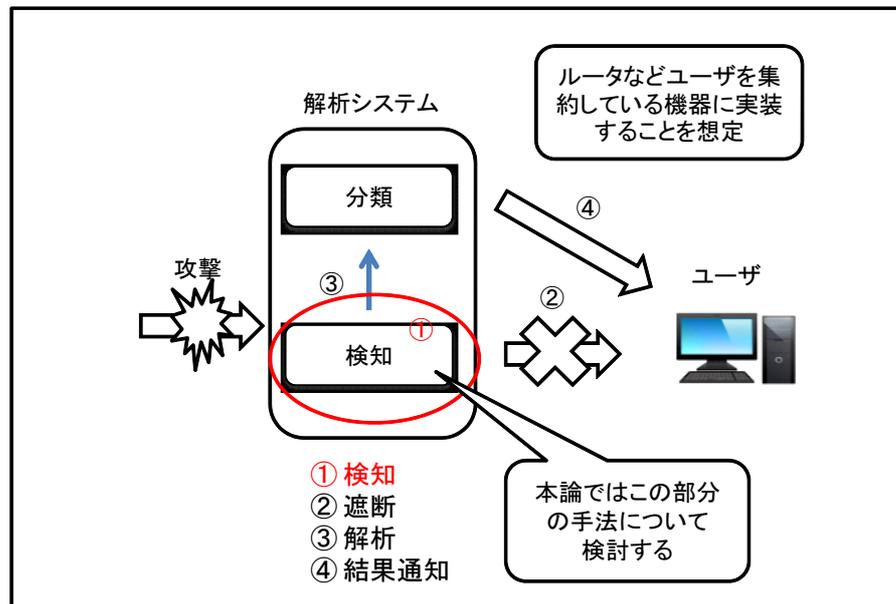


図 5.4: システム図

## 5.5 本手法の手順

本手法は，図 5.5 に示す 3 つのステップからなる．ステップ 1 では，通信データから HTTP 要求と，それに対する応答を取り出し，ペアを構成する．これを，正常な通信データ，Web 感染型マルウェアに感染する際の通信データについて行う．さらに，ペアからのアクセス遷移を解析し，HTTP 通信を生成する．またデータセットは，Drive-by-Download 攻撃の通信データを集めた D3M2015 [15] を使用している．これは，遷移する順にペアを並べ，他からの遷移がないペアから，他への遷移のないペアまでを一つの HTTP 通信とすることである．

ステップ 2 では，ステップ 1 で生成したペアに対して機械学習で用いる特徴を抽出する．この特徴に基づき機械学習を行う．各ペアについての判定を可能とするため交差検定法を用いる．交差検定法はデータを分割し，そのうち一つを教師データ，残る部分を学習データとする．それをすべての分割された部分に適用する．よってすべてのデータを学習データとして判定が可能である．

ステップ 3 では，ステップ 2 で求めた判定をもとに HTTP 通信が正しく判定されているかを検証する．また，Web 感染型マルウェアに感染する際の通信データについて，低遷移での検出が可能か計算を行う．これは，Web 感染型マルウェアに感染する際の通信データであると判定した時点まで遷移が何回起きたかを算出することである．

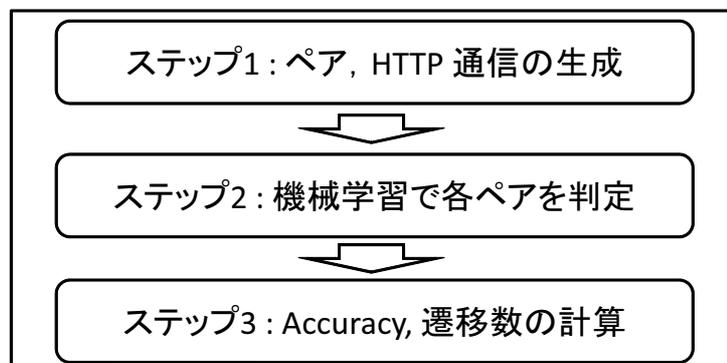


図 5.5: 本手法の概要

## 5.6 機械学習で用いる特徴量

### 5.6.1 各特徴量と根拠

機械学習で用いる特徴量として，図 5.6 に示す 7 つの特徴量を使用した．アクセス遷移の有無は既存研究である寺田ら [11] の手法 (第 4.1 節) より，(6) 「ペアの中に次のペアを指す URL

