HTTP通信ログ解析を用いた不正プログラム感染PC検知の試み

帯刀 直人[1]　鳩野 逸生[2,a)]

概要：本研究では, 神戸大学をはじめ多くの組織で保管している HTTP 通信のログを活用し, ログを分析することにより不正なプログラムに感染している PC を発見することを試みる. HTTP 通信ログには, 時刻, 送信元および宛先 IP アドレス, URL, Referer, User-Agent などの情報が含まれている. これらの情報を用いてまず通常の WWW ブラウザによる通信とそれ以外の通信を分離する. さらに疑わしい通信や疑わしい宛先ドメインを抽出することで, 不正プログラムに感染している可能性のあるネットワーク内部のコンピュータを抽出する. さらに, 本研究で提案するログ分析手法の有効性を評価するために, 神戸大学の学外向け HTTP 通信ログの分析を行い, 不正プログラム感染の可能性があるコンピュータを抽出することを試みる.

Approach to Detection of PCs Infected with Malicious Programs by using HTTP LOGs

Naoto Tatewaki[1]　Itsuo Hatono[2,a)]

Abstract: This paper deals with approach to detection of PCs infected with malicious programs, such as computer virus by analyzing the HTTP LOG data. Recently, it is difficult to detect all viruses by anti-virus software and IDSs, because various types of viruses are frequently updated. To cope with the difficulties, we analyze the HTTP logs in order to identify the traffic generated by the malicious programs. Furthermore, by using the generated traffic, we try to detect the IP addresses of the malicious PCs. In order to evaluate the effectiveness of this log analysis method, we applied the method to the HTTP logs stored in Kobe University. In the experiment, we could obtain the list of IP addresses of PCs that are estimated to be infected with malicious program.

1. はじめに

近年コンピュータウイルスの巧妙化が進んでいる. 以前のウイルスは, 大規模な DDOS 攻撃, UCE(Unsolicited Commercial Email) の大量送付など目に見える形で活動するものが多かったが, 最近は感染してもそのことに気が付かないようなウイルスが増加している. これは, 攻撃者の目的が, 個人情報や機密情報を盗み取ったり, 犯罪行為の踏み台として遠隔操作したりすることへと変化しているためである. 近年発生しているウイルスの多くは, 目立った活動は行なわず, アンチウイルスソフトにも検知されにくくするような工夫が凝らされていることが知られている [1].

また, 日々新しいウイルスが出現しているため対策が追いつかず, 従来の IDS(Intrusion Detection System) やアンチウイルスソフトの導入では感染を完全に防ぐことが困難となっている [2]. SandBox アプローチを用いたセキュリティアプライアンスなど新しい技術を用いた機器が開発され, 様々なベンダーから提供されつつあるが, すべての攻撃を発見できる訳ではない. このため, 企業や大学などのネットワーク管理者は, ウイルスの感染を完全に防御することは事実上不可能であるということを前提に対策を講じる必要であるとされている.

[1] 神戸大学 大学院システム情報学研究科
[2] 神戸大学 情報基盤センター
Information Science and Technology Center, 1-1 Rokko-dai, Nada, Kobe 657-8501 Japan
[a)] hatono@kobe-u.ac.jp

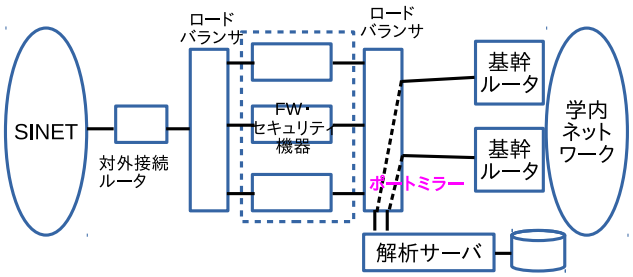

図 1 HTTP 通信取得の取得位置

上のような問題に対処するため, DNS への問い合わせのトラフィックデータを用いて, ネットワーク内部の未知のボット感染 PC を検知する手法 [3] や, ファイアウォールのログを用いて不正プログラムの活動を検出する例 [4], 複数のボットに共通する通信の特徴をモニタすることによる検知手法 [5], [6] など様々な研究が提案されている. しかし, 現在, 神戸大学においては, 対外接続の帯域に対するファイアウォールなどの通信機器の性能の制約から, 詳細なログを定常的に出力することや, 通信をリアルタイムにモニタすることが困難であるため, 神戸大学内で適用することは困難である. このような情況の下で, 本研究では, 神戸大学内で従来から蓄積されている HTTP 通信のログを用いて, 実際にウイルス等に感染してしまった PC(端末) が行う HTTP 通信を分析することにより, 事後検知を行うことを試みる.

