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

橘　弘智 [a)]・有川　祐樹 [b)]・臼崎　翔太郎 [c)]・久保田　真一郎 [d)]・
高塚　佳代子 [e)]・山場　久昭 [f)]・岡崎　直宣 [g)]

Discriminating Legitimate Accesses from a Web Access Log Recorded During DDoS Attack

Hiroaki TACHIBANA, Yuki ARIKAWA, Shotaro USUZAKI, Shin-Ichiro KUBOTA,
Kayoko TAKATSUKA, Hisaaki YAMABA, Naonobu OKAZAKI

Abstract

Web services are indispensable in everyday life, and damage caused by denial of service (DoS)/distributed denial of service (DDoS) attacks is becoming serious. An intrusion detection system (IDS) is very useful to detect various attacks including DDoS attacks. But an IDS often makes false detections, not a few legitimate accesses are reported as attacks. Then, there is a possibility that a legitimate user who is detected erroneously can not receive service . We proposed a system to mitigate HTTP-GET Flood attack that is one of DoS/DDoS attacks in the previous work. This system not only can protect servers from attacks using IDS but also can guarantee their services by introducing a server that picks out legitimate accesses in the accesses detected by the IDS. In this study, we propose a method to find out legitimate accesses that is the important part of the HTTP-GET Flood attack mitigation system. Information obtained from the access log is used in the method. Besides, since false detections such that an attack is picked out as a legitimate access make IDSs ineffective, the proposed method must keep such false detection rate low. We conducted an experiment that uses an access log of an actual server to verify the effectiveness of this system. The result of the experiment showed that the proposed method picked out many of the legitimate users that were charged by the IDS falsely and could practically avoid picking out malicious attacks by mistake.

Keywords: HTTP-GET Flood attack, Web access log, Machine learning

1. はじめに

今日の重要な社会基盤となっているインターネットにおいて, サービス不能攻撃 (Denaial of Service Attack, 以下 DoS 攻撃) や分散 DoS 攻撃 (Distributed Denial of Service Attack, 以下 DDoS 攻撃) は深刻な問題となっている．DoS/DDoS 攻撃には様々な種類が存在するが，我々は Web サーバを対象とした攻撃の一種である HTTP-GET Flood 攻撃を対象に研究を行ってきた．HTTP-GET Flood 攻撃は正常なアクセスを大量に行うことで攻撃とするため，攻撃と正当なアクセスの見分けがつきにくく解決に至っていない．

DDoS 攻撃を含む様々な攻撃を検知するために侵入検知システム (Intrusion Detection System, 以下 IDS) が使用され，サーバ管理者は IDS の検知結果に従い，検知された IP アドレスからのアクセスを遮断するなどの処置を行う．しかし，IDS は攻撃を見落とすことがないよう厳しく通信を監視するため，正当なアクセスを攻撃だと誤検知 (False Positive) してしまう傾向があり，正当なユーザがサービスを受けられなくなるという問題が発生する．

我々は以前に，IDS が攻撃だと検知したアクセスに対しサービスを提供しつつ正当なアクセスを判別する別の Web サーバ (以下，検疫サーバと呼ぶ) によって，HTTP-GET Flood 攻撃を緩和して正当なユーザのサービス利用を担保するシステムを提案した [1)]．本研究では，有川らが提案する HTTP-GET Flood 攻撃緩和システムを実現するために必要となる，検疫サーバに振り分けられたアクセスの中から正当なアクセスを判別する手法について提案し検証を行う．

2. HTTP-GET Flood 攻撃とその対策

HTTP-GET Flood 攻撃は DoS/DDoS 攻撃の一種であり，DoS/DDoS 攻撃とは，サーバに対して意図的に過剰な負荷を与えたり，サーバ等の脆弱性を悪用することで，サービスを提供できない状態にする攻撃である．DoS 攻撃が単一のホストから行われる攻撃であるのに対して，DDoS 攻撃は複数のホストにあらかじめ攻撃プログラムを仕込んでおき，それらの分散しているホストから一斉に特定のサーバを対象とした