が無い」ものを指す。X-Powered-By header は実質的管理者でないと変更できない [16]。また、WEB サーバの管理者が非表示にすべきヘッダ情報であることにより、存在すると悪性である可能性が高くなる。Referrer header は悪性リダイレクトには存在しない確率が高い事が知られている [17]。

- (1) アクセス遷移の有無 (NotUrl)
- (2) 時間あたりの遷移数 (TransPerTime)
- (3) X-Powered-By header の有無 (PhpVer)
- (4) Referrer header の有無 (Ref)
- (5) HTTP データの大きさ (Data)
- (6) 遷移が始まってからそのペアまでの遷移数 (Trans)
- (7) それまでのペアの累積悪性度 (MalGrade)

図 5.6: 特徴量

### 5.6.2 累積悪性度

図 5.6 の MalGrade について、本手法では「遷移が始まってから、そのペアまでの各特徴の和」を用いた。ここでの各特徴とは TransPerTime, PhpVer, Ref, である。3 つの特徴を使用した理由は、データを分割し、機械学習である決定木にかけたところ、結果すべてが、図 5.7 の構造を持っていたからである。決定木の性質上、判定に重要なファクターであると考えられるので、上記 3 つの特徴を用いた。

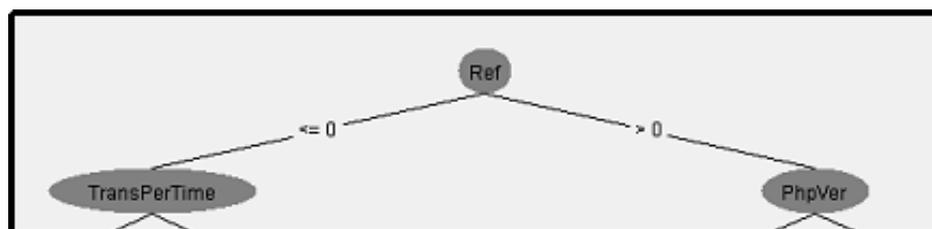


図 5.7: 決定木の共通部分

## 第 6 章

# 性能評価

### 6.1 評価に用いるデータ

Web 感染型マルウェアに感染する際の通信のデータセットとして、D3M2015 を使用する [15]。これらのデータセットは、NTT セキュアプラットフォーム研究所の高対話型の Web クライアントハニーポット Marionette [18] を使って収集された Web 感染型マルウェアの観測データ群である。Marionette の OS は Windows XP SP2 で、使用する Web ブラウザは Internet Explorer 6.0、導入されているプラグインは Adobe Reader、Flash player、Win Zip、Quick Time、JRE である。なお、Marionette は、ダウンロードされたマルウェアの実行を許可しないため、感染後のマルウェアの通信挙動がデータに含まれることはない。

正常な通信のデータとしては、早稲田大学の対外接続回線において収集したデータを使用する。なお、収集したデータには、悪質な通信も含まれる。そのため、図 6.1 の条件でフィルタリングを行うことで、悪質な通信を除外した。

HTTP リクエストのヘッダ部が以下のいずれかを満たす

- Accept フィールドが存在しない (第 2.2.4 節より)
- User-Agent フィールドに Mozilla, Opera のどちらとも含まれていない (第 2.2.3 節より)
- User-Agent フィールドにボットを連想させる文字が含まれている  
api, application, bat, bot, crawl, exe, hunny, pot, program

