九州大学学術情報リポジトリ
Kyushu University Institutional Repository

近年のハニーポット取得データ解析による C&C トラフィック分類評価

山内, 一将
九州大学 | 九州先端科学技術研究所

川本, 淳平
九州大学 | 九州先端科学技術研究所

堀 ,良彰
佐賀大学 | 九州先端科学技術研究所

櫻井, 幸一
九州大学 | 九州先端科学技術研究所

https://hdl.handle.net/2324/1662074

出版情報：暗号と情報セキュリティシンポジウム. 2015, pp.2A1-4-, 2015-01-20
バージョン：
権利関係：

KYUSHU UNIVERSITY

Copyright ©2015 The Institute of Electronics,
Information and Communication Engineers

SCIS 2015 The 32nd Symposium on
Cryptography and Information Security
Kokura, Japan, Jan. 20 - 23, 2015
The Institute of Electronics,
Information and Communication Engineers

近年のハニーポット取得データ解析による C&C トラフィック分類評価
Evaluation of C&C Traffic Classification by Recent Years Honeypot Data Analysis

山内 一将 *† Kazumasa Yamauchi
川本 淳平 *† Junpei Kawamoto
堀 良彰 ‡† Yoshiaki Hori
櫻井 幸一 *† Kouichi Sakurai

あらまし インターネットの普及に伴い，マルウェアへの感染被害の増加が世界中で深刻な問題となっている．その中でもボットネットによる被害が顕著である．一般的なボットネットでは踏み台となるコンピュータの制御や命令を行なうために Command and Control（C&C）サーバを利用しており，ボットネット対策手法の 1 つとして C&C サーバの検知が注目されている．従来のボットネットにおいて C&C サーバを制御するプロトコルとして Internet Relay Chat（IRC）を利用するものが多かった．しかし，近年では HTTP，P2P のようなユーザへの普及率が高いプロトコルを利用したボットネットも出現しており，解析するべきトラフィック量が多くなるために検知が難しい．故に IRC プロトコルに特化したボットネット検知手法には限界がある．本研究では，近年採取されたハニーポットデータの解析結果を基に，C&C サーバの利用するプロトコルに特化しない検出手法を提案する．この手法を用いて C&C サーバの検出を行い C&C サーバとボットの通信を遮断することでボットネットからの攻撃を未然に防ぐことが我々の目的である．また，正常な通信と C&C サーバによる通信の分類を行うための評価実験を行うことで，本手法の有効性について示す．

キーワード ボットネット，C&C サーバ，異常検知，機械学習

1 はじめに

近年インターネットの普及に伴い，ボットネットによる脅威が問題となっている．ボットネット対策を行うために多くの研究がなされており，その中で Command & Control (C&C) サーバを特定する手法がある [1]．C&C サーバは，攻撃者がボットネットを運用するために利用するサーバである．C&C サーバは攻撃者からの指令をボットに感染した端末群へ転送する．C&C サーバから命令を受けた端末群は指令の内容に従って，DDoS 攻撃，スパムメールの送信，脆弱性スキャン攻撃などを行う．つまり，C&C サーバ通信（C&C トラフィック）はボットネットによる攻撃の予兆として考えられており，攻撃を未然に防ぐための 1 つの手法として C&C サーバの特定が必要とされている．

1.1 既存研究

C&C サーバが利用するプロトコルは IRC，HTTP，P2P などに分類できる．本稿では，簡略化のために IRC を利用したボットネットを IRC 型ボットネットと呼ぶ．同様に HTTP，P2P に関しても HTTP 型ボットネット，P2P 型ボットネットと呼ぶ．

IRC 型ボットネットは 1993 年頃から用いられてきたボットネットである．IRC 型ボットネットにおける C&C サーバの検知手法として，文献 [2][3] がある．文献 [2] では，IRC クライアントが C&C サーバと行う通信に関して，3 種類の特徴ベクトルを定義している．これらの特徴ベクトルに対して，機械学習を用いて分類した検知率，誤検知率，見逃し率の評価を行っている．文献 [3] では，IRC プロトコルの通信特性に基いてボット検知を行っている．著者らは，IRC プロトコルを用いた通信特性を見つけるために，スコア関数やブラックリスト/ホワイトリスト方式と n-gram による分析を組み合わせている．

また，2003 年頃からはボットネット制御の中心となる C&C サーバを必要としない P2P 型ボットネットが出現している．P2P 型ボットネットでは端末間で直接通信を行い，攻撃者からの指令は端末同士で共有して拡散されるため，ボットネットの攻撃の対策が困難である．P2P 型ボットネット対策に関する既存研究として，PeerShark[4] がある．文献 [4] では，ボットネットを他の通信と区別するために，通信の頻度やデータサイズな

