トラフィック特性による正当なユーザとDDoS攻撃の識別手法の提案

臼崎　翔太郎 [a)]・有川　佑樹 [a)]・山場　久昭 [b)]・油田　健太郎 [c)]・岡崎　直宣 [d)]

Introduction of Traffic Characteristics to Distinguish Legitimate User Traffic from DDoS Attack Traffic

Shotaro USUZAKI, Yuki ARIKAWA, Hisaaki YAMABA, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

DDoS attack is a serious threat in the current information society where the Internet plays an important role as social infrastructure. Since this attack transmits data so that there is no difference in behaviors from legitimate users, it is difficult to distinguish the user from the attack traffic. Therefore, legitimate users cannot receive the service when their traffic are erroneously detected as the attack. We had previously proposed a system that guarantees continuous service use of legitimate users by introducing a quarantine server apart from the web server that performs ordinary web services. The quarantine server has a function of identifying legitimate users and attacks from the access detected as the attack by the IDS or the firewall. Our previous method finds the legitimate user by extracting feature from the access log after the communication is finished. In other words, this method performs the analyzing after the service is over. Therefore, the previous method is not suitable for continuous service of legitimate users. In this study, we propose a new method that can distinguish between legitimate users and attacks even if the services running. As a result of the experiment, we confirmed that the proposed method can distinguish between legitimate users from attacks.

Keywords: DDoS attack, distinction between human and attack, traffic characteristic

1. 研究背景・目的

DDoS(Distributed Denial of Service) 攻撃は標的のサーバに対して、数百から数千、ときには数万台のホストからサーバの処理能力を上回る大量のパケットを送信し、サービスを機能不全に陥らせるサイバー攻撃の一種である。この攻撃は正当なユーザと挙動差がないようにデータを送信するため、正当なユーザのトラフィックと識別するのが難しい。そのためファイアウォールや IDS(Intrusion Detection System)[1] を用いてトラフィック量で判定を行う場合、正当なユーザも攻撃であると誤検知されてサービスを利用できなくなってしまう。この攻撃を防ぐために様々な研究が行われている [2][3][4] が解決には至っていない。Web サービスが社会基盤としての重要な役割を担う現代において、この攻撃は大きな脅威である。

そこで我々は以前、DDoS 攻撃の中でも HTTP-GET Flood 攻撃を対象に DDoS 攻撃緩和システムを提案した [5][6]。このシステムは通常の Web サービスを行う Web サーバとは別に、検疫サーバを導入することで正当なユーザの継続的なサービス利用を担保することを目的としている。検疫サーバは IDS やファイアウォールが攻撃だと検知したアクセスに対してサービスを提供しつつサービス利用時の一連のアクセスの特徴から正当なユーザと攻撃を識別する機能を持つ。

しかし、以前の検疫サーバの識別手法はサービス運用中に識別を行うことができない。以前の識別手法はセッションと呼ばれるユーザがサービスを利用し始めてから終了するまでの一連のアクセスを利用する。識別の際は通信終了後のアクセスログからセッションを抜き出し、セッションごとにその特徴から正当なユーザと攻撃の識別を行う。つまり、この識別手法はユーザのサービス利用終了後に事後解析を行うことしかできない。そのため以前の手法は正当なユーザの継続的なサービス利用の実現に適切ではない。

そこで本研究では、検疫サーバにアクセスが行われる度にユーザに対して点数を加減算することでサービス運用中に識別を行える以前の手法より即応性の高い手法を提案する。また、提案手法がサービス運用中に識別を行うことが可能な以前の手法より即応性の高い手法であり、正当なユーザの継続的なサービス利用の実現に適切であるか検証する。

2. DDoS 攻撃

DDoS 攻撃は、ネットワーク上の大量のコンピュータが標的のサーバに一斉にデータを送信することで大きな負荷をかけ、機能停止などに追い込む攻撃である。掲示板などで参加者を募って攻撃が実行される場合と、攻撃者がボットウイルスに感染しているネットワーク (ボットネットワーク) 上のコンピュータに命令を出すことで攻撃が実行される場合 (図 1) がある。

現在、この DDoS 攻撃による被害を防ぐため様々な研究や対策が行われているが解決するには至っていない。主な理由

[a)] 工学専攻機械・情報系コース大学院生
[b)] 情報システム工学科助教
[c)] 情報システム工学科准教授
[d)] 情報システム工学科教授

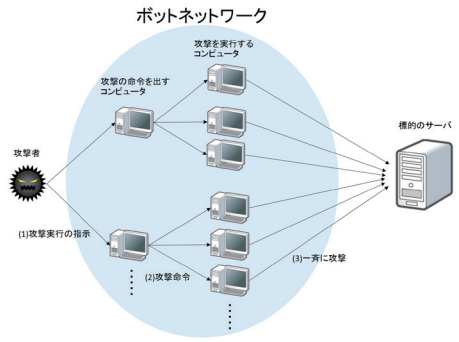


図 1. ボットネットワークによる DDoS 攻撃実行の例

として DDoS 攻撃と通常アクセスの見分けがつきにくいことが挙げられる。例えばサービス利用者が web ページを更新する際には HTTP-GET リクエストをサーバに送信するが、これが攻撃を行うために送信されている HTTP-GET リクエストだったとしてもそれを見破ることは難しい。このため攻撃者のパケットを破棄することができずにサーバの機能が停止したり、誤って正当なユーザのパケットを破棄してしまうということが起こる。

