# プロキシのログからの機械学習による RAT の検知方式

三村 守 †　　大坪 雄平 †‡　　田中 英彦 †

† 情報セキュリティ大学院大学
221-0835 神奈川県横浜市神奈川区鶴屋町 2-14-1
dgs104101@iisec.ac.jp

‡ 警察庁
100-8974 東京都千代田区霞が関 2-1-2

**あらまし** RAT の通信を検知するためのこれまでの多く手法は，パケット単位でのネットワーク監視を必要としている．しかしながら，パケット単位での記録は容量が大きいため，長期間の保存は困難である．実際には，プロキシのログ等の限られた情報から，RAT の痕跡を検知しなければならない機会は少なくない．われわれが RAT の痕跡を含むプロキシのログを分析した結果，RAT の受信サイズや間隔等の挙動には特徴があることが判明した．本稿では，プロキシのログに記録された挙動から RAT の特徴ベクトルを作成し，機械学習により RAT の痕跡を検知する方式を提案する．さらに，そのプロキシのログを用いて提案方式の有効性を示す．

# Detecting RAT Activity in Proxy Server Logs with Machine Learning

Mamoru Mimura†　　Yuhei Otsubo†‡　　Hidehiko Tanaka†

†Institute of Information Security
2-14-1 Tsuruya-cho, Kanagawa-ku, Yokohama-city, Kanagawa 221-0835, JAPAN
dgs104101@iisec.ac.jp

‡National Police Agency
2-1-2 Kasumigaseki, Chiyoda-ku, Tokyo 100-8974, JAPAN

**Abstract** Many previous methods to detect RATs on the network require capturing packets. However, it is difficult to keep captured packets because the size is too huge. Actually, we would have to detect RAT activity through limited information such as proxy server logs. We analyzed proxy server logs including RAT activity, and found that the RATs had distinctive features in behavior such as sizes or intervals. In this paper, we make feature vectors from the behavior, and propose how to detect RAT activity with machine learning. Finally, we apply our method to the proxy server logs, and show the performance.

## 1　はじめに

特定の組織を狙った標的型攻撃による情報漏洩の被害は深刻である．2015 年には，多くの組織でマルウェアへの感染や情報漏洩の可能性が公表され，大きな社会問題となっている．これらの標的型攻撃では，端末を遠隔操作するための RAT(Remote Access Trojan または Remote Administration Tool) が使用されている．標的型攻撃に用いられる RAT は，難読化されて送り込まれることが多く，ウイルス対策ソフト等の従来の対策技術では検知することは困難であ

る．RAT を用いた標的型攻撃に対しては，これまでに動的解析を実施するサンドボックスタイプの製品や，各種のログを集約してその相関関係から感染を検知する手法等，様々な対策が提案されている．しかしながら，すべての組織で十分な対策が実施されているとは限らず，実際に攻撃は対策が不十分な組織で発生している．このような組織では，攻撃者の痕跡を調査するための十分なログが記録されていないことも少なくない．よってプロキシのログ等の限られた情報から，RAT の痕跡を検知する必要がでてくる．しかしながら，近年の標的型攻撃に用いられる主要な RAT は，独自のプロトコルを用いずに，一般的な HTTP で通信を実施し，自身の通信を正規の通信に紛れこませることを意図した動作が指摘されている [1]．したがって，膨大な容量のプロキシのログから RAT の痕跡を検知するのは，コマンド＆コントロール（以下Ｃ＆Ｃ）サーバのアドレスが不明な場合や，マルウェアの通信に固有の文字列が含まれない場合には困難である．

そこで本稿では，プロキシのログからパターンマッチングを用いずに，複数行のログから抽出する挙動のみを用いて RAT を検知する方式について検討した．われわれが RAT の痕跡を含むプロキシのログを分析した結果，RAT の受信サイズや間隔等の挙動には特徴があることが判明した．したがってその特徴を数値化し，機械学習により習得させることができれば，パターンマッチングを用いずに挙動から RAT を検知できる可能性がある．本稿では，Ｃ＆Ｃサーバのアドレスが未知であり，マルウェアの通信に固有の文字列が含まれない場合にも，プロキシのログから HTTP ベースの RAT の通信を検知することを目標とする．

以下，第２節では関連研究について説明し，本研究との違いを明確にする．第３節では提案方式とその実装について説明し，第４節では実際の RAT の痕跡を含むプロキシのログを用いて実験を実施する．第５節では実験の結果を踏まえて提案方式の実用性を評価し，最後にまとめと今後の課題について示す．