[a)] 情報システム工学専攻大学院生
[b)] 工学専攻大学院生
[c)] 情報システム工学科学部生
[d)] 情報システム工学科准教授
[e)] 教育研究支援技術センター技術専門職員
[f)] 情報システム工学科助教
[g)] 情報システム工学科教授

攻撃を引き起こすものである．

寺田は，攻撃の対象によって DoS/DDoS 攻撃を以下の種類に分類できるとしている [2]．

- **ルータやサーバの脆弱性への攻撃**
  権限のない攻撃者が不正な方法でプログラムを実行したりファイルを読み書きするために，安全性が考慮されていないプログラムのコーディングによって発生する脆弱性に対して行われる攻撃
- **回線帯域への攻撃**
  正当なユーザがサーバに接続不能になるように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
- **Web サーバへの攻撃**
  システム資源を大量に消費して Web サーバをサービス不能な状態にする攻撃

HTTP-GET Flood 攻撃は Web サーバへの攻撃に分類され，図 1 に示すように TCP コネクション確立後一斉に HTTP-GET 要求を送信することで，サーバに大量の応答処理を発生させ，パフォーマンスを低下させたり，動作を停止させる攻撃である．

現在，この HTTP-GET Flood 攻撃の対策に関する様々な研究が行われているが解決には至っていない．これは，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おり，攻撃と正当なアクセスに差がないという特徴があるためである．

2.1 HTTP-GET Flood 攻撃の検知

HTTP-GET Flood 攻撃の検知を行う基本的な方法として Apache モジュールを導入する方法があり，主に mod_evasive モジュールと mod_dosdetector モジュールが使用される．これらのモジュールでは同一 IP アドレスから一定量以上のアクセスがあったかどうかで攻撃の検知を行う．そのため多量のホストから少量ずつアクセスが送られるような攻撃では検知できない．

Yatagai らは，ページアクセスの挙動を解析することで HTTP-GET Flood 攻撃を検知する手法を提案している [3]．具体的な攻撃検知の方式としては，アクセス順序の重複検知方式とページ内情報量と閲覧時間の相関に注目する検知方式の 2 種類があり，判別精度を測る実験を行った結果，4 または 5 回程度のアクセスを解析すれば高い精度で攻撃検知できるとしている．そのため，Yatagai らの手法を使うと，同一 IP アドレスからは少量のアクセスで大量のホストに分散して行われる攻撃を検知できる．

ウイルスやボットネットに感染した端末による HTTP-GET Flood 攻撃は予め設定された URI に対してリクエストを送信するため，同じウイルスに感染した端末は同じようなページアクセスを連続的に行うという特徴を持つ．アクセス順序の重複検知方式は，そのページアクセスの特徴を利用して，各 IP アドレス同士のアクセス順序を比較して同じアクセス順序でアクセスしている IP アドレスが複数確認できた場合，それらの IP アドレスからのアクセスは攻撃であると判定する．

また，ページ内情報量と閲覧時間の相関に注目する検知手法では，正当なアクセスであればページ内に含まれる情報量

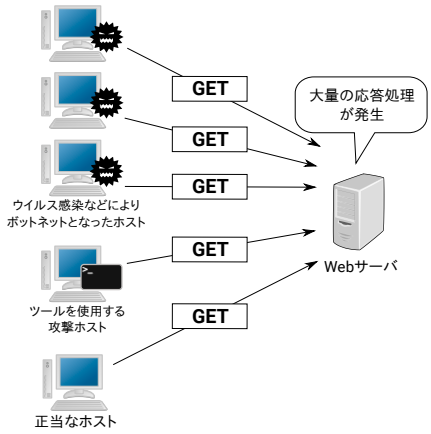

図 1. HTTP-GET Flood 攻撃

が多ければ多いほど閲覧時間が増え，逆に攻撃によるアクセスではページ内の情報量にかかわらずアクセスを行うという特徴を利用している．あらかじめ正当なアクセスをもとに記録した，各ページの情報量と閲覧時間の関係を表す近似直線を求めておき，アクセスのあったページの情報量に対して直線から求まる閲覧時間と比べて，アクセスによる閲覧時間が少ない場合に攻撃だと判定する．

