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

山内, 一将
九州大学

川本, 淳平
九州大学

堀, 良彰
佐賀大学

櫻井, 幸一
九州大学

出版情報：暗号と情報セキュリティシンポジウム. 2014, pp.4C1-5-, 2014-01-21
バージョン：
権利関係：

SCIS 2014 The 31st Symposium on
Cryptography and Information Security
Kagoshima, Japan, Jan. 21 - 24, 2014
The Institute of Electronics,
Information and Communication Engineers

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
Extracting C&C Traffic by Session Classification Using Machine Learning

山内 一将 * 川本 淳平 * 堀 良彰 † 櫻井 幸一 *

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，遠隔により操作されるマルウェアを使った，ボットネットによる被害が顕著である．ボットネットが脅威となる原因の一つに，Command and Control（C&C）サーバを利用することが挙げられる．C&C サーバは踏み台となるコンピュータであるボットの制御や，命令を行なう．これにより，同時に多数のコンピュータから攻撃させることや，ハーダーの特定を困難にすることを可能にする．C&C サーバに利用されるプロトコルとして，主に Internet Relay Chat（IRC）や，HTTP などがある．従来のボットネットは，IRC サーバを利用したものが主流であったが，近年 HTTP を利用したボットネットが増えており，対策をより困難にしている．本研究では，ボットネットの中でも HTTP 型ボットネットに焦点を置く．そのために，C&C サーバとボットの通信特性の調査を行い，それを基にセッション分類による C&C トラフィック抽出を行なう．具体的には，トラフィック観測により得られたデータをセッション毎に解析し，通常の HTTP セッションと C&C セッションの分類を行なう．HTTP 型ボットネットでは，ボットが外部サーバに攻撃を仕掛けるための命令を得るために，C&C サーバへアクセスを行なう．そこで，セッション解析において，このアクセス挙動の特性を考慮した特徴ベクトルを定義する．また，このシステムによるセッション分類を行なうために，機械学習として，Support Vector Machine(SVM), ナイーブベイズ法, ロジスティック回帰を用いた場合のそれぞれの分類精度の評価を行なう．

キーワード ボットネット，HTTP 通信，機械学習

1 序論

今やインターネットは我々の生活に必要不可欠なものとなっている．コンピュータの性能は年を経る毎に向上し，有用なアプリケーションやクラウドなどの開発が多くなされている．アプリケーションのダウンロードやクラウドの利用はオンライン上で行なう必要があり，インターネットのユーザへの普及率は著しく上昇している．一方で，ユーザのセキュリティに関する意識は低く，それが原因となってマルウェアに感染する状況が多く見られる．セキュリティ意識の低いユーザは，パスワードの使い回しや，セキュリティソフトの導入をしていないことや，セキュリティパッチの更新をしていないことなどの問題を持っている．そのような脆弱性を狙った攻撃によりマルウェアに感染したコンピュータは，個人情報を盗まれることによる銀行口座の情報，クレジットカード番号などの盗聴や，フィッシングサイトへの誘導，E メールのハイジャック等の被害を受ける．マルウェアの感染経路として，メールに添付されたファイルや URL を開くことによる感染や，悪性な Web サイトを閲覧することでマルウェア配布サイトへ誘導されて強制的にマルウェアをダウンロードさせられることによる感染，いわゆるドライブバイダウンロードなどがある．マルウェア感染による他の被害として，遠隔に操作されるマルウェアが脅威となっている．これは，個人に限定されたの被害でなく，他人や社会にも影響をおよぼす被害とされている．これに感染した端末はボットネットに接続することで，攻撃者の操作対象となる．遠隔操作型のマルウェアは，主にバックドア型のトロイの木馬が用いられ，ボット化したコンピュータは，被害者に感染を秘匿して攻撃者に操作される．そのため，被害者は自分のコンピュータが犯罪の片棒を担いでいることに気づきにくく，深刻な問題となっている．

マルウェアの一般的な対策技術として，パターンマッチ手法がある．この手法は，侵入検知システムである Intrusion Detection System(IDS) や，多くのユーザが