図 6.1: データセットから除外する条件

悪性データセットおよび良性データセットの収集日、HTTP 通信を一つの検体としたデータの数を表 6.1 に示す。各データセット共に、1000 組のペアを抽出し、HTTP 通信を作成してい

る。検体数に差が生じる原因は、ペアから HTTP 通信を作成する過程で、図 6.1 の条件でフィルタリングされたパケットや、第 5.3 節で述べた手法 (-D) により除外されたパケットが生じるためである。

表 6.1: 使用するデータセットの収集日と検体数

データセット名	収集日	検体数
悪性 (D3M2015)	2014/ 4/11, 5/ 2, 2015/ 2/ 8	515
良性	2015/10/23, 11/9	191

## 6.2 実験の概要

提案手法の性能評価を行うために、3つの実験を行う。まず、実験1では、検体を SVM (Support Vector Machine) を用いて Accuracy を求める。これは良性通信と悪性通信を判別可能かを評価するために行う。良性、悪性それぞれの検体を三つのグループに分け、交差検定法を用いて Accuracy を求めた。実験2では、特徴量の調整を行う。特徴量を減らし、Accuracy を求め、最も精度が良かったものについて TPR, FPR を求めた。実験3では、実験1, 2においてマルウェアの検知に要した遷移数を求めた。これは Web 感染型マルウェアがどれだけ少ない遷移数で検知できるかを評価するために行う。

## 6.3 実験1 検知精度の測定

各検体を SVM にかけて検知精度を測定した。性能比較のために第 5.1 節で説明した先行研究に手法 (-D) を適応したもの (以後「先行研究 (-D)」と呼ぶ) と比較した。結果を表 6.2 に示す。表中の Accuracy は良性、悪性データ全体で正しく判定した割合、TPR (True Positive Rate) は Web 感染型マルウェアを正しく Web 感染型マルウェアと判定した割合、FPR (False Positive Rate) は正常なファイルを Web 感染型マルウェアであると判定した割合である。

表 6.2: 本手法と先行研究の性能比較

評価手法	本手法 [%]	先行研究 (-D) [%]
Accuracy	95.58	84.93
TPR	94.24	92.86
FPR	3.69	26.33

## 6.4 実験 2 特徴量の調整

実験 2 では、特徴量の調整を行い、より精度を出すことを目指す。7 つある特徴量の MalGrade (累積悪性度) 以外を一つずつ減らしていき 6 つの特徴で SVM を用いる。それぞれについて Accuracy を求め、一番精度がよかったものについて TPR, FPR を求めた。

表 6.3: 特徴量の調整

削除する特徴量	Accuracy [%]
NotUrl	81.02
TransPerTime	81.02
PhpVer	96.18
Ref	71.25
Data	92.21
Trans	81.02

表 6.3 より、Phpver (X-Powered-By header の有無) がないときの Accuracy が高いことが判明した。Phpver の特徴を削除する前後の Accuracy, TPR, FPR を表 6.4 に示す。

## 6.5 実験 3 検知に要した遷移数の測定

マルウェアの検出に要した遷移数を調べるために、検知できた遷移数と実行ファイルをダウンロードするまでの遷移数を算出した。これは、実験 1, 2 での検知結果を使用している。結果

表 6.4: 特徴量の調整結果

評価手法	調整後 [%]	調整前 [%]
Accuracy	96.18	95.58
TPR	92.15	94.24
FPR	2.34	3.69

を図 6.2, 6.3, 6.4 に示す。また表 6.5 には (検知できた遷移数) / (実行ファイルをダウンロードするまでの遷移数) を計算した平均値を示す。