## 2　関連研究

HTTP ベースの RAT の通信の検知に関連する研究としては，ネットワークの監視によってマルウェアのＣ＆Ｃサーバとの通信を検知するための研究と，プロキシのログから不正な接続先を検知するための研究が挙げられる．以下，提案方式とこれらの研究との違いについて説明する．

### 2.1　ネットワーク監視による手法

文献 [2] では，パケットサイズ，パケット数，到着間隔等の特徴量を用い，Ada Boost で通常の通信と不正な通信を区別することで，マルウェアへの感染を検知する手法を提案している．文献 [3] では，パケット数，データサイズ，セッション時間，アクセス回数およびアクセス時間の標準偏差を特徴ベクトルとして，Support Vector Machine によりＣ＆Ｃトラフィックを抽出する手法を提案している．文献 [4] では，セッション毎に合計パケット数，初期段階のセッションの存続時間，データサイズ，パケット数およびパケットの平均データサイズを特徴ベクトルとし，決定木と Random Forests により RAT による通信か否かを判定する手法を提案している．これらの手法では，パケット単位でのトラフィックの監視が必要である．

文献 [5] では，DNS クエリの挙動を分析し，不正な未知のドメインを検知する手法を提案している．この手法では，ISP 規模での DNS クエリの監視が必要である．

提案方式では，RAT による通信か否かの判定に，Support Vector Machine(以下 SVM) 及び Random Forests(以下 RF) を用いている点が従来の研究と共通している．しかしながら，ネットワーク監視を必要とせず，プロキシのログのみを対象としている点が異なっている．

### 2.2　プロキシのログを使用する手法

文献 [6] では，HTTP ベースのマルウェアを分類するために，リクエストの数，GET の数，POST の数，URL の平均の長さ，パラメータ

の平均数，POSTで送信したデータの平均サイズ，平均の応答のサイズを使用している．この手法では，さらにクエリの内容も分析してクラスターに分類し，マルウェアのサンプルによるクラスターと類似性を比較し，シグネチャを自動生成している．この手法では，プロキシのログから取得できる項目を用いており，その平均に着目して挙動を抽出している．

文献[7]では，プロキシのログからクライアントのアドレス，訪問先のアドレスおよびリクエストの数を用い，クライアントと共通するサーバに着目してグループに分類し，疑わしいドメインの検出を支援する手法を提案している．この手法では，疑わしいドメインの検出をブラックリスト等の他の手法に依存している．

文献[8]では，DNSのログ，プロキシのログ等を使用し，内部ホストの訪問履歴とそのUser Agentから，組織全体の希少な訪問サイト，User Agentの傾向，ドメインの類似性等を分析し，通常状態と比較することで異常なドメインを検出する手法を提案している．この手法では，プロキシ以外にもDNSのログを必要としている．また，疑わしいドメインを検出するために，ドメインの登録情報，ブラックリスト等の外部からの情報を必要としている．

文献[9]では，マルウェアの感染がないと想定する期間のプロキシのログと，マルウェアの感染を疑う期間のプロキシのログを比較することで，効率的にログを縮退する手法を提案している．この手法では，最終的には熟練ネットワーク管理者による判断が必要である．

提案方式は，プロキシのログから取得する項目の頻度に着目して挙動を抽出し，外部からの情報や人手による判定を必要とせず，自動的にRATの通信と通常の通信を識別する．

# 3 提案方式

## 3.1 前提条件

本稿で提案する検知方式を実現するための前提条件は，プロキシのログに以下の項目が記録されていることである．

- 時刻
- リクエストの内容（メソッド，URLおよびUser Agentを含む.）
- HTTPステータスコード
- クライアントが受信したサイズ

これらの項目は，標準ログフォーマットに含まれており，多くのプロキシで取得可能であると考えられる．提案方式では，複数行のログから特徴となる挙動を抽出する．まず，HTTPステータスコードが成功の行を対象として，URLに含まれるホスト毎にあらかじめ指定した行数のログを抽出する．次に，抽出した指定行数のログに含まれる時刻，リクエストの内容およびクライアントが受信したサイズから特徴ベクトルを作成する．