* 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２２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, /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２２号 福岡 SRP センタービル７階

導入しているアンチウイルスソフトなどに組み込まれている．具体的には，既存のマルウェア通信のデータからパターンマッチングルールをデータベース化させたものをベンダーが用意し，これにマッチしたものを悪性として検知する手法である．この方式は従来用いられている信頼性の高い手法であるが，検体を入手していない未知のマルウェアに関してはデータベースに登録されていないため，検知ができないという問題点がある．これに対し，未知のマルウェアに対しても検出できる手法としてヒューリスティック手法があり，その手法は多岐にわたっている．例えば，ヒューリスティック手法の一つに動的ヒューリスティック法がある．具体的には，不審なプログラムを探して，サンドボックスのような仮想化環境でそのプログラムを実行させ，マルウェアかどうかを判定する．ただし，通常のソフトウェアも誤って検出してしまう誤検知率の高さが問題として挙げられる．また，静的ヒューリスティックでは，マルウェアのコードを分析して，そのプログラムが行なう動作を抽出することで，マルウェアの挙動と類似しているか判断を行なう．動的解析とは異なり，実際にマルウェアを動かす必要がないのでコストはかからないが，プログラム部分を暗号化するマルウェアに関してはコードを直接確認できないので検知が難しい．本研究では，遠隔操作型マルウェアを使ったボットネットに焦点を置き，未知のボットや，通信の暗号化を行なうボット検知が可能で，高い検出精度を実現できる手法について説明する．

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できた，複数のボットによって構成される．ボットネットは，数 10 台から数 100 万台のボットによって構成される．脅威とされている点として，攻撃者が Command & Control サーバ (C&C サーバ) を使うことにある．C&C サーバは，ボットネットの制御の中心となるサーバであり，攻撃者から受けた命令を同時に多数のボットへ転送することができる．これによって，命令を受け取ったボットがは，協調して標的のサーバへ攻撃を行なう．標的とされたサーバは大量のパケットを受け取ることで負荷がかかり，クライアントへのサービス提供を不能にする．これは DDoS(Distributed Denial of Service) 攻撃と呼ばれ，ボットネットが行なう主な攻撃の一つである．その他にも，スパムメールの送信や，ポートスキャン攻撃などがある．また，攻撃者は，C&C サーバやプロキシを中継させることにより，攻撃元の特定を困難にする．

2.1 ボットネット対策手法

ボットネットの対策技術として用いられている一般的な手法について紹介する．この節では，ハニーポットを用いた検知手法，IDS を用いた検知手法，DNS ベースの検知手法について述べる．

2.1.1 ハニーポットを用いた検知手法

ハニーポットとは，意図的に脆弱性を持たせることで，攻撃者に故意に不正アクセスさせ，その手法分析したり，重要なシステムに侵入させないようにするための囮のコンピュータである．Freiling らはボットネットの脅威を解決するためにハニーポットを利用している [1]．また，Nepenthes[2] はマルウェアサンプルを自動的に収集するための特別なツールである．ハニーポットを用いた手法は，ボットの挙動を解析するためには有効な手段であるが，ボットを収集することや，攻撃者にハニーポットであることを検知されないように慎重に運用するための技術が必要となる．

2.1.2 IDS を用いた検知手法

IDS は不正アクセスの兆候を検知し，ネットワーク管理者に通報する機能を持つソフトウェアである．検知方式として，シグネチャ型，異常検知型の 2 つに分けられる．

- シグネチャ型

  シグネチャ型ではパケットのペイロードに埋め込まれているルールを登録することで，次に同種のボットが通信してきた場合に検知出来るという手法である．Snort[3] は，この手法を用いたオープンソースの侵入検知ツールとして知られている．Goebel らは IRC ボットのニックネームを調べて検知を行っている [4].