しかし，アクセス順序の重複検知方式で対応できるのは同じ順序でアクセスするよう設定された攻撃のみであり，同一ホスト内でアクセス先をランダムに選択するような攻撃を検知することはできない．また，ページ内情報量と閲覧時間の相関に注目する検知方式では，等間隔でアクセスを送り続けるような攻撃であれば，近似直線の傾きがあらかじめ求めた近似直線の傾きより小さいため検知できるが，ランダムな間隔を開けてアクセスする攻撃に対しては，近似直線の傾きが偶然あらかじめ求めた近似直線の傾きより小さくなる場合でしか検知することができない．

2.2 HTTP-GET Flood 攻撃の緩和策

攻撃の影響を軽減する手法として仮想計算機を使用し攻撃を緩和する研究が行われている [4, 5, 6]．これらの研究では，攻撃者がサーバ側で攻撃を検知し何らかの対策を行ったことに気づき，攻撃対象のサーバを変更したり攻撃手段を変更して対策を回避するように攻撃するという問題をとりあげている．この問題に対して，仮想計算機による手法では攻撃者が利用できる仮想計算機の資源を制限して提供することで，攻撃が成功しているかのように見せ，攻撃者が対策を回避するのを防ぐ手法を提案している．この方法では，攻撃検出器の性能によっては正当なユーザにパフォーマンスの悪いサービスを提供してしまう可能性がある．

2.3 HTTP-GET Flood 攻撃緩和システム

我々は，通常の Web サービスを行う Web サーバ（以下，メインサーバ）とは別に，IDS に検知されたアクセスに対し一時的にサービスを行う検疫サーバを使用することで，ユーザのサービスを担保しつつ，HTTP-GET Flood 攻撃を緩和するシステムの研究を行ってきた [1]．

このシステムの目的は，IDS によって攻撃と正当なアクセスの見分けがつかず一緒に破棄されていた正当なアクセスを救い，正当なユーザのサービスを担保することである．この

システムでは，HTTP-GET Flood 攻撃時，IDS が攻撃と判定したアクセスを一旦検疫サーバで受け入れ，検疫サーバで正当なアクセス元と攻撃アクセス元を判別するという 2 段階の判定を行う．検疫サーバでの判定後，正当なアクセス元からのアクセスはメインサーバへ，攻撃アクセス元からのアクセスは破棄する挙動になるように経路制御を行うことで，IDS に誤検知される正当なアクセスを救う．

システムを構成するそれぞれの装置について以下に説明し，図 2 に本システム構成図を示す．

- **メインサーバ**
  通常の Web サービスを行う．攻撃時以外ではすべてのアクセスを受け入れる．
- **IDS**
  LAN 内に流れてくるパケットを監視して攻撃検知を行い，アラートを出す．
- **検疫サーバ**
  メインサーバと同じ Web サービスを行うサーバであり，IDS が検知したアクセス元からのアクセスを受け入れる．また，正当なアクセス元と攻撃元を判別する機能を有する．
- **経路制御装置**
  外部からのアクセスをどの経路で流すか制御する．

本システムでは非攻撃時，すべてのアクセスがメインサーバへ流れ，メインサーバが Web サービスを行う．攻撃時は以下に示す流れでアクセスが処理される．

(1) IDS が HTTP-GET Flood 攻撃を検知
(2) 検知されたアクセス元からのアクセスが検疫サーバに流れるよう IDS が経路制御装置に指示
(3) 検疫サーバで正当なアクセス元を判別
(4) 検疫サーバは，正当なアクセス元だと判別されたアクセスをメインサーバに，それ以外のアクセスを破棄するよう経路制御装置に指示

この処理の流れを図 3 に示す．

このシステムの重要な要素のひとつである，検疫サーバで正当なアクセスを判別する方法を本研究で提案する．そこで以降では，検疫サーバでの正当なアクセスを判別する方法について考案する．