HTTP は比較的厳しいポリシーをファイアウォールに設定している組織においても利用できる可能性が高いプロトコルであるため, 不正なプログラムが自らの活動のために何らかの形で HTTP 通信を利用している可能性は十分にあると考えられる. また, HTTP 通信ログには, 時刻, 送信元および宛先 IP アドレス, URL, Referer, User-Agent などの情報が含まれているため, ある程度 HTTP 通信を利用しているアプリケーションに関する情報を推定することができる. 本研究では, これらの情報を用いてまず通常の WWW ブラウザ (以下ブラウザという) による通信とそれ以外の通信を分離し, さらに疑わしい通信や疑わしい宛先ドメインを抽出することで, 不正プログラムに感染している可能性のあるネットワーク内部のコンピュータを抽出する手法の提案を行う.

さらに, 本研究で提案する手法の評価を行うため, 神戸大学において保存されている HTTP ログデータを提案手法で分析するすることにより, 不正プログラム感染の可能性があるコンピュータを抽出することを試みる.

2. HTTP 通信情報の取得 [7]

神戸大学においては, インシデント発生時の調査や不正利用の監査を主な目的として, 学内から学外への HTTP 通信のログ情報を取得し, 保存している. これは, 神戸大学情報セキュリティポリシーにおいて, 基幹ネットワーク管理者は, 学内のセキュリティ状況を把握してセキュリティ維持のための施策をとる必要がある, という規定を根拠に実施しているものである.

図 1 に, 神戸大学における HTTP 通信取得の状況を示す. 神戸大学では, 学内からの通信がファイアウォールを通る直前に設置しているロードバランサのポートをミラーし, それらのポートを解析サーバから通信モニタリングソフトウェア tshark[8] を用いることにより HTTP 通信ログを取得している (図 1). ロードバランサは, 学内からの通信を, HTTP 通信とそれ以外に分離し, それぞれ Web プロキシおよびファイアウォールへ送るために利用している. Web プロキシは複数台存在し, ロードバランサによって負荷分散されている. 基幹ルータからロードバランサは 10Gbps で接続されているが, ミラーポートは 1Gbps であることと, 解析サーバの性能を考慮するとかなりのパケットを取りこぼしていることが予想される. しかし, 100%すべてのパケットを取得するためにはかなりのコストがかかるため本構成としている.

図 1 に対して, tshark のプロトコル解析機能を用いて, 時刻, ソース IP, 相手先 IP, HTTP request method, ホスト名, URI, HTTP request version, Content Length, Referer, User_Agent, ソースポート, 相手先ポートを取得して, apache HTTP server における combined access_log フォーマット [9] に近い形に整形してファイルに出力している. 本通信の取得においては, 通信パケットをモニタして取得している関係上, HTTP リクエスト と サーバレスポンスは, 異なったパケットとして観測されるが, HTTP リクエストとサーバレスポンスに対する関連付け処理は行っていない [*1].

3. HTTP ログ情報の解析

HTTP ログ情報の解析により不正プログラムが埋め込まれたと疑われる PC を発見するにあたって以下のことを仮定する.

(1) 検査するネットワークにおいて, 1 つの IP アドレスにつき 1 台の PC のみが接続され, かつ固定されている.
(2) ブラウザによる HTTP リクエストにつけられる User-Agent には, ブラウザの種類やバージョン, OS の種類などが入ったデフォルトのものが使われている.
(3) 検知対象とする不正プログラムは, PC に感染した後定期的に外部と通信を行う.

(1) については, ログ分析の際に PC ごとにログを分けて分析を行う必要があると考えられるからである. HTTP 通信ログ情報では, IP アドレスでしか送信元を特定できな

[*1] すべての HTTP パケットが収集できていない可能性が高く, すべてのパケットに関して関連付け処理を行うことが困難であると思われるためである.

い. NAT ルータが配下に接続されている IP アドレスからの通信であると, 複数の PC からの通信が混ざって分離することが難しいと考えられる. また, 無線 LAN 接続環境等で IP アドレスが動的に変わる場合も複数の PC からの通信が混ざっていると考えられる. この場合認証情報等から各 HTTP 通信がどのユーザからのものであったか分類する必要があるが, 本研究では対象外としている.

(2) については, ブラウザによる通信を区別するために User-Agent の情報を利用するからである. 1 つのブラウザによる通信であっても, 同じブラウザを利用中に途中で User-Agent が変わると, 同じブラウザによる通信であると認識することが難しいためこのような仮定をおいている [*2].