- 異常検知型

  この手法は，通信の大きな遅延や，大量のトラフィック，不審なポートへの通信，不審なシステムの挙動のようなネットワークトラフィックの異常によりボットネットを検知する手法である．この手法では未知のボットに対しても検知可能である．また，この手法はホストベースとネットワークベースの 2 つに分けられる．

  ホストベースでは，コンピュータ内部の解析を行う．具体的には，ホスト上の OS やアプロケーションログなどを監視する．しかし，この手法は誤検知率が高くなるため限界がある．

  ネットワークベースでは，ネットワークに流れるパケットを監視する．さらに，この手法はネットワーク全体の監視，監視対象のホスト自身で送受信するパケットのみの監視の 2 つに分ける事がで

きる．Gu らは内部ネットワークに潜む，ボットに感染したクライアントを識別するために，ネットワークベースの異常検知システムである Botsniffer を提案している [5].

2.1.3 DNS ベースの検知手法

DNS ベースの検知手法では，DNS サーバのトラフィックを監視して，異常を発見する．攻撃者によって操作されるボットは，C&C サーバの場所を特定するために DNS サーバへ問い合わせを行なう．そのため，同一ドメインの解決を行っていれば，それらを協調型攻撃として検知することができる．Dagon らは DNS サーバに定期的にアクセスを行なうボットネットを検知する手法を提案している [6]．HTTP 型ボットネットには，しかし，DNS サーバに問い合わせを行わずに直接 C&C サーバに接続するものも出現しており，それらに関しては検知できない．

2.2 ボットネットが利用するプロトコル

ボットネットが C&C サーバとの通信に用いるプロトコルとして，IRC，HTTP を挙げる．

2.2.1 IRC 型ボットネット

IRC 型ボットネットは 1990 年頃から出現しており，インターネット用のチャットサーバをボットネットの制御サーバとして利用したものである．攻撃者が IRC サーバを制御する場合，同じチャネルにいるユーザをボットとして利用する．チャネル内にいるボットは C&C サーバが命令を送るまで待ち，命令が来ると攻撃を行なう．図 1 に IRC 型ボットネットの構造について示す．この場合，ボットが一度 C&C サーバへ接続すると，接続終了まで C&C サーバが一方的に命令をボットへ送信するため，PUSH 型の構造を取る．

2.2.2 HTTP 型ボットネット

HTTP 型ボットネットは 2005 年頃から出現しており，Web サーバをボットネットの制御サーバとして利用したものである．HTTP 型ボットは C&C サーバから命令を受け取るときに HTTP リクエストを送信することで C&C サーバへ接続し，命令を受け取る．図 2 に HTTP 型ボットネットの構造について示す．この場合，ボットが C&C サーバに接続すると，接続終了までボットが HTTP リクエストを送信することにより C&C サーバが命令を送るため，PULL 型の構造を取る．HTTP 型ボットネットは，正規の HTTP 通信に似せた通信を行うために，IDS 検知や IPS での検知が非常に難しい．

3 既存研究

本研究では機械学習を用いたボットネット検知手法について述べるが，これに関連した既存研究がある．この

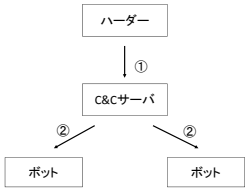

図 1: IRC 型ボットネット

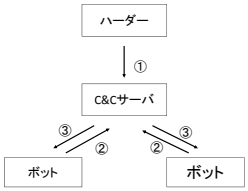

図 2: HTTP 型ボットネット

章では，それらの既存研究を紹介する．また，それらの既存研究の限界について述べる．

3.1 SVM を用いた IRC 型ボットネット検知手法

田中らは，IRC プロトコルを使ったボットネットを検知するために，IRC サーバにアクセスするトラフィックを監視して，C&C トラフィックを検出する手法を提案している [7]．具体的には，トラフィックデータをセッション毎に分割し，通常の IRC セッションと，悪性の C &C セッションに関してセッション分類を行っている．セッション分類を行なうために，C&C セッションの特性の調査を行い，それを基に特徴ベクトルを定義して特徴抽出を行う．特徴抽出のための要素は，セッションデータから得られる情報を用いて，送信パケット数，受信パケット数，送信データサイズ，受信データサイズ，セッション時間で構成されている．その特徴抽出の結果を SVM に入力し，分類精度の評価を行っている．SVM はパターン認識アルゴリズムの 1 つであり，2 クラス分類に関しては，次元数が増えても分類精度が落ちないことが利点である．SVM の有用性を示すために，著者らはナイーブベイズ法， k-nn 法との分類精度の比較も行っている．問題点として，特徴抽出が IRC 型ボットネットに特化したものであり，その他のプロトコルや，構造を持ったボットネットに対しては検知できないことが挙げられる．