3. 正当なアクセスを判別する手法

一般に，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．しかし我々は，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では挙動に差が現れると考えた．そこで本研究では，個々のアクセスではなく，ある程度まとまったアクセスに注目することで，Web サイトを利用するホストの挙動を表す情報を得ることとし，この挙動の差を正当なアクセス判別に利用する．

あるユーザが Web サイトを訪問してから離脱するまでのある程度まとまったアクセスをセッションと呼び，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した．

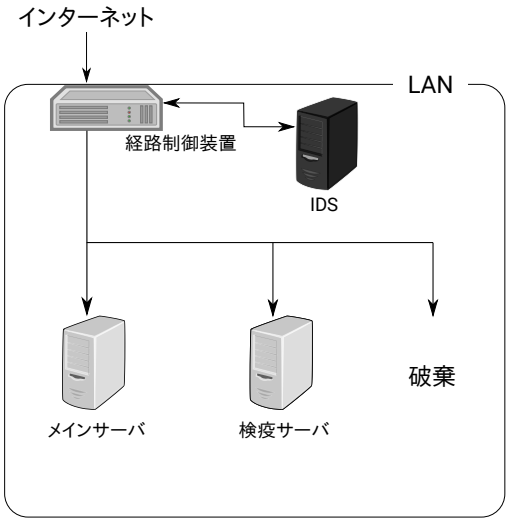

図 2. HTTP-GET Flood 攻撃緩和システムの構成

まず，メインサーバが過去に受けた正当なアクセスの情報から，正当なセッションの特徴を得て，次に，攻撃を受けた際のアクセス情報として，攻撃ツールなどを使用して擬似的に攻撃を起こし特徴を得る．判別を行いたいセッションから抽出した特徴をこれらの特徴と比較し，正当なセッションと攻撃セッションどちらに近いか判断する．

判別したいセッションの特徴が正当なセッションと攻撃セッションのどちらに近いか判断するための方法として機械学習を採用する．機械学習を使用することで挙動の特徴を表す複数の要素を総合的に加味して判別することができる．

また，アクセス情報の取得には Web サーバのアクセスログを使用することとした．パケットキャプチャデータを使用する場合では，正当なアクセスの特徴を得るために長期間パケットキャプチャを行い，正当なアクセスの情報を集める必要があるが，アクセスログを使用する場合では，過去の実運用で記録されたアクセスログを利用できるため，別途正当なアクセスの情報を集める手間がないという利点があると考えた．

3.1 人と攻撃の挙動差

本研究では，Web ページへアクセスする際，1 つのセッションにおいて，人と攻撃の挙動に以下のような差があると仮定した．

(a) 攻撃時のアクセスに比べて人によるアクセスはアクセス間の間隔が長く，ばらつきが大きい

(b) 攻撃時のアクセスに比べ人によるアクセスは 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

人はページの内容を読み欲しい情報を探すため，攻撃に比べアクセスの間隔が長くばらつきが大きくなる傾向があるだろうという考えのもと挙動差 (a) を決定した．また，挙動差 (a) を表す特徴量として以下を選択した．