表 6.5: マルウェアの検出に要した遷移数

手法	平均値
先行研究	0.30
本手法	0.27
特徴量調整後	0.28

## 6.6 考察

実験 1 から実験 3 までの考察を行う。

実験 1 により、悪性ファイルをダウンロードするデータを含まない通信データのみで、本手法が良性通信と悪性通信との判別に有効であることがわかる。また、先行研究と先行研究(-D)との違いは「悪性ファイルをダウンロードする通信を含む」か否かなので、TPR の変化は「悪性ファイルをダウンロードする通信」の有無によるものであると考えられる。これを踏まえて TPR の比較より、先行研究で「悪性ファイルをダウンロードする通信」の部分で検出できるものが、先行研究(-D)では検出できないことがわかる。本手法と先行研究(-D)の TPR を比較すると、本手法の方が高い。つまり本手法は、先行研究が悪性ファイルをダウンロードする段階で検知していた部分も検知することができている事がわかる。

実験 2, 実験 3 より特徴量 Phpver を減らした方が、Accuracy は高くなる。しかし、TPR は減少してしまうので、本研究の目的である、悪性ファイルをダウンロードするデータを含まな

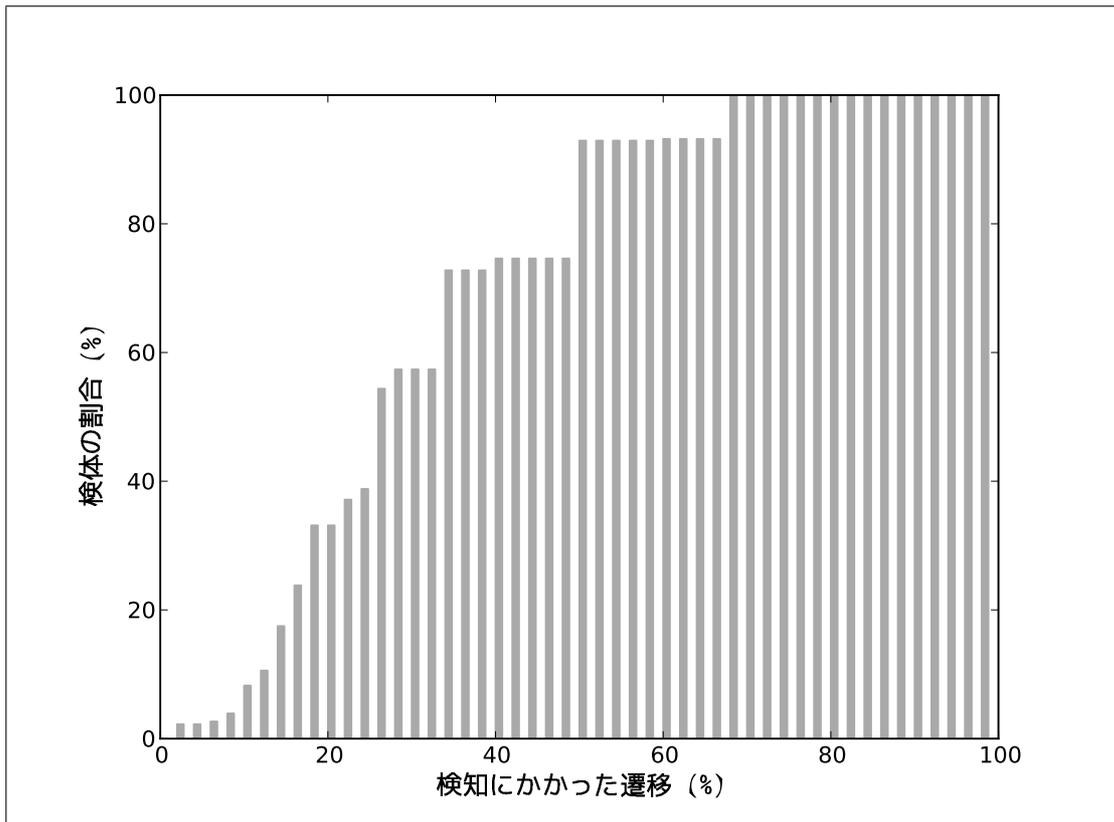


図 6.2: 検知できた遷移数と検体の割合 (先行研究)