3.2 HTTP 型ボットネット検知アルゴリズムの提案

Ashley は HTTP 型ボットネットを検出するための効率的なアルゴリズムの提案を行っている [8]．HTTP 型ボットは攻撃を行うために C&C に命令を要求する動作を行う．その命令を得る動作には周期性があると仮定し，通信時間間隔における標準偏差が，通常の通信に比べて小さくなることを利用している．ただし，ボットがネットワークを遮断や，何らかの理由で C&C サーバから命令を受け取ることができなかった場合，通信時間間隔の周期性に反する時間が現れることがある．そこで，通信時間間隔をクラスタリングすることでボット検知を行なう．クラスタリング手法として，著者らは K-means 法を改良したアルゴリズムを提案し，計算速度の向上を実

現させている．この研究では，実ネットワーク上で採取したトラフィックを用いていないため，ボットが C&C サーバにアクセスする挙動の調査を行っていない．そのため，周期的に HTTP リクエストを受け取らない C&C サーバに関しては検知ができない．

4　本研究の成果

ボットネットの中でも，HTTP を利用したボットネットを検知するために行った研究の成果として HTTP 型ボットネットを検知するための C&C サーバとボットの挙動調査，それらの挙動に基づくトラフィックデータの特徴抽出，機械学習を用いたセッション分類を述べる．この章では，それぞれの成果に関して詳しく説明をする．初めに，HTTP プロトコルを利用する C&C サーバとボットの通信の挙動調査を行った．ボットネットが用いるプロトコルとして，従来 IRC が用いられてきた．そのため，IRC に関する先行研究は多くあるのに対し，HTTP を利用したボットネットに関する先行研究は少ない．そのため，HTTP を利用したボットネットの C&C サーバの挙動調査を行った．図 3，図 4，図 5，図 6 では，各セッションデータにおける解析結果をグラフ化したものを示す．

図 3 では，各セッションデータの送信パケット数の合計，送信データサイズの合計の結果を表している．送信データサイズに関しては片対数グラフで表している．このグラフから，HTTP セッションは IRC セッションと比べてパケット数，データサイズが大きいことが分かる．通常の通信では，クライアントは Web サーバの中の閲覧したいページの URL にアクセスして，HTTP リクエストを送ることでデータを受信する．また，ボットの場合は C&C サーバから命令を受け取るために HTTP リクエストを送る．従って，ボットネットトラフィックの中で，HTTP リクエストを送っている通信を収集した．C&C セッションの中でもパケット数，データサイズが大きいデータもあるが，これは C&C サーバがネットワークエラーによりデータを再送しているか，または通常の通信に似せるためにダミーデータを送っていると考えられる．

図 4 では，各セッションデータの受信パケット数の合計，受信データサイズの合計の結果を表している．受信データサイズに関しては片対数グラフで表している．通常の通信では，クライアントは Web サーバから閲覧したいページのテキストデータや，画像データなどをダウンロードするため，コマンドの受け取りのみ行う C&C セッションと比べると，パケット数，データサイズが多いことが分かる．

図 5 では，各セッションデータのセッション時間を表している．本研究では 20 分をセッションタイムアウトとしている．このグラフから，セッション時間は HTTP セッションの方が全体的に長いことが分かる．

図 6 では，各セッションデータのアクセス回数，アクセス時間間隔の標準偏差を示している．通常の通信では，クライアントがアクセスしたい時に，クライアントの意思でアクセスを行なうため，アクセス時間にランダム性があるが，ボットが C&C サーバにアクセスする時は，実行したマルウェアのコードに従ってアクセスを行なうため，定期的な命令の受け取りを行なうものが多いと考えられる．このグラフからも分かるように，HTTP セッションでのアクセス回数は C&C セッションより全体的に多く，標準偏差は HTTP セッションより小さくなる．