アクセス回数

内容：セッション内のアクセス回数
抽出方法：アクセスログの行数を数える

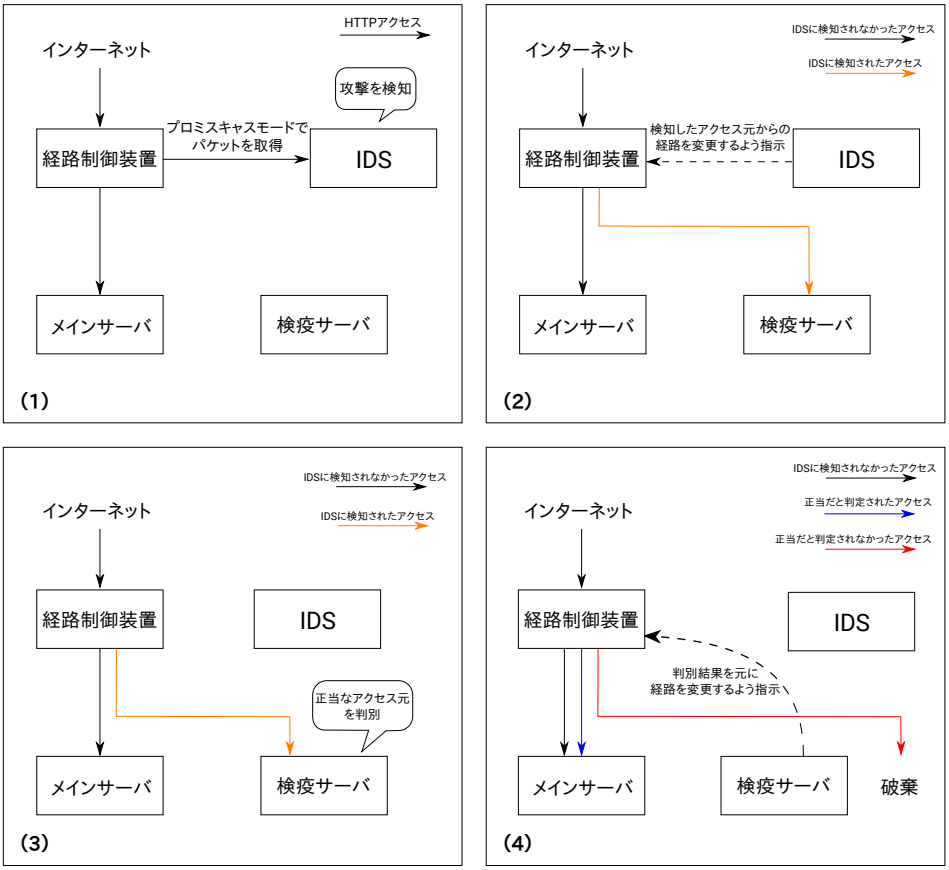

図 3. HTTP-GET Flood 攻撃緩和システム処理の流れ

アクセス間隔の平均時間・標準偏差

内容：アクセス時間間隔の平均と標準偏差

抽出方法：アクセスログの「時刻」項目を元に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

人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と考えられるが，攻撃ではリンクを辿るようなアクセスを行わないと考え挙動差 (b) を決定した．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攻撃ツールキットに Dirt Jumper があるが [7]，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．挙動差 (b) を表す特徴量として以下を選択した．

リンクのないページへ遷移した割合

内容：セッション内の全アクセス回数のうちリンクのないページへ遷移したアクセス回数の割合

抽出方法：アクセスログの「リクエストの最初の行の値」からどのページへアクセスしたかという情報を得て，セッション内の各ページ遷移について，Web サイトのリンク構造情報を元に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計算する

3.2 挙動差により判別を行う流れ

提案する手法での判別は，識別器学習フェーズと判別フェーズからなる．識別器学習フェーズでは，メインサーバの過去の運用で記録されたアクセスログを正当なアクセスログとし，攻撃ツールを使用して作成した攻撃時のアクセスログとして扱う．そして，それぞれのアクセスログをセッションに分割し，正当なセッションと攻撃セッションそれぞれの特徴を識別器に学習させる．

判別フェーズでは，検疫サーバへアクセスが流れることで記録されるアクセスログをセッションに分割したものから特徴抽出し，学習済みの識別器に適用することで正当なセッションを判別する．

3.3 アクセス回数 1 のセッションの判別手法

アクセス回数 1 のセッションにはアクセス間の間隔やページ遷移が存在せず，3.1 節で仮定した挙動差を得ることができないため，アクセス回数が 1 のセッションでは挙動差による判別ができない．アクセス回数が 1 のセッションはページアクセスのエントロピーを利用した別の手法で判定を行う．

よく閲覧されるページへ人はアクセスすることが多く，攻撃によるアクセスはランダムなアクセスになるという仮定のもと，IDS が攻撃だと検知したアクセスの中で，あまり閲覧されないページへのアクセスは，攻撃である可能性が高いと考えた．閲覧される度合いにはページアクセスのエントロピーを使用する．例えば，あるページ A へのアクセスを判別する際，「ページ A 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