* 九州大学, 福岡県福岡市西区元岡 744 番地
† 九州先端科学技術研究所，福岡市早良区百道浜 2 丁目 1 番 2 2 号 福岡 SRP センタービル 7 階
‡ 佐賀大学, 佐賀県佐賀市本庄町 1 番地

どを基に分類を行う手法を提案している．

IRC 型ボットネットや P2P 型ボットネットに比べて新しいボットネットとして HTTP 型ボットネットがある．HTTP 型ボットネットは 2005 年頃に確認されている．HTTP 型ボットネットは IRC 型ボットネット，P2P 型ボットネットに比べると新しい形態のボットネットであり，近年増加傾向にある．HTTP 型ボットネットの対策手法として，HTTP メソッドに着目した研究がある [5][6]．HTTP プロトコルでは，データを取得する際のメソッドとして GET や POST などが利用される．そこで，これら HTTP メソッドを送信する時間に着目して，著者らが提案した方式によりクラスタリングを行っている [5]．文献 [6] では，Artificial Immune System(AIS)[10] を用いたリアルタイム検知を行う手法を提案している．一般的な AIS は生物の免疫系の原理やプロセスをモデル化したものであり，この概念を著者らの手法に組み込むことで HTTP 型ボットネットの検知をより効率的に行うことができると述べている．

1.2 研究課題と貢献

ボットネットの構造や利用するプロトコルが多様化しているため，これらの変化に対応した手法が必要である．文献 [3][4][5] はそれぞれ 1 つのプロトコルに特化した検知手法となっている．そのため，網羅的なボットネット検知が困難であり，また新しいプロトコルを利用したボットネットの検知精度が低くなる．また，ボットネット対策手法の実用化に向けて実データを用いた評価が重要であるが，文献 [5][6] では実用化に向けた実データを利用しての C&C トラフィックの検知に関する評価ができていない．また，[3] では IRC プロトコルで用いられる，ニックネームと呼ばれる識別子を利用たシグネチャベースの検知を行っているため，未知のシグネチャに対する検知が難しい．

本研究ではプロトコルに特化しないボットネット検知手法を提案する．我々が行った貢献は 2 つある．

- ハニーポット取得データの解析
  ハニーポットは，攻撃者の侵入手法やマルウェアの挙動を調査，研究するためにインターネット上に設置された，意図的な脆弱性を持たせたシステムである．本研究では C&C トラフィック特性調査を行うために CCCDataSet'09(以下，C09)，CCCDataSet(以下，C10),PRACTICE'13(以下，P13) を用いた．C09,C10 はサイバークリーンセンターに設置されているハニーポットで収集されたボットの観測データであり，P13 は総務省「国際連携によるサイバー攻撃予知・即応に関する実証実験」(略称：PRACICE) の挙動観察システムで，マルウェア感染後の通信挙動を長期観測した際の通信トラフィックを示すデータである [7]．

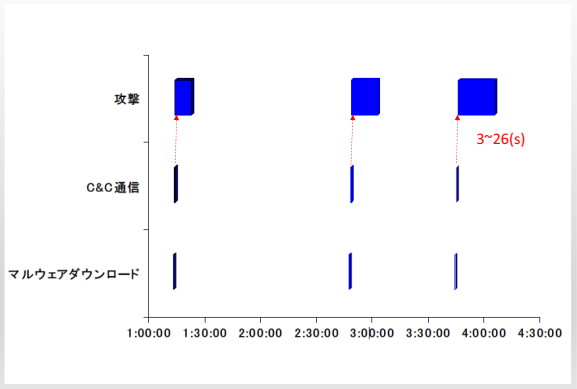

図 1: ボットネットの時系列調査

- C&C トラフィック抽出実験
  実ネットワークを流れる通信データにおいて，C&C トラフィックは通常の通信に紛れている．そのため，通常のトラフィックと C&C トラフィックを分類する必要がある．そこで前項で得られた C&C トラフィックの特性を元に特徴ベクトルを定義し，機械学習の分類アルゴリズムを用いて C&C トラフィックの抽出を行う．

本稿の構成として，まず 2 章でボットネットについて述べ，3 章で C&C トラフィックの特性調査を行う．そして，4 章で評価実験を行い，5 章で実験結果と考察ついて述べて，6 章で結論となる．

2 ボットネット

ボットネットは悪意のある活動を目的としたネットワークのことであり，攻撃者が第三者のコンピュータに悪性プログラムを忍び込ませる事により作成された複数のボットとボットネットの制御を司る C&C サーバによって構成される．規模としては小さいものでは数十台，大きいものでは数十万台ものボットで構成される．本章ではボットネットが行う挙動に関して時系列調査を行う．

ボットネットの攻撃の挙動に関して図 1 に示す．ここでは 1 台のハニーポットがマルウェアに感染してから攻撃に至るまでを時系列で示しており，C10 で取得されたハニーポット上で実行されたマルウェアのトラフィックに関して調査することで図 1 に示す挙動を観測した．図 1 ではマルウェアダウンロード，C&C サーバとの通信，