以上のように、防御が難しくサーバを機能停止に追い込む可能性のある DDoS 攻撃はインターネットが社会基盤を担う現代において大きな脅威といえる。

2.1 DDoS 攻撃の種類

DDoS 攻撃には多くの種類が存在する [7] が、攻撃対象に着目すると以下の種類に分類できる [8]。

- 回線帯域への攻撃
  大量のトラフィックを発生させて回線帯域を埋め尽くす攻撃。Smurf 攻撃や TCP SYN Flood 攻撃、DNS Amp 攻撃などがある。
- Web サーバへの攻撃
  サーバの資源を大量に使用することで資源の枯渇を誘発する攻撃。TCP Connection Flood 攻撃や HTTP-Get Flood 攻撃、Slow HTTP DoS 攻撃 [9][10][11] などがある。
- ルータやサーバの脆弱性への攻撃
  ルータやサーバの脆弱性を利用してネットワーク通信機能の停止や再起動を引き起こす攻撃。Teardrop 攻撃や Ping of Death 攻撃、Land 攻撃などがある。

本研究では Web サーバへの攻撃の一種である HTTP-Get Flood 攻撃を対象に防御策を提案する。

2.2 HTTP-Flood 攻撃

HTTP-GET Flood 攻撃は、HTTP-GET リクエストを標的のサーバへ大量に送りつける攻撃である。図 2 に HTTP-GET Flood 攻撃の概要を示す。攻撃を受けたサーバには大量の応答処理が発生し、パフォーマンスが低下したり、機能が停止したりする。HTTP-GET Flood 攻撃の個々のアクセスは HTTP プロトコルに準拠しているため、正当なユーザのアクセスと区別するのが難しい。そのため様々な対策の研究が行われているが、解決には至っていない。

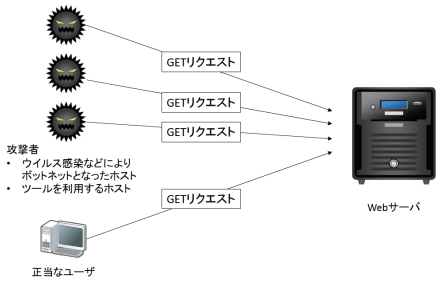


図 2. HTTP-GET Flood 攻撃の概要図

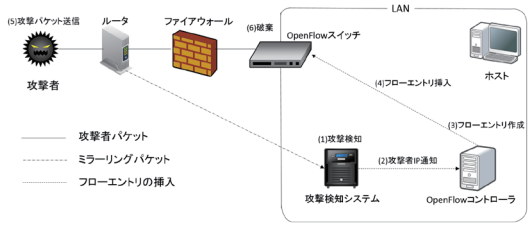


図 3. OpenFlow を用いた攻撃者遮断システムの構成図

3. 既存の緩和策と問題点

3.1 OpenFlow を用いた攻撃者遮断システム

OpenFlow を用いた攻撃者遮断手法 [12] は IDS と OpenFlow コントローラを連携させ、動的なフローエントリを OpenFlow スイッチに登録することで攻撃者の通信を遮断する緩和策であり、実験によりサーバの応答率や応答時間の改善に有効であることが確認されている。この既存手法の処理手順を図 3 に示す。

(1) IDS が攻撃者を検知する。
(2) IDS が攻撃者 IP アドレスを OpenFlow コントローラに通知する。
(3) OpenFlow コントローラが、通知された攻撃者 IP アドレスのフローを破棄するフローエントリを作成する。
(4) OpenFlow コントローラが OpenFlow スイッチにフローエントリを挿入する。
(5) 攻撃者が攻撃パケットを送信する。
(6) OpenFlow スイッチが攻撃者からのフローを破棄する。

既存手法は DDoS 攻撃と判定された IP アドレスのパケットを破棄するフローエントリを作成して攻撃パケットを排除する方法である。しかし DDoS 攻撃を IDS によってトラフィック量で判定を行う場合、正規ユーザの IP アドレスもパケット破棄の対象となってしまい、正規ユーザがサービスを利用できなくなる恐れがある。この問題を改善するためには、判定技術の向上ももちろんあるが、IDS が攻撃者と正当なユーザを正確に判定できない場合でもサービスのパフォーマンスが劣化せず、誤検知率が下がるような検討が必要である。

3.2 DNS を用いた DDoS 攻撃回避システム

DNS を用いた DDoS 攻撃回避システム [13] は Web サーバの IP アドレスを変更する方法を用いて、正規ユーザに影響を与えず DDoS 攻撃による被害を緩和するシステムである。攻撃ホストによる DDoS 攻撃を行い、同時に正規ユーザを模して Web サーバにアクセスし、その際のアクセス成功率でシス