(3) については, 不正プログラムは, 外部からの動作指令を受けるために, 定常的に外部と通信を行っていると考えられるためである.

以上のような仮定の元で, 以下に示す手順で不正プログラムが動作していると思われる PC の同定を行う.

(1) 各 IP(PC) において通常使われる「ブラウザによる通信」の特定.
(2) 不正プログラムによる通信が満たすと推定される特徴に基づいた該当ドメインの絞り込み
(3) 各 PC 毎のアクセス傾向分析による不正プログラム感染疑い PC の抽出

なお, ホワイトリストを用意して, 明らかに正規の通信だと考えられる宛先ドメインは検知対象からはずしている. これは, ログが大量に存在することから, 正規の通信は初めの段階でできるだけ検知対象から外すことで処理時間の短縮を図るためである. 以下に, 各手順の詳細を述べる.

4. ブラウザ以外の通信の特定

本研究では, ブラウザを含むソフトウェアの区別に User-Agent を用いる. 同じ User-Agent を持つログは, 同じソフトウェアが通信を行っているものとする. 各 User-Agent がブラウザであるかどうか判断するために, **図 2** (a)–(d) に示す正規表現にマッチするものを選ぶ [*3].

しかし, User-Agent のパターンマッチングだけでは, User-Agent をブラウザに見せかけたアプリケーションの通信ををブラウザによるものと判断する可能性があるので, さらに User-Agent ごとの “Referer が存在するログの行数” に注目する. **図 3** に, ある PC の 1 日における User-Agent ごとのログの行数である. 括弧外は全行数で, 括弧内はその内 Referer が存在するログの行数を示す. この例を含め, 多くの場合 Referer が存在する割合が高い User-Agent と,

```
^Mozilla/4\.0 \(compatible; MSIE 7\.0; Windows NT [56]\.[0123];
^Mozilla/4\.0 \(compatible; MSIE 8\.0; Windows NT [56]\.[0123];
^Mozilla/5\.0 \(compatible; MSIE 9\.0; Windows NT 6\.[01];.+Trident/.+\)$
^Mozilla/5\.0 \(compatible; MSIE 10\.0; Windows NT 6\.[12];.+Trident/.+\)$
^Mozilla/5\.0 \(Windows NT 6\.[13];.+Trident/7\.0;.+rv:11\.0.*\) like Gecko
```

(a) User-Agent of Internet Explorer

```
^Mozilla/5\.0 \(Windows NT [56]\.[0123]\) AppleWebKit/537\.36 \(KHTML, like Gecko\) Chrome/.+ Safari/.+
^Mozilla/5\.0 \(Windows NT [56]\.[0123]; WOW64\) AppleWebKit/537\.36 \(KHTML, like Gecko\) Chrome/.+ Safari/.+
^Mozilla/5\.0 \(Macintosh;.+Chrome/.+ Safari/.+
```

(b) User-Agent of Chrome

```
^Mozilla/5\.0 \(Windows NT [56]\.[0123]; rv:.+\) Gecko/20100101 Firefox/.+
^Mozilla/5\.0 \(Windows NT [56]\.[0123]; WOW64; rv:.+\) Gecko/20100101 Firefox/.+
^Mozilla/5\.0 \(Macintosh;.+Firefox/.+
```

(c) User-Agent of Firefox

```
^Mozilla/5\.0 \(Windows.+Version/.+ Safari/.+
^Mozilla/5\.0 \(Macintosh;.+Version/.+ Safari/.+
```

(d) User-Agent of Safari

図 2 User-Agent チェックパターン

```
14121(13900) Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36
 1169(1096) Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0
      667(0) Microsoft NCSI
       73(0) Microsoft-CryptoAPI/6.1
       22(0) Microsoft BITS/7.5
       11(0) Windows-Update-Agent
        2(0) Mozilla/4.0 (compatible; Win32)
```

図 3 ある IP におけるログ中の User-Agent 行数

Referer が存在する割合が 0 かそれに近い User-Agent に分かれることが多い [*4]. そのため, 前者がブラウザで, 後者がそれ以外のソフトウェアの可能性が高いと考えられる. したがって, “Referer が存在するログの行数”（図 3 では括弧内の数字）が一番多い User-Agent をメインで使うブラウザとするのが妥当であると思われる. ただし, 複数のブラウザを普段使っている可能性が考えられるので, Internet Explorer, Chrome, Firefox, Safari のそれぞれにおいて “Referer が存在するログの行数” が一番多い User-Agent を「通常使うブラウザ」とする. 例えば図 3 では, 枠で囲った Chrome と Firefox の 2 つの User-Agent が「通常使うブラウザ」として選ばれる.