い通信データのみで検知する，という観点からは良い結果とは言えない．加えて実験 3 より， $(\text{検知できた遷移数}) / (\text{実行ファイルをダウンロードするまでの遷移数})$  の平均が特徴量調整前に劣っていることから，特徴量を減らす必要はなく，本手法で提案した 7 つの特徴すべてが有用だと考えられる．

実験 3 より，本手法が早い段階での検知に有効である事がわかる．図 6.2，図 6.3 より，本手法の方が検知を低遷移で行っている．さらに上記 2 図における検体の割合の増加量を比較する．最大値 (最も検知した遷移) に着目すると，本手法は 24% の部分なのに対し，先行研究は 50% となっている．

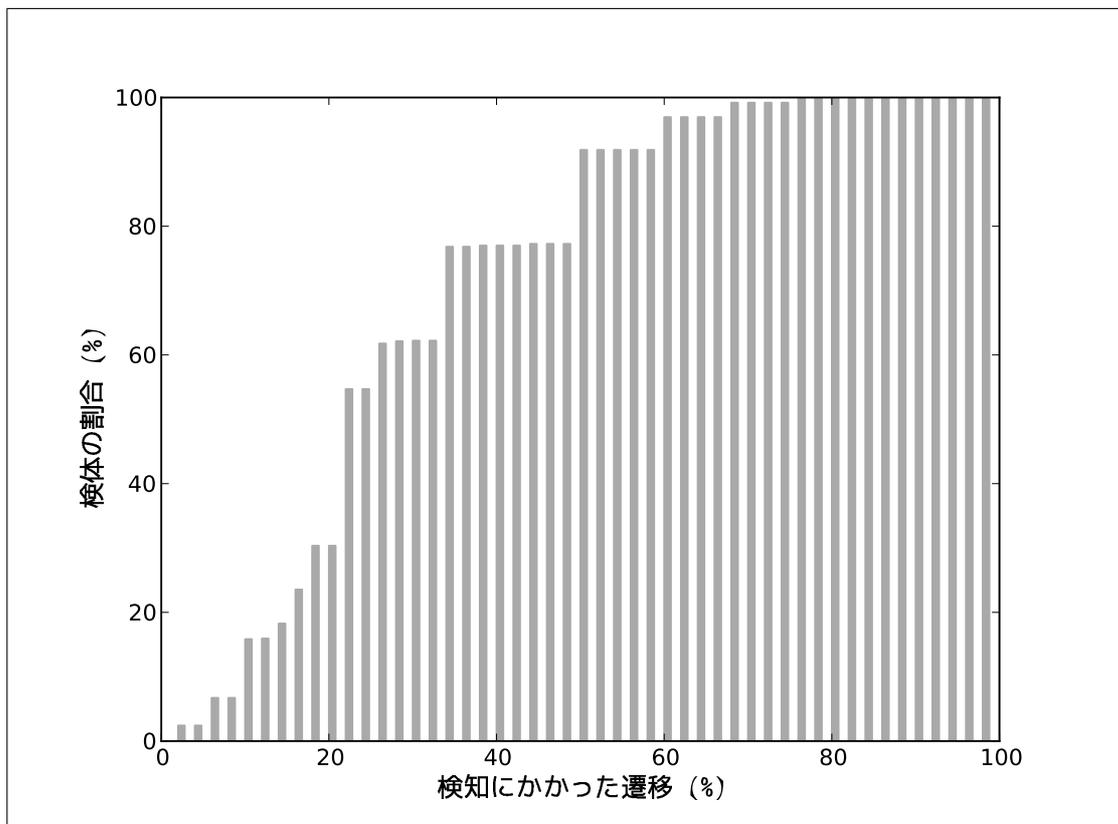


図 6.3: 検知できた遷移数と検体の割合 (本手法)