攻撃という 3 つのフェーズに分け，午前 1 時から午前 4 時までを観測している．まずマルウェアダウンロードを，ボットネットが一般ユーザに対して脆弱性探索に成功した場合に行う．ダウンロード時間に関しては，3 回の平均で 3 秒程度であった．次にマルウェアに感染してボット化した PC は C&C サーバと通信を行う．通信時間は 3 回の平均で 19 秒程度であった．C&C サーバから指令をもらったボットは攻撃を行う．攻撃の時間としては 3 回の平均で 14 分 7 秒程度であった．

このように，ボットネットが段階的に攻撃へ至るまでには何らかの予兆が観測される．その予兆の 1 つに C&C サーバとの通信がある．また，C10 の時系列分析結果では C&C サーバとボットの通信が終わってから，ボットが攻撃に至るまでの時間は 3 秒から 26 秒程度であることが分かった．これらの結果を考慮して，本研究ではボットネットの行う攻撃を未然に防ぐために C&C サーバとの通信を予兆検知することを目的とする．

3 C&C トラフィック特性調査

本研究では C&C サーバとボットの通信に着目し，パケットのヘッダ情報を利用してトラフィック解析を行う．本章では C&C セッション分析を行うために，C&C サーバの用いるプロトコルの説明と特徴ベクトルの定義について述べる．

3.1 ボットネットの通信

ボットネットを制御する C&C サーバはボットとの通信手段として大きく 2 つの形態を取る．1 つ目は，C&C サーバからボットへの一方向的な通信である．これは IRC を利用する場合に主に起こる．IRC をベースにしたボットネットは攻撃者に従来用いられてきた手法であり，これに関する既存研究も 1.1 節に示したように多くある．2 つ目は C&C サーバとボットの両方向の通信である．これは HTTP や P2P などを利用する場合に行われる．特に HTTP に関して，IRC よりも一般ユーザに普及しているプロトコルであるため HTTP 型ボットネットは増加傾向にある．HTTP の通信量は IRC に比べて大きく，異常な通信のみを正確に取り出すことが困難な傾向にある．

3.2 特徴ベクトル

本節では C&C サーバの通信特性について調べる．文献 [8] ではマルウェア検知のための特徴量として 36 個の要素を選択し，正常トラフィックと感染トラフィックの分離を行っている．しかし，文献 [8] では通信方向の区別を行っていない．C&C サーバとボットの通信を取り出す場合には送信，受信の双方向通信に着目することが有効である．例えば，IRC サーバにおいてあるクライアントがチャットをするためのグループ構成で必要となるチャンネルに参加した場合に，一種の通常な通信としてクライアント側ではパケットを送信，受信しながらメッセージを交換することが考えられる．一方で，ボットがチャンネルに参加する場合はユーザの意図していない部分での挙動となるため，チャンネルに参加したボットは自発的に C&C サーバに向けてメッセージを送ることが考えにくい．そのため，C&C サーバからボットに対してメッセージを一方的に受信することが考えられる．故に本研究で用いる特徴ベクトルには双方向の通信を考慮したものを定義する．

また，文献 [5] では HTTP 型ボットが C&C サーバへアクセスする挙動の周期性に着目していた．しかし，周期的なアクセスは HTTP 型ボットネットに限ったことではなく，HTTP と同様にセッション DNS や P2P でも同じことが言えると考えられる．また，文献 [2] では IRC 型ボットネットを検知するためパケット数，パケットサイズなどを考慮した特徴ベクトルを定義していたが，ボットが C&C サーバへアクセス挙動に関する属性が含まれていない．故にアクセス挙動を考慮した特徴ベクトルを定義することが本研究において有効である．

従って，我々は特徴ベクトルを表 1 の要素を用いて定義する．表 1 では V_1 から V_7 までの 7 つの属性を特徴ベクトルとして扱っている．本研究ではサーバ/クライアント通信に着目し，クライアントがサーバに TCP を用いて接続してから，接続を切るまでの通信を 1 つのセッションとして解析する．また，V_6, V_7 ではアクセス挙動特性を考慮した属性を定義しており，本稿でのアクセスとはクライアントがサーバに TCP を用いて接続することを指す．ボットネットではクライアントをボットとすると，接続先は C&C サーバとなり，正常なサーバ/クライアント通信との比較が可能である．

表 1 で示す特徴ベクトルの要素に関して説明する．V_1, V_2 はそれぞれボット（クライアント）が C&C（IRC または HTTP を）サーバへ 1 回のセッションで送ったパケット，データサイズの総数を指す．同様に，V_3, V_4 ではそれぞれボットが C&C サーバから受信したパケット，データサイズの総数を指す．データサイズの総数に関しては，パケットのヘッダ情報を基にセッション毎に含まれているパケットのデータサイズを合計したものである．V_5 はパケットのヘッダ情報に含まれるタイムスタンプを確認し，セッション終了時刻からセッション開始時刻の差をとった時間である．また，V_6 はセッション中にクライアントがサーバへアクセスする回数の合計を指し，V_7 はアクセス時間のばらつきを表している．