## 3.2 特徴ベクトル

提案方式において，指定行数のログから作成する特徴ベクトルを以下に示す．

① 最頻出の受信サイズ

② 最頻出の受信サイズの数

③ 最頻出のリクエストの間隔

④ 最頻出のリクエストの間隔の数

⑤ 最頻出のpathの長さ

⑥ 最頻出のpathの長さの数

⑦ POSTメソッドの数

⑧ User Agentの長さ

特徴ベクトルは，固有の文字列を用いずに，項目の頻度に着目して作成した．①から④は，受信サイズおよび間隔（直前のログの時刻との差分）のヒストグラムを図1に示すように作成し，最も頻度が高い受信サイズおよび間隔とその数とした．⑤および⑥は，図2に示すようにRATは同じpathに連続してアクセスするという特徴を数値化したものである．⑦は，一部のRATは特定のメソッドを多用することに着目

している．⑧は，User Agent の違いを考慮させるために選定した．

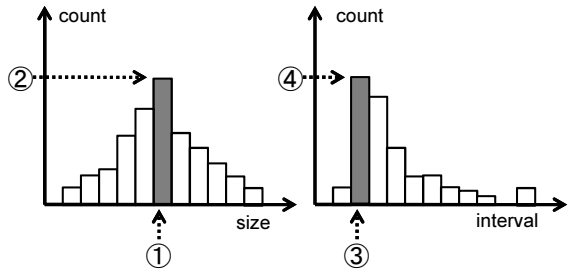


図 1: 受信サイズと間隔のヒストグラム

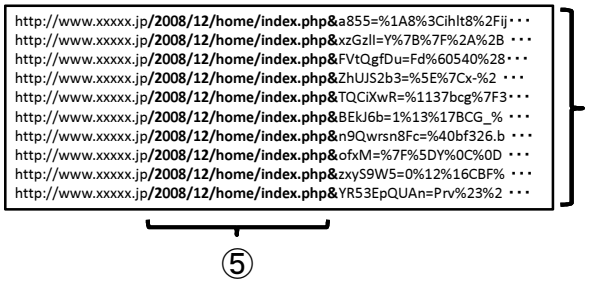


図 2: RAT のアクセスログの例

### 3.3 学習と予測

提案方式では，教師あり学習モデルである SVM と RF を用いる．そのため，訓練データとして検知対象とする RAT の痕跡を含む既知のログが必要であり，かつそのログにおいて RAT の通信と通常の通信の区別がついている必要がある．提案方式の動作は，学習フェーズと予測フェーズに分類される．

**学習フェーズ**

検知対象とする RAT の痕跡を含むログを読み込み，ホスト毎に指定行数のログから特徴ベクトルを作成する．次に，特徴ベクトルが RAT による通信であれば RAT の種類，それ以外であれば通常の通信のラベルを付与し，SVM または RF に学習させる．

**予測フェーズ**

対象とする未知のログを読み込み，学習フェーズと同様に，ホスト毎に指定行数のログから特徴ベクトルを作成する．次に，その特徴ベクトルを SVM または RF に予測させ，RAT の種類か通常の通信のラベルを出力させる．

提案方式では以上の動作により，対象とするログから HTTP ベースの RAT の種類を検知する．

### 3.4 実装

提案方式を，Python-2.7 と機械学習のライブラリを活用して実装した．SVM については libsvm-3.2[10] の C-SVC（ソフトマージン識別器）を用いた．カーネル関数については，分類するデータに関する事前知識がないことから，汎用的な用途で用いられる RBF（ラジアル基底関数）カーネルを選択した．RF については scikit-learn-0.16.1[11] の RandomForestClassifier を用いた．SVM，RF ともに，その他のパラメータについては，デフォルトの値となっている．

## 4 実験

### 4.1 実験内容

実装したプログラムと実際のプロキシのログを用いて実験を実施する．実験環境および実験に使用するプロキシのログの概要を表 1 および表 2 に示す．このログは，2015 年に標的型攻撃を受けたある組織のプロキシのログであり，標準ログフォーマットで記録されている．このログには，２タイプの RAT による遠隔操作の痕跡が含まれており，それぞれのタイプの RAT の接続先が判明している．２タイプの RAT は，2010 年以降に出現した比較的新しい RAT であり，近年の主要な標的型攻撃に使用されている HTTP ベースの RAT である．実験では，学習フェーズで訓練データを読み込み，予測フェーズでテストデータを読み込ませる．なお，今回

の実験で使用する訓練データは，テストデータに含まれている．

表 1: 実験環境

| | |
|---|---|
| CPU | Core i5-3450 3.1GHz |
| Memory | DDR3 SDRAM 8GB |
| HDD | Serial ATA 600 |
| OS | Windows 7 |