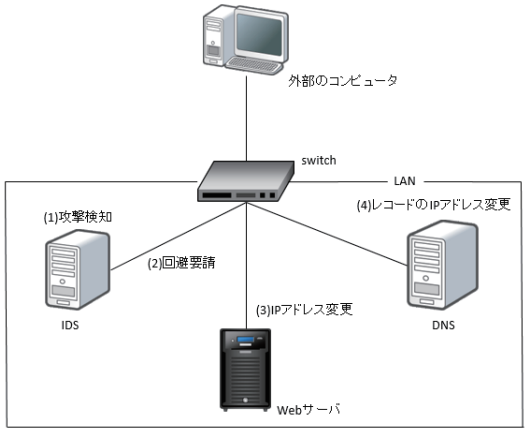


図 4. DNS を用いた DDoS 攻撃回避システムの構成図

テムの有効性を評価した結果、DDoS 攻撃に対しシステムで有効であることが確認されている。この既存手法の処理手順を図 4 を用いて説明する。

(1) IDS サーバが攻撃を検知する。

(2) DNS サーバ及び Web サーバに回避先 IP アドレスを報告し、回避要請をする。

(3) 要請を受けたサーバは IP アドレスの変更を行い DDoS 攻撃を回避する。

(4) IDS サーバからの要請に従い DNS サーバは A レコードに書かれている Web サーバの IP アドレスの値を変更する。

攻撃ホストは攻撃を開始する前に攻撃先 Web サーバの IP アドレスを取得するために Web サーバに接続し、URL から IP アドレスを取得する。そして取得した IP アドレスをもとに DDoS 攻撃を開始する。しかし攻撃を IDS が検知し、Web サーバと DNS サーバに DDoS 攻撃回避を要請するため Web サーバの IP アドレスが変わり攻撃は失敗する。攻撃ホストによっては変更された IP アドレスを再度取得して攻撃を再開することもあるが、IP アドレスが再び変更されることで攻撃が失敗する。これに対し正規ユーザは毎回 DNS サーバに接続して IP アドレスを取得するため、Web サーバへのアクセスが成功する。以上のように正規のユーザと攻撃ホストの動作の違いを利用することで、DDoS 攻撃を緩和しつつ正規ユーザにサービスを提供できる。しかし、この手法では変更された IP アドレスを追跡し続ける DDoS 攻撃は緩和することはできない。また、IDS の検知ルールに当てはまる攻撃パケットが一度でも通過すると具体的被害が発生していない時点で IP アドレスが変更されるため、IP アドレスの切り替え時間が全体的に増加して正規ユーザがサーバにアクセスできなくなる時間が増加してしまう。よってサービスを行うサーバの IP アドレスを変更せずに DDoS 攻撃を緩和できる対策が必要となる。

4. DDoS 攻撃緩和システムの概要

既存の緩和策の問題点を解決するために我々が提案したのが DDoS 攻撃緩和システムである。このシステムは通常の Web サービスを行う Web サーバとは別に、検疫サーバを導入する。既存の手法では IDS で DDoS 攻撃を検知すると正当なユーザ

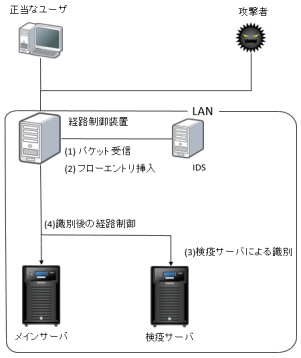


図 5. DDoS 攻撃緩和システム

まで Web サービスを利用できなくなる問題がある。しかし、DDoS 攻撃緩和システムでは一度攻撃と判定されたユーザに対して検疫サーバでサービスを行いながら正当なユーザを識別し、再度 Web サーバと通信を行わせることで正当なユーザの継続的なサービス利用を担保することができる。

4.1 システムの構成

システムの構成を図 5 に示す。

- メインサーバ
  通常の Web サービスを行うサーバ。攻撃者ではないと判断されたユーザとのみ通信を行う。正当なユーザへ正常なパフォーマンスのサービスを提供することを目的としている。
- 検疫サーバ
  メインサーバと同じサービスを行うサーバ。IDS が攻撃を検知したとき、判定期間にアクセスしたユーザにサービスを提供する。一連のアクセスの特徴から正当なユーザと攻撃を識別する機能を持つ。
- 経路制御装置
  経路制御とパケット転送を行う装置。OpenFlow コントローラ、OpenFlow スイッチから構成される。
- IDS
  侵入検知システム。LAN へのパケットを監視し、攻撃を検知する。攻撃を検出すると、OpenFlow コントローラにアラートを送信する。

4.2 動作

システムの動作を図 5 を用いて説明する。

(1) パケット受信
経路制御装置がパケットを受信してから、IDS によって攻撃が発生しているか判定が行われる。

(2) フローエントリ挿入
攻撃が発生していないと判定された場合は攻撃を行っているユーザはいないと考え、判定期間中にアクセスしてきたユーザはメインサーバと通信できるようにフローエントリを登録する。またアクセスが Time_Limit 秒なければ正当なユーザがサービスの利用を終了したと考えるとき、フローエントリの保有時間には Time_Limit 秒を設定する。既に送信元 IP アドレスがフローテーブ

ルに登録されている場合はフローテーブルに従ってパケットの転送や破棄を行う。

攻撃が発生したと判定された場合、IDS は判定期間中にアクセスしてきた送信元 IP アドレスを OpenFlow コントローラに通知する。経路制御装置は通知された送信元 IP アドレスを検疫サーバへ転送するフローエントリを登録する。フローエントリの保有時間には Time_Limit 秒を設定する。

検疫サーバへ転送するフローエントリの優先度は、メインサーバへ転送もしくは破棄するときよりも小さい値が設定される。この設定により、既に正当なユーザもしくは攻撃と判定されているユーザが検疫サーバへ送信されることを防ぐ。

(3) 検疫サーバによる識別
検疫サーバはユーザからの一連のアクセスを観察して正当なユーザと攻撃を識別する。

(4) 識別後の経路制御
検疫サーバは経路制御装置に識別した IP アドレスと識別結果を送信する。経路制御装置は正当と判定された IP アドレスはメインサーバへ転送し、攻撃と判定された IP は破棄するフローエントリを登録する。

5. 実装のための関連技術

我々は DDoS 攻撃緩和システムの機能の一部を実現している。この節では DDoS 攻撃緩和システムの機能の実現に使用した関連技術について説明する。

5.1 Ryu

Ryu[14] は SDN アプリケーション作成に必要なライブラリやツールを提供するフレームワークであり Python で記述される。本研究では Ryu を使用して OpenFlow コントローラの機能の一部を作成した。

5.2 Snort

Snort[15] は IDS の一種である。本研究では IDS に Snort を採用して DDoS 攻撃を検知する機能を実装した。

5.3 Open vSwitch

Ryu[16] は SDN アプリケーション作成に必要なライブラリやツールを提供するフレームワークであり Python で記述される。本研究では Ryu を使用して OpenFlow コントローラの機能を作成した。