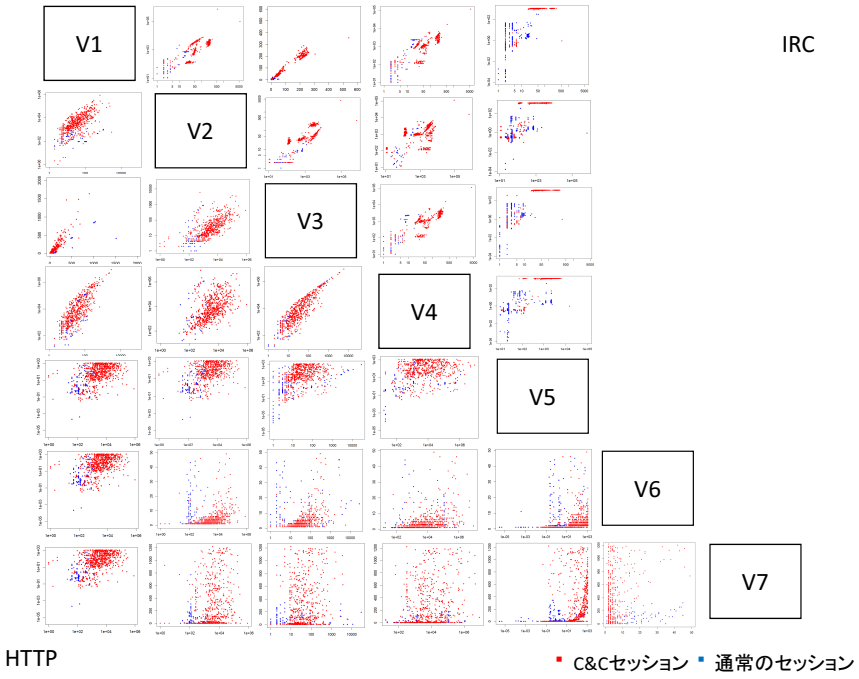

図 2: C&C セッション分析 (全結果)

3.3 C&C セッション分析

本節では通常のセッションと C&C セッションに関して，提案する特徴ベクトルを用いて分類可能であるかについての分析を行う．図 2 ではセッション分析を行った結果を示している．図 2 は IRC と HTTP の通信に関してそれぞれ特徴ベクトルを用いて特徴量抽出を行い，その結果についてグラフ化したものであり．通常の HTTP または IRC セッションは青で，C&C セッションは赤で示している．また，IRC に関しては，セッション中に C&C サーバへの再接続を行わないので V_6, V_7 に関しては考慮しない．図 2 の IRC と HTTP での解析結果の比較から，IRC の方がデータの分布範囲が狭いことが分かる．これに対し，HTTP ではデータの分布範囲が広く，IRC よりも通信の多様性が見られる．

図 2 から一部のデータを拡大したものを図 3, 図 4, 図 5 に示す．図 3 に関しては，ほとんどの IRC セッションが送信パケット数，受信パケット数共に 25 から 500 に集中しており，割合はおよそ 1 対 1 であるデータが多い．これに対し C&C セッションでは送信パケット数，受信パケット数が比較的少なく，割合はおよそ 1 対 1 から外れるものが多い．図 4 に関しては，ほとんどの HTTP セッションと C&C セッションは送信パケット数，受信パケット数の双方で 5 から 10000 までに幅広く分布している．しかし，HTTP セッションはおよそ 1 対 1 の割合で分布しているのに対して C&C セッションはおよそ 1 対 1 から外れるものが多い．図 5 に関しては，HTTP セッションではアクセス回数に関係なくアクセス時間間隔の標準偏差が大きいことが分かる．これに対して，C&C セッションではアクセス回数が増えてもアクセス時間の標準偏差が急激に増えることがない．

これらの結果より，我々が提案する特徴ベクトルを用いたセッション分類を高い検知率，低い誤検知率で実現できると考えられる．次の章では実データを用いた評価実験を行う．

表 1: 特徴ベクトル

V_1	送信パケット数 (PKT)
V_2	送信データサイズ (Byte)
V_3	受信パケット数 (PKT)
V_4	受信データサイズ (Byte)
V_5	セッション時間 (s)
V_6	アクセス回数 (回)
V_7	アクセス時間標準偏差

4 評価実験

C&C トラフィックを検出するために，実データを用いた実験についての説明を行う．図 6 は，今回行った実験の流れを示している．本章では我々が行った実験に関してセッションデータ抽出，セッションデータ解析，機械学習を用いた分類に関してそれぞれ説明する．