表 2: 実験データ

| | 訓練データ | テストデータ |
|---|---|---|
| 期間 | 1 日 | 約 1 か月 |
| 容量 | 約 250MB | 約 40GB |
| RAT の種類 | 2 タイプ | 2 タイプ |

### 4.2 実験結果

まず，ログの行数（以下ｎ）毎の特徴ベクトルの数を表 3 に示す．特徴ベクトルの総数は，訓練データ，テストデータともにｎを減らすたびに増加している．ｎと特徴ベクトルの総数が反比例とならないのは，指定したｎに満たない数回のアクセスのみのログは特徴ベクトルに反映されないためである．訓練データに含まれる RAT の特徴ベクトルの数は，テストデータの 10%未満となっている．

次に，実験の検知率および誤検知数を表 4 に，所要時間を表 5 示す．SVM はｎを 10 以下にすると顕著に学習時間が長くなり，検知率では RF に劣るものの，誤検知数は少ない結果となった．また，ｎを 50 以上にした場合には顕著に検知率が低下した．これに対し，RF ではｎを減らしてもあまり学習時間は長くならず，全般的に高い検知率を示すが，誤検知も少し発生する結果となった．また，どちらの場合にも，ｎを 5 以下に減らした場合には，誤検知数が顕著に増加した．

表 3: 特徴ベクトルの数

| ログの | 訓練データ | | テストデータ | |
|---|---|---|---|---|
| 行数ｎ | RAT | 総数 | RAT | 総数 |
| 100 | 42 | 1605 | 429 | 160209 |
| 50 | 87 | 3648 | 905 | 365171 |
| 30 | 148 | 6627 | 1564 | 663101 |
| 20 | 225 | 10553 | 2390 | 1056916 |
| 10 | 456 | 22865 | 4888 | 2289650 |
| 5 | 918 | 48185 | 9890 | 4819510 |

表 4: 検知率

| ログの | 検知率 (DR) | | 誤検知数 (FPC) | |
|---|---|---|---|---|
| 行数ｎ | SVM | RF | SVM | RF |
| 100 | 22.1% | 97.7% | 0 | 10 |
| 50 | 61.0% | 97.7% | 0 | 0 |
| 30 | 88.2% | 98.8% | 0 | 3 |
| 20 | 95.3% | 98.5% | 0 | 3 |
| 10 | 96.8% | 98.9% | 1 | 4 |
| 5 | 98.6% | 99.2% | 32 | 47 |

### 4.3 MWS データセットへの適用

今回の実験で使用した訓練データを用い，MWS データセット [12] で追加実験を実施した．実験に使用したデータは，BOS 2015 に含まれるすべての pcap ファイルであり，この中には RAT の痕跡が含まれている．この pcap ファイルをプロキシのログに相当する擬似ログに変換するために，まず HTTP プロトコルを抽出し，リクエストとレスポンスの対応付けを実施した．さらに，そのリクエストとレスポンスのペアから以下の情報を抽出し，擬似ログを作成した．

- 時刻
- リクエストの内容
- HTTP ステータスコード
- レスポンスのサイズ

同様の手法により，NCD in MWSCup 2014 から RAT の痕跡を含まない通常の通信の擬似

表 5: 所要時間

| ログの行数 n | 学習時間 | | 予測時間 | |
|---|---|---|---|---|
| | SVM | RF | SVM | RF |
| 100 | 33s | 33s | 1h23m | 1h25m |
| 50 | 34s | 33s | 1h25m | 1h27m |
| 30 | 37s | 34s | 1h27m | 1h31m |
| 20 | 42s | 34s | 1h30m | 1h36m |
| 10 | 57s | 34s | 1h38m | 1h48m |
| 5 | 1m45s | 35s | 2h02m | 2h12m |

ログを作成した．作成した擬似ログの概要を表 6 に示す．学習フェーズでは今回の実験に使用した訓練データを読み込み，予測フェーズでは BOS 2015 および NCD in MWSCup 2014 から作成した擬似ログを読み込ませた．機械学習は RF を選択し，n は 30 とした．その結果，BOS 2015 から 4 つの不正な接続先をすべて検知し，NCD in MWSCup 2014 から誤検知は発生しなかった．

表 6: 擬似ログの概要

| | BOS 2015 | NCD |
|---|---|---|
| 期間 | 12 日 | 1 日 |
| 容量 | 約 1 MB | 約 8 MB |
| RAT の種類 | 1 タイプ | — |