5.4 Mininet

Mininet[14] は OpenFlow コントローラや OpenFlow スイッチ、複数のホスト、ルー タなどを組み合わせて、仮想ネットワークを 1 つの LinuxOS 上で作成できるネッ トワークエミュレータである。仮想ネットワークは Python でスクリプトを書くことで作成することができる。この Mininet を利用することで簡単にネットワークを作成し OpenFlow の動作確認を行うことができる。本研究でも OpenFlow コントローラのプログラムを作成する際、プログラムの動作確認を行うためにこの Mininet を活用した。

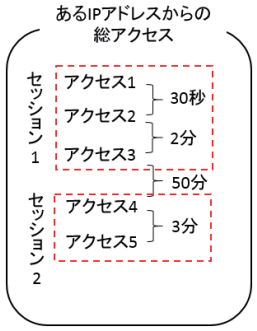


図 6. セッションの概要図

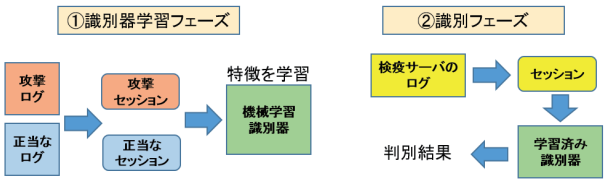


図 7. 以前の手法の識別の流れ

6. 以前の識別手法

6.1 概要

第 2 章でも述べたように、HTTP-GET Flood 攻撃の個々のアクセスは正式な HTTP プロトコルに準拠したものであるため正当なアクセスと区別することが難しい。そこで以前の研究では、人が Web サイトにアクセスして欲しい情報を得ようとする場合と、HTTP-GET Flood 攻撃を起こして Web サーバに高負荷を与えようとする場合の一連のアクセスの挙動差を利用して識別を行う。

以前の手法では一連のアクセスをセッションと呼ぶ。セッションの概要を図 6 に示す。セッションはあるユーザが Web サイトを訪問してから離脱するまでのアクセスのまとまりである。本研究では同じページに留まる時間は長くても 30 分程度と仮定し、最後のアクセスから 30 分以上間隔があいた後のアクセスは新たな別のセッションとする。

正当なセッションの特徴はメインサーバが過去に受けた正当なユーザからのアクセスログから取得する。攻撃のセッションの特徴は、攻撃ツールによって擬似的に Web サーバを攻撃してそのサーバのログから取得する。判定を行いたいセッションから抽出した特徴を上記の特徴と比較することで、判定対象のセッションが正当なユーザか攻撃か識別を行う。

判定対象のセッションが正当なセッションと攻撃のセッションのどちらに近いかは機械学習を用いることで判断する。まず正当なセッションと攻撃セッションの特徴を識別器に学習させ、そのあと検疫サーバのログを識別器に読み込ませることで識別を行う。識別の流れを図 7 に示す。

6.2 識別に利用する挙動差

以前の研究では、Web ページへアクセスする際に正当なユーザと攻撃ではいくつかの挙動差があると仮定した。以下にその挙動差とそれぞれの挙動差が存在すると仮定した根拠を述べる。

- **正当なユーザによるアクセスは攻撃に比べてアクセス間の間隔が長い**

攻撃の場合は Web サーバに負荷をかけるために短い間に大量のアクセスをする必要があるためアクセスの間隔は短くなりやすいが、正当なユーザは目的の情報やファイルを探したり取得するための時間が必要となるためアクセス間の間隔が攻撃より長くなりやすいと考えた。

- **正当なユーザは Web ページ内にあるリンクを辿ってページを遷移することが多い**
  正当なユーザは興味のある内容に関連したページを閲覧するためリンクを辿ってアクセスを行うことが考えられるが、攻撃ではリンクを辿るようなアクセスは行われないと考えた。例えばボットネットを用いて HTTP-GET Flood 攻撃を行うことができる有名なツールに Dirt Jumpeer[17] があるが、Dirt Jumper でボットネットを操作し攻撃を起こす際、ボットがアクセスするページは URI を入力することで指定するためリンクの有無に関係なく次にアクセスされるページが決定される。
- **正当なユーザはよく閲覧されるページへアクセスしやすい**
  攻撃はページの閲覧される度合いに関係なくランダムにアクセスするが、正当なユーザはページの用途を考慮してアクセスするため閲覧される度合いが高いページにアクセスしやすいと考えた。

6.3 使用した特徴量

以前の研究では 6.2 節で仮定した挙動差を表すことができる特徴量を選択した。識別はセッション単位で行うため、セッションごとに以下の特徴量の抽出を行う。

- **アクセス回数**
  アクセスログの行数を数えることで取得する。
- **アクセス間隔の平均時間・標準偏差**
  アクセスログの「時刻」項目から各アクセス間の間隔を計算し、それらの平均と標準偏差を計算する。
- **リンクのないページへ遷移した割合**
  アクセスログからどのページへアクセスしたかという情報を取得して、Web サイトのリンク構造情報からリンクのないページへ遷移している回数を計測し、セッション内の遷移した合計回数で割ることで算出する。

6.4 識別の流れ

まず正当なアクセスと攻撃の特徴量を識別器に学習させる。正当なユーザの特徴量はメインサーバの過去の運用によって記録されたアクセスログから取得し、攻撃の特徴量は攻撃ツールによって Web サーバを攻撃することによって得たアクセスログから取得する。特徴量を識別器に学習させる識別器学習フェーズの詳細を図 8 に示す。このフェーズには「攻撃」、「セッション分割」、「特徴量抽出」、「学習」の 4 段階がある。「攻撃」では、攻撃が記録されたアクセスログを得るために攻撃ツールで Web サーバに攻撃を行う。「セッション分割」では用意したアクセスログのアクセス群をセッション単位に分割したデータを作成する。「特徴量抽出」ではセッション単位に分割したデータから特徴量を抽出する。「学習」では抽出した特徴量を識別器に読み込ませて学習させる。