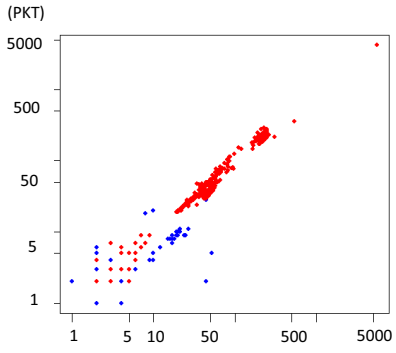

図 3: $V_1 - V_3(IRC)$

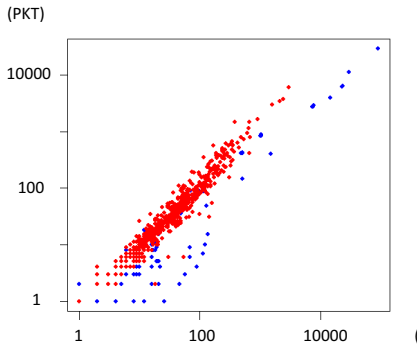

図 4: $V_1 - V_3(HTTP)$

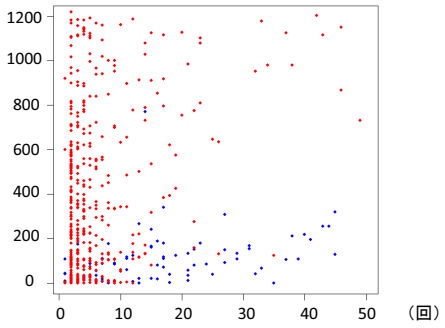

図 5: $V_6 - V_7(HTTP)$

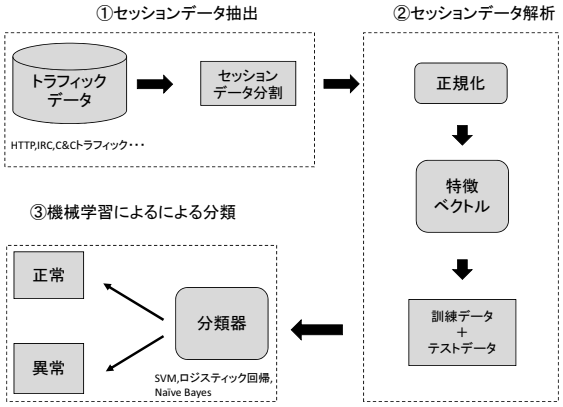

図 6: 実験の流れ

4.1 セッションデータ抽出

今回の実験では予め採取されているデータを利用する．そのため，正常，異常と判断することのできるデータを初めに用意する．データ収集の方法として，Linux コマンドライン上でパケットを観測可能なツールである tcpdump を利用している．CCCDATASet の性質上，ハニーポットが 20 分毎に初期化されるため，取り出すセッションデータに関してセッション時間のタイムアウトを 20 分としている．また，正常なデータと異常なデータでは採取時期や取得方法が異なるので，本節ではそれぞれのデータ取得方法に関して説明する．

4.1.1 正常なデータ

正常なデータは我々の研究室にあるサーバを監視することで収集する．採取期間は 2012 年 8 月から 9 月で今回は大学内のネットワークは安全であると仮定している．実験用のデータとして取り出す方法として採取されるデータのポート番号に着目し，IRC 通信は 6667 番を利用しているものを，HTTP 通信は 80 番を利用しているものを抽出した．

表 2: 抽出したユニーク IP アドレス数，セッションデータ数

	Normal	C&C		
		C09	C10	P13
IRC(IP アドレス数)	736	6	19	0
HTTP(IP アドレス数)	763	51	139	15
IRC(セッションデータ数)	903	190	573	0
HTTP(セッションデータ数)	1270	84	255	406

4.1.2 異常なデータ

異常なデータは，C09，C10，P13 に含まれる C&C サーバとボットの通信を指す．これらのデータ・セットはハニーポット上で実行されたマルウェアのデータを収集したものである．それぞれのデータ・セットの採取期間に関して，C09 は 2009 年 3 月 13 日，14 日，C10 は 2010 年 3 月 5 日から 11 日の 7 日間，P13 は 2013 年 5 月 18 日から 25 日の 7 日間である．それらのデータの中にはボットネットの通信が含まれており，我々はこれらの通信から C&C トラフィックを取り出す必要がある．そのために，パケットのペイロード部の情報を精査する．具体的には，クライアントが C&C サーバへ接続する際に用いるコマンドに着目する．一般的に IRC では JOIN コマンドを用いてサーバへログインし，HTTP では GET コマンドを用いてサーバにデータを要求する．我々は，ボットと C&C サーバで同じ状況を想定し，JOIN，GET を使っている通信に関して取り出す．