表 1. 情報科 Web サーバアクセスログ詳細

期間	総アクセス数	IP アドレス種類数	セッション数
Q1(2015/9/20 - 2015/12/20)	71,899	13,054	52,176
Q2(2015/12/20 - 2016/3/20)	73,443	12,999	54,011
Q3(2016/3/20 - 2016/6/19)	86,268	14,018	50,979
Q4(2016/6/19 - 2016/9/18)	71,735	16,765	54,084

のアクセスなので攻撃ではないとみなし、逆にエントロピー値よりも高い場合は攻撃だと判定する．通常時に閲覧される度合いを判別に利用するので，メインサーバの過去の運用でとられたアクセスログからエントロピーを計算する．

あるページ i へのアクセスという事象が持っている情報量 (I) を使用する．

$$I = -\log_2 P_i$$

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

式中の P_i はページ i にアクセスされる確率を表す．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．

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

4. 正当なセッション判別手法の検証実験

提案する正当なセッション判別手法の有効性を示すため，実運用で得られたアクセスログをもとに判別精度を測定する実験を行う．

このとき，攻撃アクセスを正当なものだとする誤検知の割合を低く抑えることができるかを重視して検証する．これは，検疫サーバが IDS に攻撃と検知されたものの中から正当なアクセスを検出するという役割をもっており，攻撃を正当なものだとする誤検知は IDS の効果を減らしてしまうためである．

4.1 実験で使用するアクセスログ

Apache[11] のアクセスログを検証に利用する．Apache を使用して公開されている Web サイトへクライアントがアクセスすると，クライアントがアクセスした時刻や GET 要求ファイル名などがアクセスログに記録される．Apache は 1996 年から世界シェア 1 位を維持してきた Web サーバソフトウェアであり，使用している企業・団体が多いと考えられることに加え，特別な操作なしにアクセスログが記録される設定であるため，本提案システム用に特別な準備は必要ない．

Apache ではアクセスログに記録する内容を増やすといったようなカスタマイズが可能だが，本提案システムの可用性を保つためデフォルトの設定で記録されるアクセスログ項目から特徴量を抽出する．

正当なセッションと攻撃セッションが混ざったものから正当なセッションを判別できるか検証するためには，正当なアクセスを記録したアクセスログと攻撃を記録したアクセスログを用意する必要がある．

正当なアクセスを記録したアクセスログとして，メインサーバの過去の運用で記録されたアクセスログを利用する．メインサーバの過去の運用で記録されたアクセスログには大規模な攻撃が含まれていないことを前提にする．また，利用するアクセスログの期間は 90 日間とした．これは法務省の「犯罪の国際化及び組織化並びに情報処理の高度化に対処するための刑法等の一部を改正する法律案」[12] で説明されているように，捜査機関が通信履歴の電磁的記録に係る差押えを行う場合，90 日間を上限とし保全要請を実施する場合があるということを参考に，少なくとも 90 日間はアクセスログを保管していると考えたためである．

表 2. BoNeSi に設定したパラメータ

パラメータ	設定した内容
送信元 IP アドレス数	34,000
1 秒間の送信パケット数	無制限
送信先 URI	ダミー Web サーバ上の全 URI

表 3. 攻撃が記録されたアクセスログ詳細

総アクセス数	攻撃元 IP アドレス種類数	セッション数
375,153	34,000	34,000

本実験では，正当なアクセスが記録されたアクセスログとして宮崎大学情報システム工学科 Web サーバ（以下，情報科 Web サーバ）のアクセスログを利用した．約 1 年間分のアクセスログを 90 日間ごとに区切り，それぞれ Q1 から Q4 とした．情報科 Web サーバのアクセスログ詳細を表 1 に示す．

HTTP-GET Flood 攻撃が記録されたアクセスログは，BoNeSi[13] というソフトウェアを使用して作成する．ローカル環境下に Web サーバを作成し，その Web サーバに対して攻撃することで攻撃が記録されたアクセスログを作成する．本実験では情報科 Web サーバを攻撃緩和の対象として検証を行うため，情報科 Web サーバと同じ Web ページの構造をもつ Web サーバ（以下，ダミー Web サーバ）を作成した上で攻撃し，情報科 Web サーバが攻撃された際のアクセスログを擬似的に作成した．