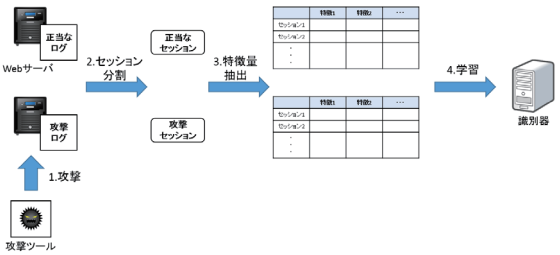


図 8. 識別器学習フェーズ

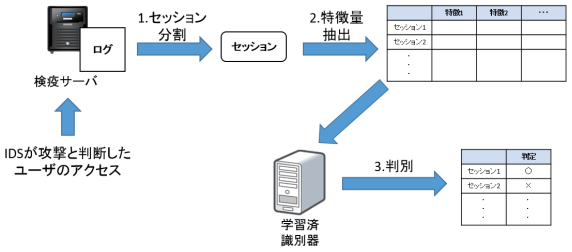


図 9. 正当なセッションを判別するフェーズ

識別器に特徴量を学習させると次に正当なユーザと攻撃の識別を行う。IDS によって DDoS 攻撃が発生したと判定されると、判定期間中にアクセスしてきたユーザのアクセスは経路制御装置によって検疫サーバに転送され、アクセスログが記録される。記録されたアクセスをセッション分割したものから特徴量を抽出し、識別器に読み込ませることで判別を行う。正当なセッションを判別するフェーズの詳細を図 9 に示す。このフェーズには「セッション分割」「特徴量抽出」「判別」の 3 段階がある。「セッション分割」と「特徴量抽出」に関しては識別器学習フェーズと同様である。「判別」では、抽出された特徴量を識別器学習フェーズで学習を行った識別器に読み込ませることで、正当なセッションか判別する。

6.5 アクセス回数 1 のセッション判別

アクセス回数 1 のセッションは 6.4 節で述べた手法では識別することができない。これはアクセス回数 1 のセッションではアクセス間の間隔やページ遷移が存在せず、仮定した挙動差を得ることができないためである。そこでアクセス回数 1 のセッションはエントロピーを用いた別の手法で識別を行う。あるページ i へのアクセスという事象が持つ情報量を I、ページ i にアクセスされる確率を P_i とする。

$$I = -\log_2 P_i$$

$$P_i = \frac{\text{ページ } i \text{ へアクセスがあった回数}}{\text{アクセス総数}}$$

このときページアクセスのエントロピー H は以下のように表すことができる。

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i$$

6.2 節で述べたように正当なユーザはよく閲覧されるページへアクセスすることが多く、攻撃はランダムにアクセスしやすいと仮定している。IDS が攻撃だと判定したユーザの中であまり閲覧されていないページへアクセスしているユーザは、よく

閲覧されているページへアクセスしているユーザに比べ、より攻撃である可能性が高いと判断する。このとき閲覧される度合いとしてページアクセスのエントロピーを利用する。ページ i へのアクセスを識別する際は「ページ i へのアクセス」という事象が持っている情報量がエントロピーよりも低い場合はよく閲覧されるページへのアクセスであるため正当なユーザと判定する。逆に情報量がエントロピーより高い場合はあまり閲覧されないページへのアクセスであるため攻撃と判定する。通常時の閲覧される度合いを判定に使うため、メインサーバの過去の運用で得られたアクセスログからエントロピーを計算する。

6.6 以前の識別手法の問題点

DDoS 攻撃緩和システムは正当なユーザの継続的なサービス利用を目的としている。この目的を達成するためにシステムにはサービスの運用中に識別が行える手法が必要である。しかし以前の識別手法はユーザのサービス利用終了後にログを解析して識別することしかできないため、目的の実現に適切でない。

7. 提案手法

この節では以前の手法の問題を解決するための新たな識別手法について説明する。

7.1 提案手法の概要

6.6 節で述べたように DDoS 攻撃緩和システムにはサービスの運用中に識別が行える手法が必要である。そこで我々はユーザからアクセスがある度に一連のアクセスの特性に対して評価を行い、サービス運用中に識別が行える手法を提案する。

提案手法は検疫サーバにアクセスがある度に送信元のユーザに対してポイントを加減算する。リクエストに正当なユーザーと思われる特性がある場合は送信元のユーザにポイントを追加する。逆にリクエストに攻撃と思われる特性がある場合は送信元のユーザからポイントを差し引く。ポイントが正の閾値に達した場合はユーザは正当なユーザであると判定される。負の閾値に達した場合は攻撃と判定される。本研究では正の閾値を 10 点、負の閾値を −10 点とした。閾値に達する前に Time_Limit 秒経過して通信が終了した場合、ポイントが 0 点以上なら人間、0 点未満なら攻撃と判定する。

以上のように一連のアクセスの評価と判別を行うことでサービス運用中に識別を行うことができる。

7.2 使用する特性

使用する特性は 6.2 節で述べている以前の手法で利用した挙動差を基に定めている。詳細は以降の節で説明する。

7.3 加算を行う特性

ユーザから検疫サーバへアクセスがあった際、正当なユーザである可能性が高いと考えポイントを加算するのは、アクセスに以下の特性がある場合である。

- **リンクを辿った**
  連続でリンクを辿った場合は連続で辿った回数を考慮して点数を加算する。1 回リンクを辿ったときの点数を Link_point、連続でリンクを辿った回数を Link_number とするとき以下の点数を加算する。

$$Link\_point^{Link\_number}$$

- **人気のあるページにアクセスした**
  アクセスしたページが人気であるか否かは 6.5 節で述べたエントロピーを用いた手法によって判定する。
- **平均送信間隔の welch の t 検定により人間であると判定された**
  判定したいユーザの現在の平均送信間隔と攻撃の平均送信間隔を使って welch の t 検定を行う。検定の結果、判定したいユーザの平均送信間隔が攻撃の平均送信間隔より長いと判定された場合、人間と判定する。本研究では有意水準は 0.05 とした。攻撃の平均送信間隔は以前の識別手法と同じく攻撃ツールを使用して擬似的に作成した攻撃ログから学習しておく。