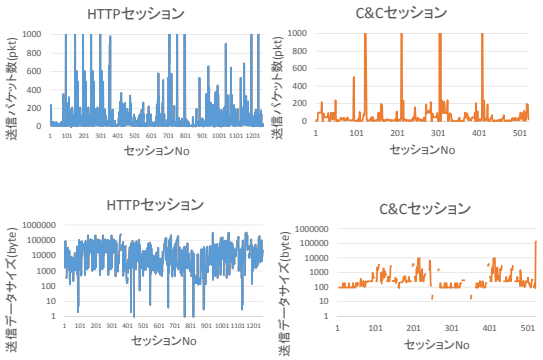

図 3: 送信パケット数，送信データサイズ

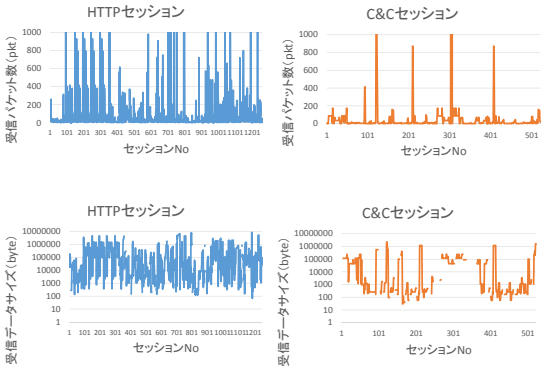

図 4: 受信パケット数，受信データサイズ

また，表 1 では，全てのグラフの結果に関して，各要素に対する全データの平均値を示す．

次にこれらの予備実験を基に，セッションデータの特徴抽出を行った．トラフィックデータから特徴的な要素

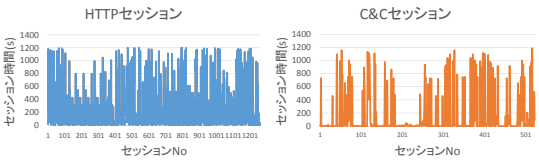

図 5: セッション時間の比較

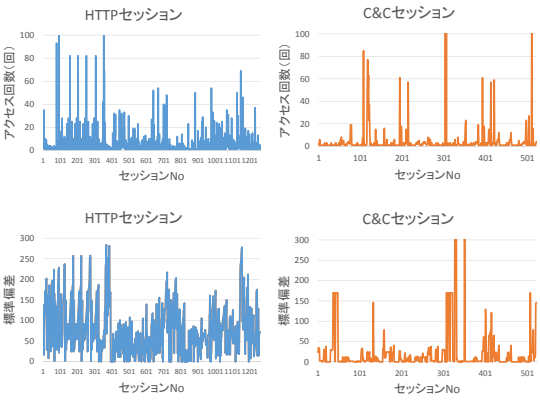

図 6: アクセス回数及びアクセス時間の標準偏差の比較

を探し，抽出する．HTTP を利用したボットネットの場合，通常の通信に似せた挙動を行なうため，検知が難しい．4.1 節で示したグラフからも分かるように，個々の要素のみでは，高い精度でのセッション分類が実現できない．そのため，抽出するべき要素を複数用意する必要がある．そこで，本研究では，抽出するべき要素を組みわせて特徴ベクトルを定義する．具体的な要素として，4. 章の結果を基に，送信パケット数，送信データサイズ，受信パケット数，受信データサイズ，セッション時間，アクセス回数，アクセス時間標準偏差とする．

最後に機械学習を用いたセッション分類を行った．特徴ベクトルによって出力された多次元の数値ベクトルを読み込ませるために，多次元ベクトルにも対応できる様な分類アルゴリズムを選択した．本研究では SVM，ナイーブベイズ，ロジスティック回帰の 3 つの識別モデルを用いて分類を行い，分類精度の評価，実行時間の比較を行った．