ブラウザのバージョンアップなどで User-Agent の文字列が変更されることがあるので, この「通常使うブラウザ」の判断は, 1 日ごとに行う. また, 以下のルールも適用する.

- 更新の頻度が高い Chrome, Firefox, Safari は 1 日の中で 2 つのバージョンが利用される場合も多いと推測されるので, “Referer が存在するログの行数” が 2 番目に多い User-Agent までを「通常使うブラウザ」に加える.
- 上記の条件を満たしていても, Referer が存在する割合が 6 割未満の User-Agent は「通常使うブラウザ」には加えない.
- ブラウザを使わなかった日など, 「通常使うブラウザ」の条件を満たす User-Agent が存在しない場合も考えられるので, その日は「通常使うブラウザ」による通信は無かったものとする.

[*2] 通常使用されているブラウザにおいては, User-Agent を変更できるものも存在するが, 今回は対象としない.
[*3] 本研究では, ブラウザの種類が Internet Explorer, Chrome, Firefox, Safari, OS の種類が Windows, Mac OS のものにマッチするようにしているが, ネットワーク内の環境によっては, ブラウザ・OS の種類を追加したほうがよいと考えられる
[*4] HTTP ログの事前分析による.

5. 該当ドメインの絞り込み

5.1 部分ドメインの抽出

前節で述べた手順により抽出したログの中から, さらにドメインの部分ドメインを抽出する. ただし, 本研究における部分ドメインとは, URL において, FQDN を指定している場合, その FQDN の第 1・第 2 レベルドメインまでを取り出したものを指す. 第 2 レベルドメインが ac, co, go, or, com, ad, ne, gr, ed, lg, gov, edu のいずれかである FQDN に限り, 第 3 レベルドメインまでを取り出す. URL において, IP アドレスを指定している場合は, その IP アドレスの第 1・第 2 オクテットを取り出したものとする. 部分ドメインを用いるのは, 下位のレベルのドメインだけを変えて上位は同じドメインの FQDN が利用される場合が多いということが HTTP 通信ログの事前の分析により判明しているからである.

この手順のみ, 対象の部分ドメインのアクセスの傾向をはかるために, 前手順で抽出したログを, ネットワーク内のすべてのソース IP アドレスのものをまとめて分析を行う. なぜなら, 例えば正規の通信の宛先であっても, ある人は頻繁にアクセスして, 別の人はほとんどアクセスしないといった通信傾向の違いがある可能性があるが, ネットワーク全体で見ると, 正規の通信先の通信傾向, 不正な通信先の通信傾向がそれぞれ似たようなものになると考えられるからである.

5.1.1 分析対象期間の途中から出現する部分ドメインの抽出

本手順では, 分析対象期間の途中から出現し, そこから毎日に近い形で定期的に通信のある宛先部分ドメインの抽出を行う. これは何らかの不正プログラムがネットワーク内の PC に入り込んで, 外部と頻繁に通信を行っていることが考えられるからである.

ネットワーク内すべてのソース IP アドレスに対する前手順後の抽出ログをまとめ, すべての宛先部分ドメインについて, 分析対象期間中の出現日数 D_A と, 初めて出現した日から分析対象期間の最終日までの日数 D_B を調べ, 初めて出現してからの出現頻度

$$P = D_A/D_B$$

を求める. P が一定頻度以上, D_B が分析対象日数より一定数小さい値以下の宛先部分ドメインを抽出する. それらは, 分析対象期間の途中から出現し, そこから定期的にアクセスする宛先部分ドメインである. D_B が分析対象日数に近い部分ドメインは, 分析対象期間の初めのほうから出現していて, それらには分析対象期間の前からもずっとアクセスがあった宛先が多く含まれると考えられる. そのような宛先には, 正規のものも多くあることが予想されるので, D_B が分析対象日数より一定の小さい値以下の部分ドメインに限定している.

また, D_A,D_B の日数がともに小さいものは, 普段はあまりアクセスがないが, たまたまアクセスがあった宛先と考えられるので, D_A,D_B が一定数以下の部分ドメインは除く.

5.1.2 URL チェックサイトによる検査

さらに, **5.1.1 節**で抽出した宛先部分ドメインを含む URL を, URL チェックサイト Virustotal[10] で調べて, 悪性であると検知した部分ドメインを抽出する. これらの宛先部分ドメインは不正な通信先の可能性が高いと考えられる.

6. 評価実験

本節では, 提案手法を評価するため, 神戸大学の学内ネットワークの HTTP 通信ログを用いて行った実験について述べる. 実験に際しては, 学内ネットワークのソース IP アドレスの一部をハッシュ化により暗号化したデータを用い, プライバシーに配慮している. また, 実験対象の IP アドレスレンジにおいては, ほとんど Windows PC のみが利用されていることが分かっている.