7.4 減点を行う特性

ユーザから検疫サーバへアクセスがあった際、攻撃である可能性が高いと考えポイントを減点するのは、アクセスに以下の特性がある場合である。

- **リンクを辿っていない**
  連続でリンクを辿っていない場合は連続で辿らなかった回数を考慮して点数を減点する。1 回リンクを辿らなかったときの点数を Not_link_point、連続でリンクを辿らなかった回数を Not_link_number とするとき以下の点数を減点する。

$$Not\_link\_point^{Not\_link\_number}$$

- **人気のないページにアクセスした**
  アクセスしたページが人気であるか否かは 6.5 節で述べたエントロピーを用いた手法によって判定する。
- **平均送信間隔の welch の t 検定により攻撃であると判定された**
  判定したいユーザの現在の平均送信間隔と正当なユーザの平均送信間隔を使って welch の t 検定を行う。検定の結果、判定したいユーザの平均送信間隔が正当なユーザの平均送信間隔より短いと判定された場合、攻撃と判定する。本研究では有意水準は 0.05 とした。正当なユーザの平均送信間隔は以前の識別手法と同じく過去のメインサーバのログから学習しておく。

7.5 加減算する点数の決定法

それぞれの特性に割り当てる点数の決め方について説明する。点数決定までの流れを図 10 に示す。まず、正当なユーザのアクセスログと攻撃ログを学習用と判定用に 2 分割する。正当なユーザのアクセスログにはメインサーバの過去の運用によって記録されたアクセスログを用いる。攻撃のログには攻撃ツールによって Web サーバを攻撃することによって得たアクセスログを用いる。提案手法ではあらかじめ正当なユーザと攻撃の平均送信間隔および各ページの情報量を学習しておく必要がる。よって学習用ログは上記の情報を学習するた

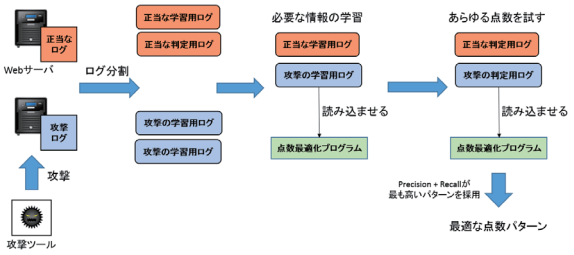


図 10. 点数最適化の流れ

	正当な特性 1	正当な特性 2	…	正当な特性 N	攻撃の特性 1	攻撃の特性 2	…	攻撃の特性 N	Precision + Recall
パターン1	+1	+1	…	+1	-1	-1	…	-1	1.0
パターン2	+1	+1	…	+1	-1	-1	…	-2	1.5
・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・
パターン1000	+10	+10	…	+10	-10	-10	…	-10	1.0

パターン2のPrecision + Recallが1番大きい場合，パターン2を採用

図 11. 点数最適化の例

めに使用する。判定用ログは検疫サーバへのアクセスの代用として判定に用いる。次にそれらのログを点数最適化プログラムに読み込ませて点数を決定する。点数最適化プログラムは提案手法のロジックであらゆる点数パターンの識別を擬似的に行えるプログラムである。加点を行う特性の場合は 1 点から正の閾値まで、減点を行う特性の場合は-1 点から負の閾値までの全ての点数のパターンを試し、*Precision* と *Recall* の合計値が最も高い点数パターンを最適な点数として採用する。*Precision* は正当だと判定した IP アドレスのうち実際に正当だった割合である。*Recall* は正当な IP アドレスのうち判定結果が正しかった割合である。点数最適化の例を図 11 に示す。

7.6 識別方法

識別の流れを図 12 に示す。まず 7.5 節で説明した方法で各特性の点数を決定する。

次に検疫サーバの識別を行うプログラムである識別プログラムに正当なログと攻撃ログを読み込ませて、正当なユーザと攻撃の平均送信間隔、各ページのエントロピーを学習する。正当なユーザのアクセスログにはメインサーバの過去の運用によって記録されたアクセスログを用いる。攻撃のログには攻撃ツールによって Web サーバを攻撃することによって得たアクセスログを用いる。

以上の準備が完了すると識別が行える。IDS によって DDoS 攻撃が発生したと判定されると、判定期間中にアクセスしてきたユーザのアクセスは経路制御装置によって検疫サーバに転送される。検疫サーバへのユーザのアクセスが発生する度に一連のアクセスの特性に応じてユーザに対してポイントを加減算し、ポイントが正の閾値に達したユーザは正当なユーザ、ポイントが負の閾値に達したユーザは攻撃と識別する。

8. 実験概要

DDoS 攻撃緩和システムにはサービスの運用中に正当なユーザか攻撃か識別できる手法が必要である。そこで実験により、

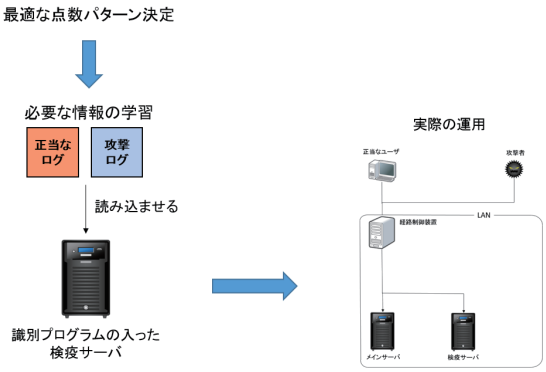


図 12. 識別の流れ

提案手法がサービス運用中に正当なユーザと攻撃を識別できているか検証する。実験は宮崎大学で取得したログと攻撃ツールによって作成したログを提案手法のロジックを実装したプログラムに読み込ませる擬似的な方法で行った。