表 1: セッション解析結果の平均値

	HTTP セッション	C&C セッション
送信パケット数の合計 (pkt)	88	47
送信データサイズの合計 (byte)	33140	1060
受信パケット数の合計 (pkt)	129	165
受信データサイズの合計 (byte)	33671	2070
セッション時間 (s)	249	154
アクセス回数 (回)	25.72	15.43
アクセス時間間隔	122	21.2

SVM では，訓練データサンプルから，各データ点との距離が最大となるマージン最大化超平面を求める．この学習モデルは，高次元ベクトルであっても高精度で分類できる．なぜなら，本研究では，クラス間の境界面を求めるための計算をカーネルトリックの一つである Radial Basis Function(RBF) を用いて行なうことで非線形に拡張した SVM を使っているからである．

ナイーブベイズは単純な確率的分類器である．2 クラスの分類自体の同時確率分布を完全にモデリングして，どちらのクラスに分類されるかの確率を算出する．そして，事後確率が最も高いクラスに分類する．事後確率の算出方法はベイズの定理に基づいており，式は次のように表すことができる．

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

事象 B に関して，P(B—A) は事象 A が起きた後での事象 B の確率 (事後確率)，P(B) は事象 A が起きる前の事象 B の確率（事前確率）となる．

ロジスティック回帰では，訓練データの 1 または 0 と表記されたラベルにより分類を行い，事後確率をモデル化する．このモデルでは，テストデータに対して予測値を算出させる．その式は以下の様になる．

$$\hat{y} = \beta_0 + \beta_1 x_0 + \beta_2 x_1 \cdots\cdots$$

x_i には i 番目の属性に含まれる値が入る．この場合 y の値連続変数となるため，0 または 1 の値になるとは限らない．そこで，y の範囲を抑えるために，対数変換することでどちらのクラスに属するかを推定している．

5 実験

C&C サーバを特定するために，HTTP トラフィックデータから C&C セッションを抽出するための実験についての説明を行う． 図 8 では，今回行った実験の流れを示している．まず，始めに HTTP 通信を行っているトラフィックデータを収集する．収集の対象となる通信が用

いるプロトコルは HTTP1.1 である．HTTP トラフィックのデータは私が所属する研究室のサーバを監視することで集めている．C&C トラフィックデータは，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ている．次に，これらのトラフィックデータをセッション毎に分割する．HTTP 型ボットネットでは，一回のセッションでサーバとのコネクションの確立を複数回行なう機構であるため，送信元 IP アドレスと，宛先 IP アドレスが同じトラフィックを集約した．表 2 では，HTTP セッションと，C&C セッションのそれぞれの収集したクライアントとサーバのユニーク IP アドレス数について示している．HTTP セッションでは 763 個，C&C190 個の IP アドレスを取得した．

表 2: ユニーク IP アドレス数

	ユニーク IP アドレス数
HTTP セッション	763
C&C セッション	190

次に，それぞれの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．特徴ベクトルとしては 4 章で定義したものを用いる．また，分類結果の比較を行なうために，既存研究 [7] で定義された特徴ベクトルも適応した．図 7 には既存の特徴ベクトルと提案する特徴ベクトルを定式化したものを示す．

最後に機械学習による分類を行なう．本研究では教師

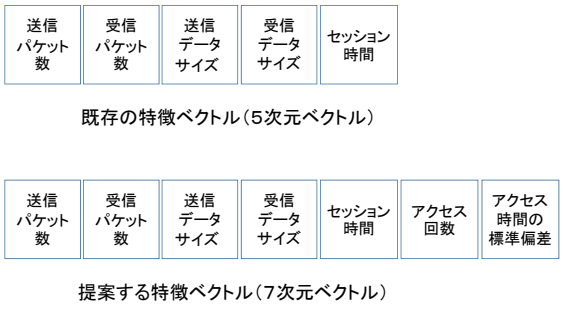

図 7: 定義した特徴ベクトル

あり学習として使われている 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た．これらのモデルは教師あり学習であり，訓練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である．