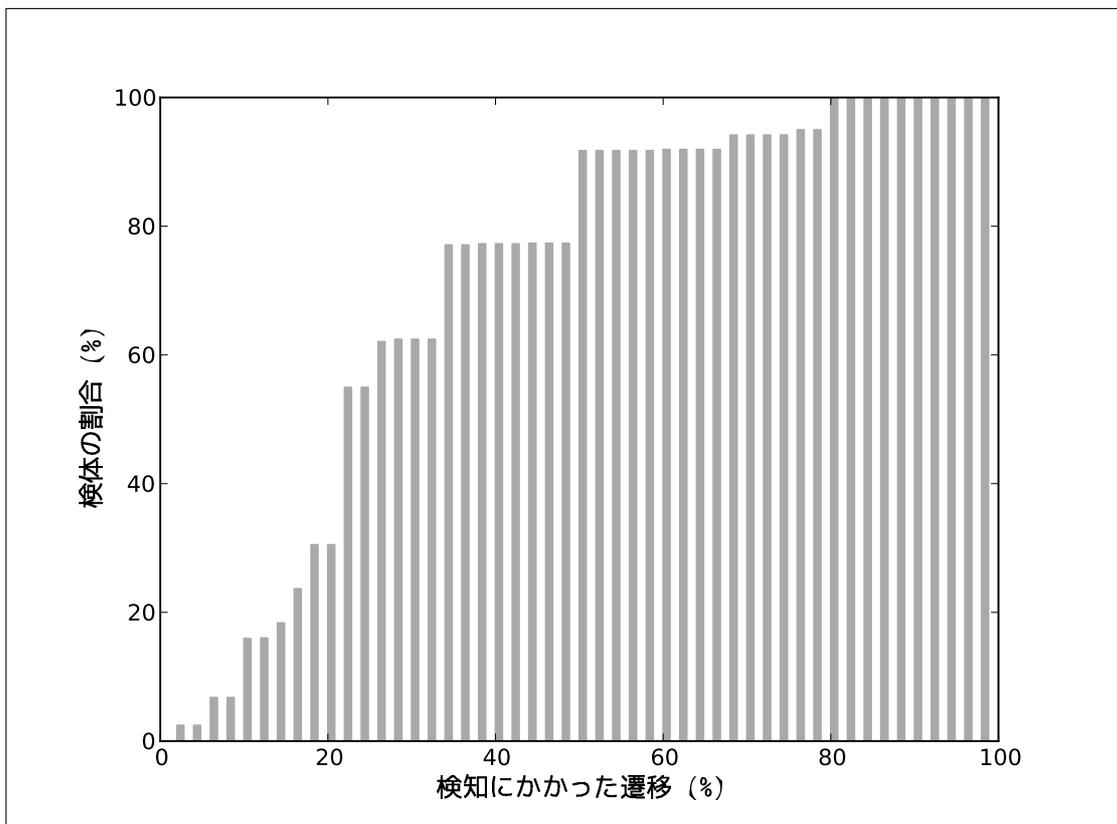


図 6.4: 検知できた遷移数と検体の割合 (特徴量調整後)

# 第 7 章

## 結論

### 7.1 まとめ

本研究は，HTTP の要求とその応答をペアとし，その URL の変化に注目した．これにより正常な通信と，Web 感染型マルウェアの自動的なダウンロードを識別する手法を提案した．実験の結果，検出率の向上を実現し，Web 感染型マルウェアを低遷移で検出可能であることを示した．また，マルウェアをダウンロードする途中の通信に難読化が施され，URL が読み取れない場合，従来の方法では次のペアが指す URL がないものと判断される．本研究ではアクセス遷移がない場合となるので悪性と判定される指標となり，難読化が施されていても本手法を適用可能である．さらに本手法はマルウェア本体の解析を行わない．このことにより，マルウェア本体が難読化されているもの，未知の動作をするものであっても，本手法を適用可能である．よって静的解析や，ヒューリスティック法では困難なマルウェアも検知可能である．さらに，検知に用いる通信データとしては，悪性ファイルをダウンロードする前の部分のみ使用する．そのことにより，先行研究よりも高精度な検知が可能となる．