# 5 評価

## 5.1 検知率

実験の結果，SVM については，n を 20 以下にすると検知率は 95%以上となった．誤検知については，n を 10 以上にするとほとんど発生しなくなった．したがって，SVM の場合の最適な n は，10〜20 程度であると言える．

RF については，n を 30 以下にすると検知率は 98%以上となった．誤検知については，n を 10 以上にすると 10 件未満となった．したがって，RF の場合の最適な n は，10〜30 程度であると言える．

誤検知や見逃しの原因は，HTTP ベースの RAT に特徴ベクトルが類似しているサイトであった．たとえば，動画のストリーミング再生や，何らかの API を提供するサイトに関しては，同一の受信サイズ，あるいは同一に近い受信サイズの通信が定期的に繰り返される傾向が認められた．また，検知率には反映されていないが，設定した n に満たない場合にはそもそも特徴ベクトルが作成できないため，見逃しの可能性がある点にも注意する必要がある．

## 5.2 所要時間

SVM については，n を 10 以下にすると，顕著に学習時間が長くなる傾向が認められた．この原因は，特徴ベクトルが増加したためであると考えられる．予測時間については，特徴ベクトルの数に応じて緩やかに長くなる傾向が認められた．

これに対し，RF については n を減らしても，学習時間はあまり長くならず，概ね一定となる傾向が認められた．予測時間については，SVM と同様に特徴ベクトルの数に応じて緩やかに長くなる傾向が認められた．

この結果から，訓練データが少なく，特徴ベクトルが少ない場合には SVM の方が高速であることが確認できた．しかしながら，SVM では特徴ベクトルが多くなると顕著に学習時間が長くなることから，訓練データが多い場合には RF の方が高速になるものと考えられる．これは，学習が高速な RF の一般的な特性によるものであると考えられる．

## 5.3 実用性

提案方式は，標的型攻撃を受けた組織のプロキシのログを詳細に分析する用途と，ネットワークをリアルタイムで監視する用途を想定している．どちらの用途においても，提案方式では標準ログフォーマットに含まれている項目のみを用いるため，様々な組織の機器や様々な状況に

適応可能であると考えられる．仮に一部の項目が取得できなかったとしても，他の取得できた項目のみを用いて提案方式を活用することも可能である．たとえば，拡張ログフォーマットに含まれる User Agent を取得できなかった場合，共通ログフォーマットに含まれる他の項目のみから特徴ベクトルを作成することも可能である．参考としてこの場合の検知率を示すと，ｎを 30 に設定した SVM で 93.7%，RF で 97.4%であり，誤検知数はいずれも５件以下であった．これに対し，既存の研究ではネットワーク監視，外部からの情報の取得等を前提としているため，現実的には適応できる状況は限られていると考えられる．

攻撃を受けた組織のログを詳細に分析する用途に着目すると，提案方式は，特に標的型攻撃の対策が不十分であり，攻撃の痕跡がプロキシのログのみである状況で特に有用であると考えられる．このような詳細に分析する用途では，検知率に優れる RF を用いることで，ほとんど見逃すことなく HTTP ベースの RAT を検知することが可能であると考えられる．本稿における実験データは，訓練データがテストデータに含まれている．これは，デジタルフォレンジック技術等を用いて復元した一部のマルウェアを分析し，そのＣ＆Ｃサーバの一部の URL が判明した場合を想定している．このような場合，判明したＣ＆Ｃサーバの一部の URL のアクセスログを訓練データとし，他のまだ発見されていない未知のＣ＆Ｃサーバの URL を検知するために利用することが可能である．

提案方式は，SVM，RF ともに約１か月分で約 40GB のログを２時間以内に処理しており，リアルタイムで監視する用途での運用も可能であると考えられる．リアルタイムで運用する場合には，発生する誤検知の数をオペレータが処理できる数に抑える必要がある．10〜20 程度のｎを設定して SVM を選択すれば，誤検知をなるべく抑えつつ，95%以上の検知率を実現することが可能である．さらに検知率を高めるためには，RF を選択してｎを 10〜30 程度に設定すれば，98%の検知率を実現することができる．RF の場合にはやや誤検知が発生するが，その数がオペレータが処理できる量であれば運用に支障はないものと考えられる．

### 5.4　制約

提案方式が機能するためには，前提条件で示したとおり，標準ログフォーマットに含まれている項目がプロキシのログに記録されている必要がある．従来の機械学習によりマルウェアの挙動を検知する手法のほとんどは，パケット単位でのネットワーク監視を前提条件としている．しかしながら，標的型攻撃の被害を受けた組織において，パケット単位での記録を長期間保存している可能性は低いものと考えられる．したがって，提案方式の前提条件はより実用的であると言える．