6.1 実験概要

本実験で用いる HTTP 通信ログの概要は以下のとおりである.

- 神戸大学学内で取得した HTTP 通信ログで, 1 つのソース IP アドレスにつき 1 台の PC のみの接続が確認されているアドレス範囲のログ
- 本アドレス範囲では, 学外への通信は HTTP, HTTPS, FTP のみが許可されている.
- 2013 年 11 月 1 日から 2014 年 10 月 31 日までの 1 年間のうち, 土日祝日や休暇日を除く全 240 日分 [*5]
- 全部で約 843,200,000 行あり, 1 日あたりの平均が約 3,513,000 行
- 1 日あたり平均 983 個のソース IP アドレスが存在

この HTTP 通信ログに対して, 前章で述べた分析方法を適用し, どれだけのソース IP アドレスが不正プログラム感染の可能性があると抽出されるか実験を行う. 実験結果では, 各手順ごとの分析結果を示してから, 最終的な抽出結果を示す.

抽出結果の評価については, 最終的な抽出結果における各ソース IP アドレスに対して, ログをより詳細に調査し, 疑わしい宛先ドメインがはじめて出現した日に, 疑わしい実行形式ファイルのダウンロードがされたかを確認することにより行う.

6.2 実験結果

まずはじめに, 240 日のそれぞれの日において, 各ソース

[*5] 該当 IP レンジを利用している部署においては, 土日祝日において出勤することはまれであり, PC はほとんど利用されていないことが分かっているためである.

Source IP Address	User-Agent
10.2.04	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.04	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
10.2.04	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
10.2.04	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.07	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; YTB730; GTB7.5; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729;
10.2.07	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; MAFSJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; MAFSJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36
	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; Trident/7.0; MAARJS; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; rv:33.0) Gecko/20100101 Firefox/33.0
	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
10.2.07	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
10.2.07	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
10.2.07	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.143 Safari/537.36 Sleipnir/6.1.0
	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.104 Safari/537.36 Sleipnir/6.1.1
	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
⋮	⋮

図 4 同定された通常使われるブラウザ

表 1 各部分ドメインの D_A,D_B と P の値

D_A	D_B	P	Partial Domain
107	240	0.4	bab○○○○○
59	169	0.3	23.○○○○○
239	240	0.9	fenr○○○○○
211	240	0.9	221.○○○○○
159	239	0.7	202.○○○○○
115	240	0.5	biji○○○○○
240	240	1	digi○○○○○
240	240	1	ceip○○○○○
118	240	0.5	211.○○○○○
232	240	0.9	asah○○○○○
19	137	0.1	117.○○○○○
197	232	0.8	209.○○○○○
81	240	0.3	111.○○○○○
63	240	0.3	117.○○○○○
240	240	1	adv○○○○○
95	240	0.4	124.○○○○○
240	240	1	cube-○○○○○
⋮	⋮	⋮	⋮

IP アドレスごとに「通常使うブラウザ」の特定を行った. **図 4** に, 2014 年 10 月 31 日における各ソース IP アドレスごとの「通常使うブラウザ」のうち, 一部のソース IP のものについて示す.

次に, この「通常使うブラウザ」以外の通信から, 疑わしい通信の抽出 (5 節) を実施した. 各ソース IP アドレスごとの抽出結果をまとめて, 含まれる部分ドメインごとに, 240 日中の出現日数 D_A と, 初めて出現した日から最終日の 2014 年 10 月 31 日までの日数 D_B を調べて, 初めて出現してからの出現頻度 P を求めた. 結果の一部を**表 1** に示す（なお, P は, 小数点下 2 桁目を四捨五入した値になっている. また, ドメインの一部は伏せてある）.

この結果には 7832 個の部分ドメインが含まれていることがわかった.

次に, 分析対象期間の途中から出現し, そこから毎日に近い形で定期的に通信のある部分ドメインの抽出 (5.1.1 節) を実施した. 先ほどの結果の中から, $P \geq 0.5$, $D_B \leq 230$ を満たす部分ドメインを抽出した. また, 出現日数が少ない（$D_A, D_B \leq 5$ となる）部分ドメインは除いた. その結果 128 個の部分ドメインが残った.