このようにして採取されたデータに関して表 2 では抽出されたユニーク IP アドレス数セッションデータの数について示している．P13 に関しては上記の条件で抽出

	Normal(IRC)	C&C	
		C09	C10
V_1	$88(6.0 \times 10^3)$	6(24)	5(250)
V_2	$1187(3.6 \times 10^6)$	$67(1.5 \times 10^4)$	$77(1.9 \times 10^4)$
V_3	$75(6.1 \times 10^3)$	2(6.9)	3(632)
V_4	$1336(2.2 \times 10^6)$	$177(1.7 \times 10^5)$	$185(1.6 \times 10^6)$
V_5	$583(2.8 \times 10^5)$	8(75)	6(111)
V_6	1(0)	1(0)	1(0)
V_7	0(0)	0(0)	0(0)

表 3: IRC セッション解析結果の平均値，() 内は分散値

	Normal(HTTP)	C&C		
		C09	C10	P13
V_1	$88(1.5 \times 10^7)$	$60(1.4 \times 10^2)$	$47(1.3 \times 10^4)$	4(5.7)
V_2	$33140(3.9 \times 10^9)$	$194(5.7 \times 10^2)$	$177(2.1 \times 10^9)$	$126(1.4 \times 10^2)$
V_3	$129(1.7 \times 10^6)$	50(900)	$35.4(1.4 \times 10^7)$	3.4(74)
V_4	$33671(2.1 \times 10^{12})$	$66320(1.9 \times 10^9)$	$42212(2.6 \times 10^9)$	$1135(1.1 \times 10^4)$
V_5	$249(1.2 \times 10^5)$	2.6(2.8)	$0.27(1.3 \times 10^4)$	1.7(3.7)
V_6	$9.15(1.3 \times 10^6)$	3.8(0.13)	$35.7(7.8 \times 10^5)$	1.1(0.3)
V_7	$122(1.3 \times 10^5)$	0.64(2.3)	3.1(6.67)	1.5(0.5)

表 4: HTTP セッション解析結果の平均値，() 内は分散値

を行ったが，IRC トラフィックは検出されなかった．

4.2 セッションデータ解析

次にセッションデータの解析を行い，それらを数値列ベクトルとして出力させる．本研究では表 1 に示した特徴ベクトルを利用する．表 3, 表 4 には IRC，HTTP それぞれの解析結果で得られた平均値と分散値について示す．表 3 に関して，C&C に比べて IRC の方が V_1 から V_5 の平均値，分散値が小さいことが分かる．また，C10 の V_1 から V_5 に関して C09 と近い平均値を得ることができたが分散値は大きい事がわかる．IRC セッションの場合は，一回のセッションでサーバへの再接続を行わないため $V_6 = 1, V_7 = 0$ と一意に決まる．表 3 に関して，C&C の方が HTTP よりも V_6 以外の平均値が小さい. また，P13 に関して，V_1 から V_7 の分散値が小さい．C09,C10 に関しては V_4 の分散値が大きい．このことから，受信データサイズは C&C の場合では大きいものがあることが分かる．HTTP は IRC に比べてセッションデータの形態が多様であることも分かる．

次に，定義した特徴ベクトルを用いてセッション解析した結果に関して解析データの管理を簡素化するためにデータの正規化を行う．i 番目のセッションデータに関して，j 番目の属性値を正規化したい場合，次の式で表すことができる．

$$x_{i,j} = (x_{i,j} - min(x_{n,j})/max(x_{m,j}))$$

ここで，j 番目の属性値に関して n 番目のセッションデータで最小値を，m 番目のセッションデータで最大値を取る．これにより，各データの特徴量は最小値が 0，最大値 1 の実数値となる．

4.3 機械学習を用いた分類

本節では機械学習による分類ついて説明する．今回の実験では教師あり学習として用いれている 3 つの識別モデル，SVM，ナイーブベイズ，ロジスティック回帰を用いる．文献 [2] では IRC 型ボットネットでの C&C トラフィックを抽出するために複数の識別モデルでの比較を行うことで SVM の有効性を示している．しかし，今回の実験では IRC,HTTP の両方を扱うことに加えて，3.3 節での C&C セッション分析から HTTP 通信の挙動パターンは IRC に比べて多いことが分かったので，複数の識別モデルを用いた評価が必要となる．これらのモデルは教師あり学習であり，学習データと呼ばれる入出力のペアの事例が複数与えられているデータが必要となる．それを基に，テストデータの新しい入力データに関しても正しい出力ができることを目的としたものが教師あり学習ある．我々は，ボットネット対策のための実用的なシステムとしてどの識別モデルを使うことが適切であるかに関して分類精度と実行時間から評価を行う．機械学習の機能を実現させるために我々は R[12] を利用している．