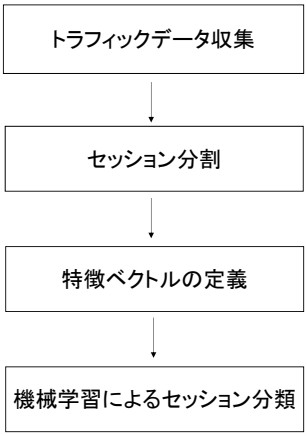

図 8: 実験の流れ

表 3 では，本実験で取得したセッションデータ数について示す．訓練データは，HTTP セッションが 999 個，C&C セッションが 297 個の計 1296 個，テストデータは，通常のセッションが 263 個，C&C 個が 122 個の計 385 個で行った．

表 3: セッション数

	HTTP セッション	C&C セッション	合計
学習データ	999	297	1296
テストデータ	263	122	385

6 実験結果

図 9，図 10 はセッション分類の結果を示している．これらのグラフでは，既存研究 [?] で定義された特徴ベクトルと，本研究で利用した特徴ベクトルとの比較を各アルゴリズムで行ったときの結果を表している．その中でも，図 9 では，Precision（適合率），Recall（再現率）を，図 10 では，Accuracy（正確度），F-measure（F 値）の結果を表している．適合率は，正常と予測したデータのうち，実際に正常であるものの割合であり，

$$Precision = \frac{TP}{TP+FP}$$(TP:True Positive，FP:False Positive（誤検知率））

とする．再現率は，実際に正常であるデータのうち，正常であると予測されたものの割合であり，

$Recall = \frac{TP}{TP+FN}$(FN:False Negative（見逃し率）)

とする．正確度は予測した結果の正解率である．F 値は適合率と再現率の調和平均であり，

$$F - measure = \frac{2*Recall*Precision}{Recall+Precision}$$

とする．結果として，既存の特徴ベクトルより，提案した特徴ベクトルの方が良い結果を得る事が分かる．また，SVM で分類したの場合，正確度，再現率，F 値が高く，ロジスティック回帰で分類した場合，正確度，適合率，F 値が高いことも分かる．詳しい考察は次の章で行なう．

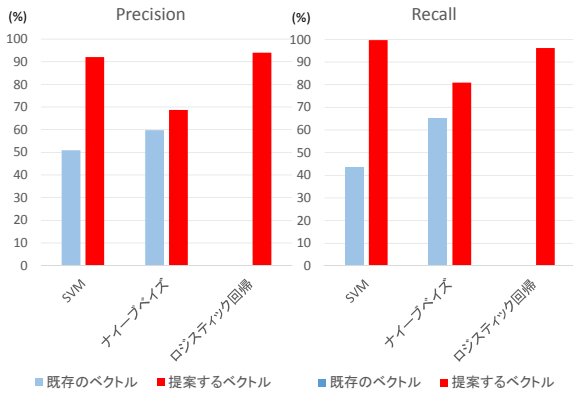

図 9: 適合率，再現率

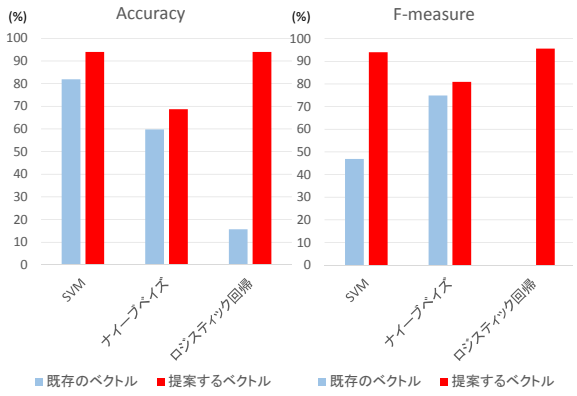

図 10: 正確度，F 値

表 4 は各機械学習で分類行った時にかかった実行時間を示している．この表から，ロジスティック回帰での分類は学習，予測のどちらにも時間がかかることが分かる．また，既存の 5 次元で構成されたベクトルと提案の 7 次元で構成されたベクトルを比べた場合，提案するベクトルの方が時間かかることも分かる．