表 2 virustotal における検出エンジン数ごとの部分ドメインの数

No. of Virustotal Detection	Number of Partial Domains
8	3
7	1
6	2
5	6
4	4
3	11
2	13
1	21
0	67

さらに, この 128 個の部分ドメインに関して, そのドメインを含むログ中の URL を URL チェックサイト Virustotal で検査した. 2015 年 2 月現在で約 60 のウイルス対策エンジンによるチェックができ, 検査の結果, 1 つ以上のエンジンで悪性と判定された部分ドメインが 61 個存在した. **表 2** に, Virustotal における検出エンジン数ごとの部分ドメインの数を示す.

次に, 悪性判定の部分ドメイン 61 個のみについて, 各ソース IP アドレスごとに, 初めて出現してからの出現頻度 P と, 1 日平均ログ数 N を求めた (小数点下 2 桁目を四捨五入). $P \geq 0.8$, $N \geq 10$ となる部分ドメイン, すなわち初めて出現してから最終日まで, 8 割以上の日数で出現していて, 出現した日は 1 日当たり 10 回以上の通信をしている部分ドメインがあるソース IP アドレスを, 最終的に不正プログラム感染の可能性があるものとして抽出した [*6]. 最終

[*6] 本 IP レンジを利用している部署においては基本的に毎日出勤して PC を利用することが分かっている. 休暇, 出張等があることも考慮して 8 割と設定している. また, 1 日 10 回という数字は, 予備的な解析において 10 回より少ない場合候補が多くなりすぎ, この後の処理を行うことが困難であったためである.

Source IP Address	Partial Domain	First Appearance Date	P	N
10.2.fb[redacted]i33c4.△	sizlseOOOOO	2014/4/15	0.9	288.1
	databOOOOO	2014/5/14	0.9	15
	reimaOOOOO	2014/8/27	0.9	14.2
	infosOOOOO	2014/9/19	0.8	15.5
10.2.fb[redacted]i33c4.△	databOOOOO	2014/5/16	0.8	10.3
10.2.fb[redacted]i33c4.△	greeneOOOOO	2014/6/13	0.9	234.7
10.2.d[redacted]2642.△	databOOOOO	2014/5/21	0.9	11.1
	advanOOOOO	2014/9/26	0.9	335.9
10.2.92[redacted]fad10.△	yulOOOOO	2014/6/9	0.9	245.5
	turbOOOOO	2014/6/9	0.9	28.6
	databOOOOO	2014/6/12	0.9	19.3
	undeOOOOO	2014/10/20	1	58.6
10.2.92[redacted]fad10.△	data-OOOOO	2014/10/3	1	62.5
10.2.86[redacted]bb2a.△	greeneOOOOO	2014/6/20	0.9	261.6
10.2.7[redacted]6f215.△	mybuOOOOO	2013/12/2	0.9	170.9
	infosOOOOO	2014/6/26	0.8	14.8
10.2.7[redacted]6f215.△	statiOOOOO	2014/5/8	0.9	21.7
10.2.7[redacted]6f215.△	mightyOOOOO	2014/7/28	0.9	342.8
	infogensOOOOO	2014/7/30	0.9	16.7
	perfoOOOOO	2014/8/19	0.9	13.1
	data-OOOOO	2014/10/22	0.9	105.6
10.2.7[redacted]6f215.△	databOOOOO	2014/5/14	0.9	10.7
10.2.7[redacted]6f215.△	databOOOOO	2014/5/15	0.9	16.6
10.2.7[redacted]6f215.△	databOOOOO	2014/5/14	0.9	10
10.2.[redacted]9f3e.△	databOOOOO	2014/5/15	0.9	11.3
10.2.[redacted]9f3e.△	databOOOOO	2014/5/14	0.8	10.8
10.2.3[redacted]2189.△	neurowlOOOOO	2014/10/1	1	185.2
10.2.3[redacted]2189.△	databOOOOO	2014/5/14	0.9	10.6
10.2.3[redacted]2cb1.△	databOOOOO	2014/7/14	0.9	12.1
10.2.[redacted]555f.△	frameOOOOO	2014/10/3	1	331.8