BoNeSi はボットネットトラフィックをシミュレートし DDoS 攻撃の影響を調査するためのツールであり，異なる IP アドレスからの ICMP，UDP および TCP を使用したフラッド型の攻撃を作成できる．攻撃の種類や規模などを設定するために指定するパラメータが様々あるが，今回は表 2 に表すようパラメータを指定し 35 秒間の攻撃を行った．攻撃の結果記録されたアクセスログの詳細を表 3 に示す．

特定の Web ページへのアクセスした記録をもとに特徴量を求めるため，特定ページに付随する CSS ファイルや画像ファイル，JavaScript ファイルなどは除外した．除外した拡張子は，gif，jpg，png，ico，css，js である．同様の理由で，Web サーバ上のファイルやフォルダを管理するためのプロトコルである WebDAV の使用によるアクセスログと，URI の打ち間違いなどによるエラーアクセスログを除外した．これらの調整を行った後のアクセスログ詳細を表 4 に示す．

表 4. 調整後アクセスログ詳細

	アクセス数	IP アドレス数	セッション数
情報科 Q1	65,312	9,716	35,289
情報科 Q2	70,117	9,763	34,558
情報科 Q3	67,110	10,449	31,608
情報科 Q4	68,196	11,781	33,263
攻撃	313,466	33,997	33,997

4.2 検証方法

4.1 節で説明したアクセスログをもとに，挙動差による判別手法とエントロピーによる判別手法それぞれの判別精度を確認し，提案する判別手法の有効性を検証する．

準備したアクセスログをセッションに分割し，アクセス回数 2 以上のセッションを集めたグループ (以下, セッショングループ 1) と，アクセス回数 1 のセッションを集めたグループ (以下, セッショングループ 2) に分ける．

挙動差による判別手法の精度は，セッショングループ 1 をもとに K-分割交差検証を行って算出する．K-分割交差検証は，データセットを K 個に分割し，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い，次に別のデータをテストデータとして選択し，残りのデータを訓練用データとして再度精度の算出を行うという処理を K 回繰り返し精度の平均をとることで，未知のデータに対する推定の精度を測る方法である．

エントロピーによる判別手法の精度は，セッショングループ 2 のそれぞれのセッションがもつページアクセスの情報量を，あらかじめ正当なアクセスログをもとに計算したエントロピーと比較して判別を行い測定する．

4.3 実験システム

実験システムには以下のような機能を実装しており，図 4 に示す流れで判別手法の精度測定を行う．

- セッションに分割する機能
  正当なアクセスを記録したアクセスログと攻撃を記録したアクセスログを読み込み，それぞれをセッションに分割する．
- セッションから特徴量を抽出する機能
  セッショングループ 1 の各セッションから，3.1 節で述べた特徴量を特徴ベクトルとして抽出する．
- 挙動差による判別手法の精度測定を行う機能
  特徴ベクトルをもとに K-分割交差検証により判別精度を測定し，結果を出力する．
- エントロピーを計算する機能
  正当なアクセスログに記録されたアクセス先ページの情報をもとにエントロピーを計算する．
- エントロピーによる判別手法の精度測定を行う機能
  セッショングループ 2 の各セッションをエントロピーをもとに判別して判別精度を測定し，結果を出力する．

挙動差による判別手法を検証する機能の実装には Python のオープンソース機械学習ライブラリである scikit-learn[14] を使用した．また，機械学習の学習アルゴリズムは scikit-learn が公開している学習アルゴリズム選択チートシートを使用して LinearSVC を選択した．