提案方式では，訓練データとして検知対象とする RAT の痕跡を含むログが必要であり，かつそのログにおいて RAT の通信と通常の通信の区別がついている必要がある．通常は，RAT の痕跡を含むログは標的型攻撃を受けた組織が保有しているため，入手できる機会は限られている．ログを入手した場合には，関連する検体の分析結果から得られる特徴，ブラックリスト等を用いて RAT の通信を抽出し，それ以外を通常の通信として訓練データを作成することができる．すでに別の訓練データや特徴ベクトルを保有している場合には，それらを用いて RAT の通信を抽出することも可能である．また，検体等が入手できれば，ハニーポットを運用することで作成することも可能である．

運用においては，設定したｎが多い場合には見逃しの可能性があることに注意する必要がある．この制約に関しては，RF を選択してｎを少なめに設定し，ホワイトリスト等を併用して誤検知を除外することで，ある程度は緩和することが可能である．

## 6　おわりに

本稿では，プロキシのログから特徴ベクトルを作成し，機械学習により HTTP ベースの RAT を検知する方式を提案した．さらに，提案方式

を実際に標的型攻撃を受けた組織のプロキシのログに適用し，近年の主要な標的型攻撃で使用されているRATを95%以上の高精度で検知できることを示した．最後に，実験結果を考察し，提案方式の実用性を評価した．

今後の課題としては，他のタイプのRATや，他の標的型攻撃を受けた組織のプロキシのログへの適用が挙げられる．本稿では主要な2つのタイプのRATに対する有効性を示したが，他のタイプのRATに対する効果は明確ではない．他の新たなタイプのRATが出現した場合には，特徴ベクトルを再検討する必要がでてくる可能性がある．また，リアルタイム検知システムへの応用も今後の課題である．

## 参考文献


[1] 標的型サイバー攻撃分析レポート 2015年版 ～「気付けない攻撃」の高度化が進む～ (online) http://www.go-tm.jp/apt2015/ (2015-07-24).

[2] 市野 将嗣，市田 達也，畑田 充弘，小松 尚久：トラヒックの時系列データを考慮したAdaBoostに基づくマルウェア感染検知手法，情報処理学会論文誌，Vol.53, No.9, pp.2062–2074 (2012).

[3] 山内 一将，川本 淳平，堀 良彰，櫻井 幸一：機械学習を用いたセッション分類によるC&Cトラフィック抽出，2014年暗号と情報セキュリティシンポジウム (2014).

[4] 蒋 丹，面 和成：初期段階におけるRemote Access Trojanの検知手法，コンピュータセキュリティシンポジウム2014 (2014).

[5] Babak Rahbarinia, Roberto Perdisci, Manos Antonakakis：Segugio: Efficient Behavior-Based Tracking of New Malware-Control Domains in Large ISP Networks, *Proc. 2015 IEEE/IFIP International Conference on Dependable Systems and Networks* (2015).

[6] Roberto Perdisci, Wenke Lee, Nick Feamster：Behavioral Clustering of HTTP-based Malware and Signature Generation using Malicious Network Traces, *Proc. 2010 USENIX Symposium on Networked Systems Design and Implementation* (2010).

[7] Manh Cong Tran and Yasuhiro Nakamura：A Supplementary Method for Malicious Detection, Journal of Communications, Vol.9, No.12, pp.923-929 (2014).

[8] Alina Oprea, Zhou Li, Ting-Fang Yen, Sang Chin and Sumayah A. Alrwais：Detection of Early-Stage Enterprise Infection by Mining Large-Scale Log Data, CoRR, Vol.abs/1411.5005 (2014).

[9] 田中 功一，堀川 博史，蜂野 博史，西垣 正勝：ログ解析によるマルウェア侵入検知手法の提案，マルチメディア，分散，協調とモバイルシンポジウム2014 (2014).

[10] libsvm (online) https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (2015-07-24).

[11] scikit-learn (online) http://scikit-learn.org/ (2015-07-24).

[12] 神薗 雅紀，秋山 満昭，笠間 貴弘，村上 純一，畑田 充弘，寺田 真敏：マルウェア対策のための研究用データセット ～MWS Datasets 2015～，情報処理学会研究報告，Vol.2015-CSEC-70, No.6 (2015).