8.1 実験環境

8.2 使用したプログラム

本研究の実験では 2 つのプログラムを使用した。

- **点数最適化プログラム**
  設定した範囲内の全ての点数パターンで実験を行い最適な点数を求めるプログラム。
- **識別プログラム**
  最適化した点数パターンを利用して正当なユーザと攻撃の識別を行うプログラム。

8.3 使用したログ

本研究では以下の 2 つの Apache[18] ログを使用した。

- **正当なユーザのアクセスを記録したログ**
  宮崎大学情報システム工学科 Web サーバから取得した約 1 年間分のログ。
- **HTTP-GET Flood 攻撃を記録したログ**
  BoNeSi[19] という攻撃ツールで Web サーバに攻撃し作成したログ。

9. 実験方法

9.1 *k*-分割交差検証

本研究では図 13 で示すように *k*-分割交差検証によって実験を行う。データセットをいくつかに分割し、ま ず 1 つをテストデータ、その他を訓練用データとして精度の算出を行う。次に別のデータをテストデータとして選択し、残りをデータを訓練用データとして再度精度の算出を行う。この処理を分割した回数行う。それぞれの結果の平均が実験結果となる。本研究では 4 分割で実験を行った。

9.2 実験手順

1. **ログの分割**
   正当なログと攻撃のログをそれぞれ 4 分割して 1 つを判定用ログ、それ以外を学習用ログとする。

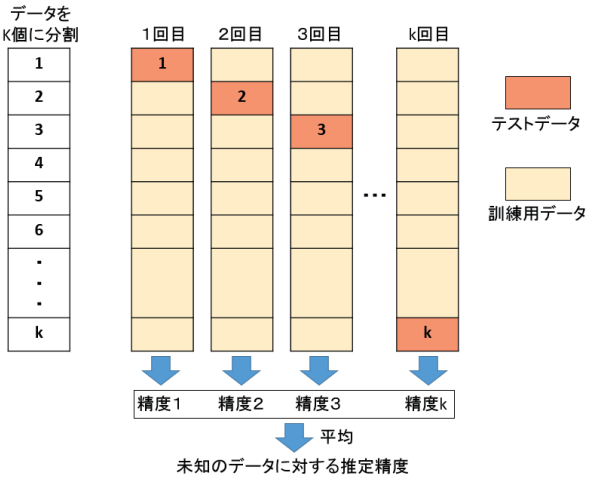
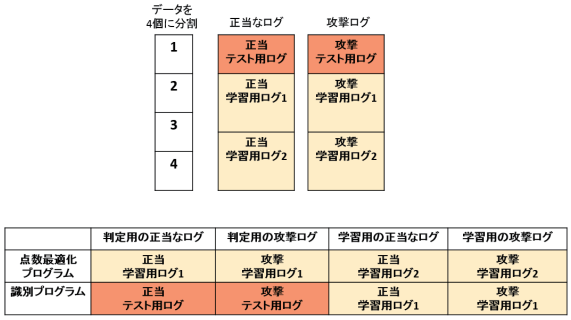


図 13. *k*-分割交差検証



	判定用の正当なログ	判定用の攻撃ログ	学習用の正当なログ	学習用の攻撃ログ
点数最適化プログラム	正当 学習用ログ1	攻撃 学習用ログ1	正当 学習用ログ2	攻撃 学習用ログ2
識別プログラム	正当 テスト用ログ	攻撃 テスト用ログ	正当 学習用ログ1	攻撃 学習用ログ1

図 14. プログラムに読み込ませるログ

2. **判定に使用する点数パターンの決定**
   学習用ログを点数最適化プログラムに読み込ませて、判定に使用する点数パターンを決定する。
3. **学習**
   識別プログラムに学習用ログを読み込ませて正当なユーザと攻撃の平均送信間隔および各ページの情報量を学習させる。
4. **識別**
   テスト用のログと点数パターンを識別プログラムに読み込ませて判定精度の算出を行う。

以上の手順を 4 分割交差検証で行う。また今回は各プログラムに図 14 のようにログを読み込ませて実験を行った。

9.3 パラメータ

実験ではパラメータを以下のように設定した。

- **セッション終了とする送信間隔**
  以前の手法と同じく送信間隔が 1800 秒以上空いた場合はセッション終了とした。
- **正負の閾値**
  正の閾値を+10 点、負の閾値を-10 点とした。

9.4 評価指標

評価指標を図 15 と数式を使って説明する。

- *Accuracy*
  全 IP アドレスのうち判定結果が正しかった割合。

$$Accuracy = \frac{TP + FP}{TP + FP + FN + FP}$$

		実際	
		正当なユーザ	攻撃
判定結果	正当なユーザ	TP	FP
	攻撃	FN	TN

図 15. 評価指標の説明補助

表 1. 実験結果

	Accuracy	*Precision*	*Recall*	検知時間	正当なユーザの検知時間
1 回目	72%	53%	73%	1395 秒	551 秒
2 回目	73%	54%	74%	1381 秒	525 秒
3 回目	72%	55%	70%	1400 秒	629 秒
4 回目	72%	58%	68%	1386 秒	594 秒
平均	72%	55%	71%	1391 秒	594 秒