### 7.2 今後の課題

#### 7.2.1 次のペア推定

本研究では，次のペアが指す URL がない場合，第 5.2.3 節で述べたようにホストに基づく推定を行った．しかし正確な遷移先が遷移元とは違うホストの場合がある．よって，次のペアの推定をより正確に行うことで，検知率の向上が期待できる．

### 7.2.2 特徴量効率化の手法

本手法では，第 5.6.2 節で述べたように，MalGrade (それまでのペアの累積悪性度) について，決定木の結果から使用する特徴量を選んだが，教師データに左右される可能性がある．教師データに応じて動的に決定する様な方法を取れば，検知率の向上が期待できる．

# 謝辞

本修士論文の作成にあたり，日ごろよりご指導を頂いた早稲田大学基幹理工学研究科の後藤滋樹教授に深く感謝いたします．また，本研究を進めるにあたり，後藤研究室の皆様には様々なアドバイスとご協力をいただきました．重ねて感謝いたします．

## 参考文献

- [1] IBM Security Service , “2015 年 上半期 Tokyo SOC 情報分析レポート” , IBM , [https://www-304.ibm.com/connections/blogs/tokyo-soc/resource/PDF/tokyo\\_soc\\_report2015\\_h1.pdf?lang=ja](https://www-304.ibm.com/connections/blogs/tokyo-soc/resource/PDF/tokyo_soc_report2015_h1.pdf?lang=ja) , September , 2015 .
- [2] JPCERT , “Web サイト改ざんに関する注意喚起” , JPCERT , <https://www.jpccert.or.jp/at/2013/at130027.html> , June , 2013 .
- [3] Independent Tests of Anti-Virus Software , <http://www.av-comparatives.org/>
- [4] Kazuhiro Furuhashi , “userAgent ( ユーザーエージェント一覧 )” , OpenSpace , <http://www.openspc2.org/userAgent/> , October , 2012 .
- [5] McAfee , “マカフィー、6月のサイバー脅威の状況を発表” , McAfee , <http://www.mcafee.com/japan/security/monthly/PC201305.asp> , June , 2013 .
- [6] 吉澤亨史 , “不正攻撃サイト報告の 14.3 % は短縮 URL、ドライブバイダウンロード攻撃やまず” , CNET Japan , <http://japan.cnet.com/news/business/20426859/> , March , 2011 .
- [7] Nick Johnston , “Blackhole Exploit Kit Gets an Upgrade: Pseudo-random Domains” , Symantec , <http://www.symantec.com/connect/blogs/blackhole-exploit-kit-gets-upgrade-pseudo-random-domains> , June , 2012 .
- [8] Charlie Osborne , “Blackhole malware toolkit creator 'Paunch' suspect arrested” , CBS Interactive. , <http://www.zdnet.com/blackhole-malware-toolkit-creator-paunch-arrested-7000021740/> , October , 2013 .