図 5 HTTP ログの分析結果

Source IP Address	Partial Domain	First Appearance Date	Detetion Result
10.2.fbd[redacted]i633c4.△	sizlseOOOOO	2014/4/15	Detected
	databOOOOO	2014/5/14	Undetected
	reimaOOOOO	2014/8/27	Detected
	infosOOOOO	2014/9/19	Detected
10.2.fbd[redacted]i633c4.△	databOOOOO	2014/5/16	Undetected
10.2.fbd[redacted]i633c4.△	greeneOOOOO	2014/6/13	Detected
10.2.da[redacted]62642.△	databOOOOO	2014/5/21	Unknown
	advanOOOOO	2014/9/26	Detected
10.2.92c[redacted]fad10.△	yulOOOOO	2014/6/9	Detected
	turbOOOOO	2014/6/9	Detected
	databOOOOO	2014/6/12	Detected
	undeOOOOO	2014/10/20	Undetected
10.2.92c[redacted]fad10.△	data-OOOOO	2014/10/3	Detected
10.2.861[redacted]bb2a.△	greeneOOOOO	2014/6/20	Detected
10.2.7c[redacted]6f215.△	mybuOOOOO	2013/12/2	Detected
	infosOOOOO	2014/6/26	Undetected
10.2.7c[redacted]6f215.△	statiOOOOO	2014/5/8	Detected
10.2.7c[redacted]6f215.△	mightyOOOOO	2014/7/28	Undetected
	infogensOOOOO	2014/7/30	Detected
	perfoOOOOO	2014/8/19	Unknown
	data-OOOOO	2014/10/22	Detected
10.2.7c[redacted]6f215.△	databOOOOO	2014/5/14	Undetected
10.2.7c[redacted]6f215.△	databOOOOO	2014/5/15	Undetected
10.2.7c[redacted]6f215.△	databOOOOO	2014/5/14	Undetected
10.2.5f[redacted]f9f3e.△	databOOOOO	2014/5/15	Undetected
10.2.5f[redacted]f9f3e.△	databOOOOO	2014/5/14	Undetected
10.2.33[redacted]52189.△	neurowlOOOOO	2014/10/1	Detected
10.2.33[redacted]52189.△	databOOOOO	2014/5/14	Undetected
10.2.32[redacted]c2cb1.△	databOOOOO	2014/7/14	Undetected
10.2.1f[redacted]c655f.△	frameOOOOO	2014/10/3	Detected

図 6 不正プログラムのダウンロード実績による評価結果

的な抽出結果を**図 5** に示す．ソース IP アドレスごとに, 疑わしい宛先部分ドメインと, そのドメインの初出現日, 出現頻度 P, 1 日平均ログ数 N の値を示している．

ソース IP アドレスのみで数えると 19 個の不正プログラム感染の可能性がある IP アドレスが抽出された．（ソース IP アドレス, 部分ドメイン）の組で数えると 30 個が抽出された．

6.3 抽出結果の評価

本研究における通信ログ分析方法を評価するために, 以下の調査を行った．

(1) 抽出結果の（ソース IP アドレス, 部分ドメイン）の組ごとに, 部分ドメインの初出現日において, その部分ドメインに初めてアクセスがある以前に Windows 実行形式ファイル（exe ファイル）のダウンロードが行われているかログを調査する．

(2) exe ファイルのダウンロードが行われている場合, Virustotal においてダウンロードファイルの分析を行い, 悪性かどうかを調べる（悪性かどうかについては, 1 つ以上の検索エンジンで悪性と判断されれば「悪性」とし, そうでないものは「良性」とする）．ただし, 分析は, Virustototal に, ダウンロードの URL を与えることにより行っている．仮に exe ファイルが消去されていた場合でも, Virustotal では履歴が表示される場合が多いためである．

ファイル分析の結果, 悪性なプログラムであると検知された場合, 実際にそのソース IP アドレスの PC には不正なプログラムが入り込んだ可能性が高いといえる．

図 6 に, （ソース IP アドレス, 部分ドメイン）の組ごとの調査結果を示す．exe ファイルのダウンロードがない場合, あるいはすべての exe ファイルが良性であった場合は “Undetected”, 1 つ以上の exe ファイルが悪性であった場合は “Detected” としている．ただし, 原因は不明であるが, Virustotal で分析ができない場合があったので, 悪性の exe ファイルは無かったが, 分析できない exe ファイルがあった場合は “Unknown” としている．また, windowsupdate プログラムやセキュリティソフトによるものなど, ドメイン名から明らかに良性であると考えられるファイルは, はじめから良性と判断した．

表 3 不正プログラムダウンロードの集計

	Number of Source IP Addresses
Detected	11
Undetected	8
Total	19

図 6 における検査結果ごとのソース IP アドレスの数を**表 3** に示す．ただし, 複数の部分ドメインがあるソース IP アドレスについては, 1 つでも “Detected” があれば, 不正プログラム感染の可能性が高いとして “Detected” としている．

この調査により, HTTP 通信ログ分析の結果, 不正プログラム感染の可能性があるソース IP アドレス（PC）として抽出した 19 個のうち, 不正プログラム感染の可能性が高いものが 11 個含まれていることがわかった．Virustotal によるファイル分析の結果を確認したところ, はっきりと断定はできないが, これらの不正プログラムの多くは, アドウェアや有償のメンテナンスソフトの購入を促す迷惑ソフト等であると推測される．また, これらのプログラムはフリーソフト等をインストールする際に, 表示されたメッセージ等をよく読まずに進めてしまい, 一緒にインストールされてしまうものが多いと考えられる．

残りの 8 個については, 疑わしいドメインの初出現日に

悪性の exe ファイルのダウンロードは無かったが, 不正プログラムに感染している可能性がないとは言えない. 不正プログラムに感染しているとすれば次のようなことが考えられる.