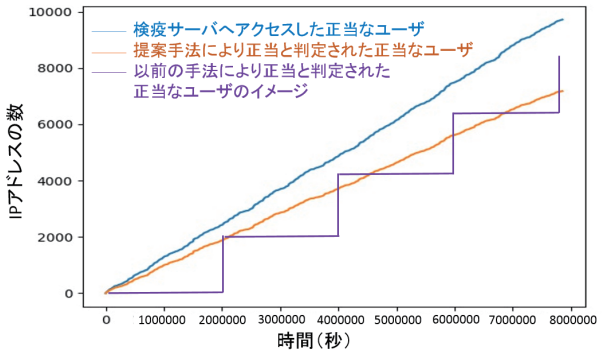


図 16. 正当と判定される正当なユーザの推移

- *Precision*
  正当だと判定した IP のうち実際に正当であった割合。

$$Precision = \frac{TP}{TP + FP}$$

- *Recall*
  正当な IP のうち判定結果が正しかった割合。

$$Recall = \frac{TP}{TP + FN}$$

- **平均検知時間**
  1IP アドレスあたりの検知時間の平均

10. 実験結果と考察

実験結果を表 1 と図 16 に示す。提案手法の正当なユーザの平均判定時間は 594 秒となり、検疫サーバで最低 30 分以上かけて取得したログを解析することしかできない以前の手法より即応性が高いことが分かる。また、以前の手法はログの事後解析しか行えないため、図 16 に示すように提案手法の方が以前の手法より即応性が高くなる。以上より提案手法は以前の手法より即応性が高く、正当なユーザの継続的なサービス利用の実現に適切であるといえる。

判定精度に関しては、*Precision* が 51%、*Recall* が 71%となった。我々はこの判定精度は実用的なレベルに達していないと考えている。よって、今後はさらに判定精度を向上させる判定方法や特性を考える必要がある。

11. まとめ

DDoS 攻撃は大きな脅威である。我々はこの攻撃の対策として DDoS 攻撃緩和システムの研究を行なっていた。DDoS 攻撃緩和システムにはサービス運用中に正当なユーザと攻撃を一連のアクセスの特性を利用して識別できる即応性の高い識別方法が必要である。しかし、以前の識別手法はサービス運用中に識別を行えない即応性の低いものだった。そこで、本研究ではサービス運用中に正当なユーザと攻撃を識別できる新しい方法を提案した。実験によって、提案手法はサービス運用中に識別が行える以前の手法より即応性の高い手法であることを確認した。今後は判定精度を上げるために識別手法や利用する特性について検討していきたい。

参考文献

1) IT トレンド: 不正侵入検知・防御システム（IDS・IPS）とは? ⟨https://it-trend.jp/ids-ips/article/explain⟩ (accessed 2018-01-24).

2) T. Yatagai, T. Isohara, and I. Sasase: Detection of HTTP-GET Food Attack Based on Analysis of Page Access Behavior, in *Proceedings IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp.232–235, 2007.

3) Q. Liao, H. Li, S. Kang, and C. Liu: Application Layer DDoS Attack Detection Using Cluster with Label Based on Sparse Vector Decomposition and Rhythm matching, *Security and Communication Networks*, Vol. 8, No. 17, pp. 3111–3120, 2015.

4) K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris: Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments, *Computer Networks: The International Journal of Computer and Telecommunications Networking archive*, Vol. 62, pp.122–136, 2014.

5) 有川佑樹、 岡崎直宣、 山場久昭、 高塚佳代子、 久保田真一郎: OpenFlow によるネットワーク制御と擬陽性排除サーバを用いた DDoS 攻撃緩和手法の検討, 火の国情報シンポジウム 2016, 火の国情報シンポジウム 2016 論文集, 2016.

6) 橘弘智, 有川祐樹, 臼崎翔太郎, 久保田真一郎, 高塚佳代子, 山場久昭, 岡崎直宣 “DDoS 攻撃ログデータ解析による人と攻撃通信判別に関する研究” 宮崎大学工学部紀要（46）pp. 239–246, 2017.

7) J. Mirkovic, and P. Reiher: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, *Newsletter ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2, pp. 39–53, 2004.

8) 寺田真敏: DoS/DDoS 攻撃とは, 情報処理, Vol. 54, No. 5, pp. 428–435, 2013.

9) 警察庁: Slow HTTP DoS Attack に対する注意喚起について ⟨https://www.npa.go.jp/cyberpolice/detect/pdf/20151216.pdf⟩ (accessed 2018-01-24).

10) E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello: Mobile executions of Slow DoS Attacks, *Logic Journal of the IGPL*, Vol. 24, No. 1, pp. 54–67, 2016.

11) I. Duravkin, A. Carlsson, and A. Loktionova: Method of slow-attack detection, Problems of Infocommunications Science and Technology, in *2014 First International Scientific-Practical Conference*, pp. 171–172, 14–17, 2014.

12) 下川大貴、小刀祢知哉、池部実、吉田和幸: OpenFlow を用いた攻撃者遮断システムの提案と評価, マルチメディア・分散協調とモバイルシンポジウム 2014 論文集, pp.197–204, 2014.

13) 安部幸太郎、森口一郎: DNS を用いた DDoS 攻撃回避システム, 東京情報大学研究論集, Vol. 17, No. 2, pp.63–72, 2014.

14) SDN Framework ⟨https://osrg.github.io/ryu-book/ja/Ryubook.pdf⟩, (accessed 2015-10-30).

15) Snort ⟨https://www.snort.org/⟩ (accessed 2016-02-10).

16) Open vSwitch ⟨http://openvswitch.org/⟩ (accessed 2015-10-30).

17) Dirt Jumper Ver.5 Technical Security Notes ⟨https://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/research_DirtJ5.pdf⟩ (accessed 2017-01-26).

18) Apache HTTP Server Project ⟨https://httpd.apache.org/⟩ (accessed 2018-01-23).

19) GitHub: Markus-Go/bonesi: BoNeSi - the DDoS Botnet Simulator, ⟨https://github.com/markus-go/bonesi⟩ (accessed 2018-01-23).