表 4: 実行時間

	SVM	ナイーブベイズ	ロジスティック回帰
既存のベクトル			
訓練データ (s)	0.77	0.22	1.07
テストデータ (s)	0.30	0.47	0.85
合計 (s)	1.07	0.69	1.92
提案するベクトル			
訓練データ (s)	0.85	0.27	1.37
テストデータ (s)	0.36	0.54	1.25
合計 (s)	1.21	0.81	2.62

7 考察

特徴ベクトルに関して，我々が定義したアクセス挙動特性を考慮した特徴ベクトルの方が高い分類精度を得る事ができた．提案する特徴ベクトルは，既存研究 [7] で定義された 5 次元の特徴ベクトルに，2 次元のアクセス挙動特性を加えたものとなっている．従って，追加した 2 次元の要素が精度向上に大きく関連していることが分かる．また，用いた機械学習に関して，SVM とロジスティック回帰では精度に関して良い結果が得られた．さらに，適合率，再現率の結果より，2 つのアルゴリズムを比べた場合に，ロジスティック回帰では誤検知率が低く，SVM では見逃し率が低いことが分かった．ナイーブベイズは，実装が簡単で，学習に要する時間が短いが，分類精度が良くない．ナイーブベイズはある事象の出現頻度に基づいた分類であるため，本研究での特徴ベクトルの分類には適していなかったと考えられる．実行時間に関しては，ベクトルの次元数が低い既存のベクトルの方が短いことが分かった．また，今後の実験として，より高次元のベクトルを扱う場合についても実験する必要がある．

8 結論

本研究では，ボットネットが利用するプロトコルの中でも，HTTP を利用したものに着目した．そして，C&C サーバの挙動特性をトラッフィク観測して，調査を行った．調査の結果，ボットは C&C サーバから命令を取得する毎に，1 セッションの中で C&C サーバへアクセスしていることが分かった．また，通常の通信に比べて，マルウェアを実行することによる C&C サーバへの機械的な通信の方がデータ量も少ない．それらの調査を基に，特徴ベクトルを定義してセッションデータの特徴抽出を行った．そして機械学習を用いてセッション分類を行った．機械学習では，SVM，ナイーブベイズ，ロジスティッ

ク回帰を使い，分類精度，実行時間の観点から評価を行い，SVM を用いる時に良い結果が得られることが分かった．今後の課題として，ボットネットのプロトコルや構造に依存しない対策手法を提案することである．また，実環境に適応させるために C&C トラフィック検出システムの設計を行いたい．

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] F. Freiling, T. Holz, and G. Wicherski. “Botnet Tracking: Exploring a Root-cause Methodology to Prevent Denial of Service Attacks. ” In Proceedings of 10th European Symposium on Research in Computer Security (ESORICS’05), 2005.

[2] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling. “The nepenthes platform: An efficient approach to collect malware. ” In Proceedings of International Symposium on Recent Advances in Intrusion Detection (RAID’06), Hamburg, September 2006.

[3] M. Roesch. “Snort - lightweight intrusion detection for networks. ” In Proceedings of USENIX LISA’99, 1999.

[4] J. Goebel and T. Holz. “Rishi: Identify bot contaminated hosts by irc nickname evaluation. ” In Proceedings of USENIX HotBots’07, 2007.

[5] G. Gu, J. Zhang, and W. Lee. “BotSniffer: Detecting botnet command and control channels in network traffic. ” In Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS’08), 2008.

[6] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using timezones. In Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS’06), January 2006.

[7] Satoshi Kondo, Naoshi Sato, “Botnet Traffic Detection Techniques by C&C Session Classification Using SVM”, *IWSEC 2007, LNCS 4752*, pp.91-104, 2007

[8] Daryl Ashley, “AN ALGORITHM FOR HTTP BOT DETECTION” *Daryl Ashley Senior Network Security Analyst University of Texas at Austin - Information Security Office ashley@infosec.utexas.edu*,January 12,2011