機械学習のモジュールに関して，SVM では kernlab パッケージ [13]，ロジスティック回帰では glmnet パッケージ [14]，ナイーブベイズ法では e1071 パッケージ [15] を利用している．また，SVM 適応の際のカーネル関数はラジアル基底関数

$$k(\vec{x}, \vec{y}) = exp\frac{-||\vec{x}-\vec{y}||^2}{2\sigma^2}$$

を用い，予備実験により適切と思われる σ を設定している．また，SVM とロジスティック回帰では学習データから適切なチューニングパラメータを決定するために交差検定を行う．ナイーブベイズに関しては，今回の実験ではチューニングするパラメータがなかったため，交差検定を行っていない．

今回の実験で用いた学習データとテストデータに関して説明する．本手法を実際のシステムに取り入れる場合，過去のデータでの学習結果によって新しいデータの分類を行う．このように実用面に則した実験を行うために，C&C セッションのデータに関して表 5 で示す組み合わせで評価する．

表 5: 学習データ・テストデータの組み合わせ

	学習データ	テストデータ
実験-1	C9	C10
実験-2	C9	P13
実験-3	C10	P13
実験-4	C9, C10	P13

5 実験結果と考察

今回の実験では，過去のデータを学習して新たなデータを予測可能かに関する評価を行った．C&C サーバの通信を集めたデータ・セットに関して，採取時期が異なるものを 3 種類用意し，表 5 に従って機械学習を用いた分類実験を行った．機械学習手法の 1 つである SVM を用いた場合の実験結果に関して，表 6 に示す．表 6 の結果から実験-1 が 4.1%で最も誤検知が低いが，検知率は 71.9%と最も低い．また，実験-3 では検知率が 95.1%で最も高いが，誤検知率が 19.1%と最も高い．これらと比較して，実験-4 の結果は検知率，誤検知率の観点から高精度であると言える．今回取得したデータ・セットにはそれぞれの通信パターンがあることが特徴ベクトルを用いたセッション解析で分かっている．実験-4 では，機械学習で学習させる C&C 通信のパターンが多く，通常の通信と区別を明確に行うことができたたために，実験-1，実験-2，実験-3 よりも適切な識別モデルを生成できたと考えられる．

また，図 7 では提案する特徴ベクトル (以下，提案ベクトル) と既存の特徴ベクトル [2]（以下，既存ベクトル）の比較について示す．用いた機械学習は SVM.，ロジスティック回帰，ナイーブベイズでありそれぞれの結果に関する比較も行っている．表 7 の結果から，全ての機械学習手法において，既存ベクトルよりも提案ベクトルの方が検知率が高く，誤検知率が低いことが分かる．また，機械学習手法の性能評価に関して，ロジスティック回帰，ナイーブベイズでは網羅的に C&C セッションを検知するため誤検知が高い．C&C トラフィック自体は攻撃を受ける前の予兆であるため，攻撃を受ける可能性がある場合にアラートを挙げるようなシステムにする場合は攻撃を受けるという信頼性は低くなるが，安全性を重視したい場合には有効であると考えられる．しかし，誤検知率が高くなるに連れて誤ったアラートを発する頻度が増えるので攻撃対策の信頼性やコストなどを考慮すると実用的ではない．故に，SVM を用いた場合に検知率 89.1%．誤検知率 6.8%という結果となっており，最も良い分類精度を得るため実用的であると考えられる．しかし，実際に正常なデータを誤検知したものは 106 個であり，個数としては大きいためさらなるシステムの改良が必要となる．

表 6: 異なる時期のデータセットを利用した実験結果

	検知率	誤検知率
実験-1	71.9	4.1
実験-2	72.8	16.6
実験-3	95.1	19.1
実験-4	89.1	6.8

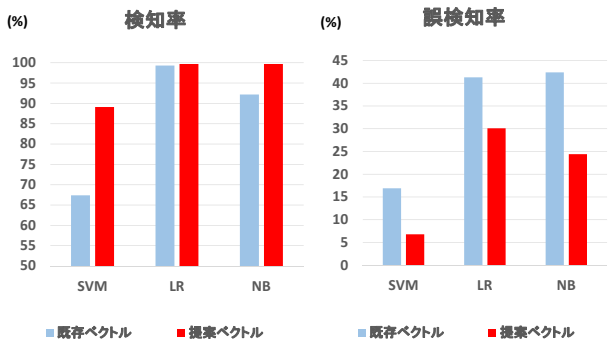

図 7: 提案ベクトル，既存ベクトルとの比較に関する実験結果
(ナイーブベイズ：NB，ロジスティック回帰：LR)

今回の見逃し，誤検知の原因として 7 次元の特徴ベクトルの解析結果を基に考えると，SVM では受信，送信パケット，受信，送信データサイズは通常の通信と比べて小さいデータを C&C 通信と判断している．しかし，正常な通信の中でも SVM のテストデータにおける分類予測で C&C 通信に該当するデータが存在している．これらの C&C 通信と予測される正常データに関して，セッション時間が長く，アクセス時間の標準偏差が大きいものは正常データとして判断されている．一方で，受信，送信パケット，受信，送信データサイズが小さいデータの中でも，アクセス時間，アクセス回数，アクセス時間の標準偏差の値が小さい通常の通信は SVM の予測では C&C 通信と誤検知されている．