- 疑わしいドメインの初出現日以前に感染し, 新たに通信先を追加した, あるいは通信先を変更した.
- exe ファイル以外のファイルのダウンロードによって感染した.
- HTTP 通信によるダウンロード以外の方法によって感染した. 例えば, 外部メディアの接続やメールの添付ファイルなどが考えられる.

不正プログラム感染の可能性をより正確に確かめるためには, 今回のログ分析で抽出した PC を 1 台ずつ直接調べる方法が考えられるが, 今回の結果だけではウイルスに感染したと断定できない以上, 神戸大学情報セキュリティポリシーに基づいて調査を命令することは困難であると情報セキュリティ担当者が判断しているため実施していない [*7].

7. おわりに

本研究では, HTTP 通信のログ情報を分析し, 不正プログラムが外部と行う通信を検知することによって, ネットワーク内部で動作する不正プログラムを見つけ出すことを目的とし, 不正プログラムが動作していると疑われる PC を特定するための手法について述べた.

さらに, 本研究で提案した手法用いて, 実際に学内 HTTP 通信のログを分析し, 不正プログラム感染の可能性がある端末の抽出を試みた. 抽出結果に対して, 不正プログラムのダウンロードの有無に関する検査を行った結果, 抽出された PC の多くで不正プログラムのダウンロードが行なわれており, 可能性が高い PC が含まれていることが分かった.

今後の課題として, ログ分析の前提としてあげた, 1 つの IP に 1 台の PC が接続されているという条件を満たさない場合の対応があげられる. また, 今回対象とした PC が接続されているネットワークでは, IPv4 のみが利用できる環境であるため IPv4 アドレス単位で処理を行うことで PC と対応させることができる. しかし, 神戸大学の他の部局では IPv4/IPv6 のデュアルスタック環境が利用できるため, IPv6 での通信も考慮する必要がある. さらに, Web の HTTPS 通信への移行が拡大してきているが, HTTPS 通信の場合, URL, User Agent などの情報を取得することが困難なため, 本研究で提案した手法を直接利用することはできない. そのため, HTTPS 通信をある程度解析できる次世代ファイアウォールなどと組み合わせた手法を考える必要があると思われる.

[*7] 該当の PC に不審な通信が行なわれていることは, 部署を通じて通知済みである.

参考文献

[1] 情報処理推進機構, “情報セキュリティ白書 2015,” 情報処理推進機構 (2015)
[2] Samuel Gibbs, “Antivirus software is dead, says security expert at Symantec,” *The Guardian*, http://www.theguardian.com/technology/2014/may/06/antivirus-software-fails-catch-attacks-security-expert-symantec (2014)
[3] 佐藤 一道, 石橋 圭介, 豊野 剛, 三宅 延久, “DNS トラフィックデータを利用したボット感染者検出方法,” 情報処理学会研究報告, Vol. 2009-IOT-7, No. 11, pp. 1–6(2009)
[4] 加藤 淳也, 門田 剛, 畑田 充弘, 竹内 文孝, “ファイアウォールログを利用したマルウェア活動の検出手法について,” 情報処理学会シンポジウム論文集, Vol. 2009, No. 11, pp. 259–264(2009)
[5] 水谷 正慶, 金井 瑛, 武田 圭史, 村井 純, “通信の共通性を利用した悪性プログラム検知手法の実装と評価,” 情報処理学会論文誌, Vol. 50, No. 9, pp. 2137–2146(2009)
[6] 水谷 正慶, 武田 圭史, 村井 純, “Web 感染型悪性プログラムの分析と検知手法の提案,” 電子情報通信学会論文誌. B, Vol. J92-B, No. 10, pp. 1631–1642(2009)
[7] 鳩野逸生, “HTTP 通信ログ解析による学内情報機器の利用状況推定,” インターネットと運用技術シンポジウム 2014 論文集 2014, pp. 63-70, (2014)
[8] Gerald Combs, at al.: wireshark. 入手先 ⟨https://www.wireshark.org/⟩ (2014)
[9] The Apache Software Foundation: apache ログファイル. 入手先 ⟨http://httpd.apache.org/docs/2.2/en/logs.html⟩ (2014)
[10] Virustotal, https://www.virustotal.com/