- [9] SophosLabs , “セキュリティ脅威レポート 2014” , Sophos , <http://www.sophos.com/ja-jp/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf> , January , 2014 .
- [10] /packer/ ,  
<http://dean.edwards.name/packer/>
- [11] 寺田 剛陽 , 古川 秀忠 , 東角 芳樹 , 鳥居 悟 , 検知を目指した不正リダイレクトの分析 , 情報処理学会 コンピュータセキュリティシンポジウム 2010 3F1 , pp.765–770 , October 2010.
- [12] 安藤 慎悟 , 寺田 真敏 , 菊池 浩明 , 通信の遷移に着目した不正リダイレクトの検出による悪性 Web サイト検知システムの提案 , 電子情報通信学会技術研究報告 2011 pp.205–210 , July 2011 .
- [13] Hesham Mekky , Ruben Torres , Zhi-Li Zhang , Sabyasachi Saha , Antonio Nucci , “Detecting malicious HTTP redirections using trees of user browsing activity” , INFOCOM 2014 , pp.1159–1167 , 2014 .
- [14] 小崎頌太 , 後藤滋樹 , HTTP 通信の遷移に基づく Web 感染型マルウェア検出法 , 電子情報通信学会総合大会 , p.180 , March 2014 .
- [15] マルウェア対策研究人材育成ワークショップ 2015 (MWS2015) ,  
<http://www.iwsec.org/mws/2015/>
- [16] 酒井 祐亮 , 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 , 情報処理学会報告 , pp.1–6 , March , 2013 .
- [17] Yuta Takata , Shigeki Goto and Tatsuya Mori , Analysis of Redirection Caused by Web-based Malware , Proceedings of the Asia-Pacific Advanced Network 2011 , v.32 , pp.53-62 , August , 2011 .
- [18] Mitsuaki Akiyama, et al: Design and Implementation of High Interaction Client Honeypot for Drive-by-download Attacks, IEICE Transactions on Communication, Vol.E93-B No.5 pp.1131–1139, May, 2010.

- [19] 永井 信弘, 千葉 大紀, 後藤 滋樹, HTTP 通信の時間軸解析による Web 感染型マルウェア検知, 情報処理学会全国大会講演論文集 2013(1), pp.549-551, March, 2013.
- [20] マルウェア対策研究人材育成ワークショップ 2011 (MWS2011),  
<http://www.iwsec.org/mws/2011/>
- [21] マルウェア対策研究人材育成ワークショップ 2012 (MWS2012),  
<http://www.iwsec.org/mws/2012/>
- [22] MWS2013 実行委員会, 研究用データセット MWS2013 Datasets について,  
<http://www.iwsec.org/mws/2013/about.html>
- [23] Microsoft, “Microsoft Security Intelligence Report Volume 18”, Microsoft Corporation, <https://www.microsoft.com/en-us/download/confirmation.aspx?id=46928>, December, 2014.
- [24] Mynavi Corporation, “ドライブ・バイ・ダウンロード攻撃が大幅増加 - IBM が上半期脅威レポート”, Mynavi Corporation, <http://news.mynavi.jp/articles/2015/09/08/ibm/>, September, 2015.
- [25] McAfee Labs, “McAfee Labs 脅威レポート:2015 年第 2 四半期”, McAfee, <http://www.mcafee.com/jp/resources/reports/rp-quarterly-threat-q2-2015.pdf>, August, 2015.
- [26] IBM, “セキュリティ脅威レポート IBM X-Force”, IBM, [http://www-01.ibm.com/software/jp/cmp/security\\_report/](http://www-01.ibm.com/software/jp/cmp/security_report/), September, 2015.
- [27] Mynavi Corporation, “マカフィー、ドライブバイダウンロード攻撃に関する脅威が多数、IE や JRE の脆弱性の解消を - マカフィーレポート”, Mynavi Corporation, <http://news.mynavi.jp/articles/2013/10/11/mcafee9/>, October, 2013.
- [28] Symantec Corporation, “シマンテックインテリジェンスレポート: 2013 年 1 月”, Symantec, [http://www.symantec.com/content/ja/jp/enterprise/white\\_papers/sr\\_wp\\_spam\\_report\\_1301.pdf](http://www.symantec.com/content/ja/jp/enterprise/white_papers/sr_wp_spam_report_1301.pdf), February, 2013.

- 
- [29] 川島 弘之, “Web 改ざんの次の段階、ドライブ・バイ・ダウンロード攻撃が約 4 倍に ~ IBM が警鐘”, Impress Watch Corporation, an Impress Group company, [http://cloud.watch.impress.co.jp/docs/news/20130826\\_612630.html](http://cloud.watch.impress.co.jp/docs/news/20130826_612630.html), August, 2013.
- [30] TCPDUMP & LIBPCAP,  
<http://www.tcpdump.org/>
- [31] Wireshark, tshark,  
<http://www.wireshark.org/>