表 7 では実験-4 において提案方式を用いた場合の学習データでのモデル予測時間とテストデータの分類時間を示している．結果としてナイーブベイズでの学習データの読み込み時間が最も早く，SVM でのテストデータの分類時間が最も早いことが分かった．SVM，ロジスティック回帰では交差検定法により学習データからパラメータ

チューニングを行ったのでナイーブベイズに比べて時間を要している．

表 7: 機械学習アルゴリズムの実行時間の比較
(ナイーブベイズ：NB，ロジスティック回帰：LR)

	SVM	LR	NB
学習データ (s)	1.97	2.01	0.03
テストデータ (s)	0.12	0.81	0.38
合計 (s)	2.09	2.82	0.41

6 結論

本研究では，多様なプロトコルを利用する C&C サーバを特定するための新たな特徴ベクトルを提案した．この特徴ベクトルはクライアントとサーバの通信を区別することに加えて，クライアントがサーバへアクセスする挙動を表現したものとして定義した．この手法を用いることで C&C サーバが用いるプロトコルに特化せずに C&C トラフィックが検知可能な手法となることを示した．また，機械学習手法の分類精度に関して検知率はロジスティック回帰とナイーブベイズが高いが，誤検知率は SVM が低いことが分かった．また，実行時間に関して，学習データの読み込みはナイーブベイズが最も高速でテストデータの分類は SVM が最も高速であることが分かった．

今後の課題として，今回扱った実データで用いられていない DNS，P2P などのトラフィックでの評価実験を行う．また，実用化に向けて，本実験での一連の流れを自動化するようなシステムの設計を行いたいと考えている．

謝辞

この研究の一部は,「国際連携によるサイバー攻撃の予知技術の研究開発 (総務省)」の支援を受けている．

参考文献

[1] Vania, J. Meniya, A. Jethva, H.B.: A Review on Botnet and Detection Technique, International Journal of Computer Trends and Technology,Volume 4, Issue 1, pp.23-29, (2013)

[2] Kondo, S. and Sato, N.: Botnet Traffic Detection Techniques by C&C Session Classification Using SVM, *Proc.Second International Workshop on Security* , pp.91-104, (2007)

[3] Goebel, J. and Holz, T.: Rishi: Identify bot contaminated hosts by IRC nickname evaluation, *Proc. 1st USENIX HotBots*, (2007)

[4] Narang, P. Ray, S. Hota, C. and Venkatakrishnan, V.: PeerShark-Detecting Peer-to-Peer Botnets by Tracking Conversations, *Proc. IEEE Security & Privacy Workshops* , pp.108-115, (2014)

[5] Ashley, D.: AN ALGORITHM FOR HTTP BOT DETECTION, Research paper, University of Texas - Information Security Office, (2011)

[6] Kumar Tyagi, A. and Nayeem, S.: Detecting HTTP Botnet using Artificial Immune System, *International Journal of Applied Information Systems*, Volume 2, No.6, pp.34-37, (2012)

[7] マルウェア対策研究人材育成ワークショップ 2014(MWS2014)`http://www.iwsec.org/mws/2014/about.html`(accessed 2014-12-05)

[8] 市野将嗣，市田達也，畑田充弘，小松尚久：トラフィ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，Volume 53, No.9 pp2062-2074, （2012）

[9] Gu, G. Perdisci, R. Zhang, J. and Lee, W.: BotSniffer: Detecting botnet command and control channels in network traffic, *Proc. 15th Annual Network and Distributed System Security Symposium* (2008)

[10] Castro, L.N. and Timmis, J.:*Artificial Immune Systems, A New Computational Intelligence Approach*, Springer, (2002).

[11] Gu, G. Perdisci, R. Zhang, J. and Lee, W.: BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection, *Proc. 17th USENIX Security Symposium*, (2008)

[12] R project, `http://www.r-project.org/`(accessed 2014-11-10)

[13] A. Kratzoglou et.al. “kernlab: Kernel-based Machine Learning Lab”, Reference manual, CRAN, November 2013, `http://cran.r-project.org/web/packages/kernlab/kernlab.pdf`(accessed 2014-11-10)

[14] J. Friedman et. al. “glmnet: Lasso and elastic-net regularized generalized linear models”, Reference manual, CRAN, May 2014, `http://cran.r-project.org/web/packages/glmnet/glmnet.pdf`(accessed 2014-11-10)

[15] D. Meyer et. al. “e1071: Misc Functions of the Department of Statistics (e1071), TU Wien”, Reference manual, CRAN, September 2014, `http://cran.r-project.org/web/packages/e1071/e1071.pdf`(accessed 2014-11-10)