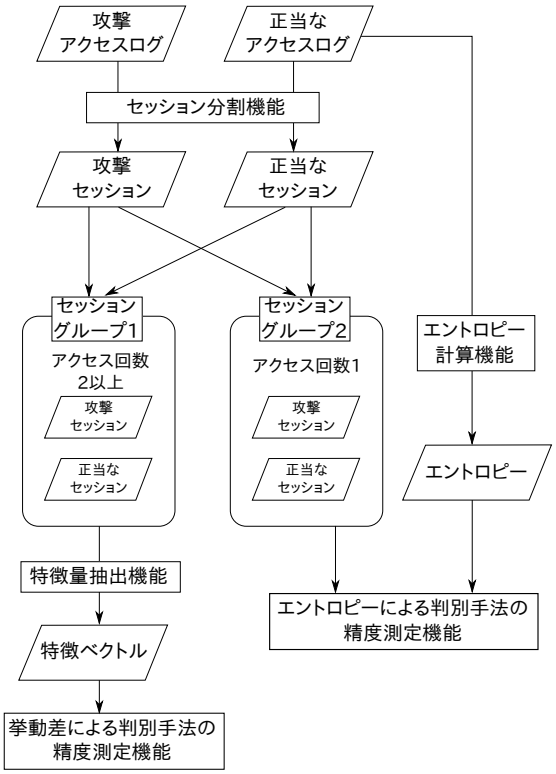

図 4. 実験システム処理の流れ

表 5. セッショングループ内訳

	セッショングループ 1	セッショングループ 2
情報科 Q1	8,828	26,461
情報科 Q2	8,905	28,540
情報科 Q3	8,914	25,361
情報科 Q4	9,020	26,968
攻撃	26,461	2,202

4.4 判別精度の評価

挙動差による判別手法とエントロピーによる判別手法の精度を，4.3 節で用意した実験システムにより測定し，4.4.1 節で説明する指標で評価する．正当なアクセスログは，情報科アクセスログ Q1 から Q4 の 4 つあるので，計 4 回実験を行い，それぞれで精度を測定する．また，用意したアクセスログをグループ分けした結果，表 5 に示すような内訳になった．

表 5 からわかるようにセッショングループ 1 において，正当なアクセスを記録したアクセスログから得られたセッションの数と比べて，攻撃を記録したアクセスログから得られたセッションの数が多すぎるため，バランスの偏りが精度に影響を及ぼさないよう攻撃のセッションをアンダーサンプリングし，Q1 から Q4 それぞれの場合と，攻撃のセッション数が同じになるようにした．

4.4.1 評価指標

判別手法の精度評価で用いる指標として Accuracy，Precision および Recall を用いる．それぞれの値は次式によって求める．

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

表 6. 挙動差による判別手法の評価結果

	Accuracy	Precision	Recall
Q1	95%	99%	90%
Q2	95%	99%	91%
Q3	96%	99%	92%
Q4	95%	99%	91%
平均	95%	99%	91%

表 7. エントロピーによる判別手法の評価結果

	Accuracy	Precision	Recall
Q1	63.20%	99.76%	60.29%
Q2	56.63%	99.74%	53.42%
Q3	58.40%	99.69%	54.96%
Q4	59.35%	99.69%	56.19%
平均	59.40%	99.72%	56.22%

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

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

式中の TP，FP，FN，TN はそれぞれ True Positive，False Positive，False Negative，True Negative を表し，True/False は判定結果が正しかったか，Positive/Negative はどちらに判定したかを表す．本研究では正当なアクセスを判別することが目的なので，正当なアクセスだと判定した場合に Positive，それ以外だと判定した場合に Negative となる．

Accuracy は判定結果全体と正解ラベルがどれくらい一致しているかを判断する指標である．また，Precision は真だと判定したものの中で実際に真であるものの割合を表し，Recall は実際に真であるものの中で判定も真であったものの割合を表す．

本実験では，攻撃を正当なものだとする誤検知を低く抑えることができるかを重視するため，Precision が高い値をとる結果が望ましい．

4.4.2 挙動差による判別手法の評価結果

識別器のハイパーパラメータについては，損失関数をヒンジロス，正則化項を L2 正則化，ペナルティ項 C を 100 に指定した際に精度が一番高くなった．その時の各評価指標の値を表 6 に示す．

この結果から，挙動差による判別手法では情報科アクセスログ Q1 から Q4 の全てにおいて Accuracy 95%で判別できることがわかった．また，Precision は 99%となり高い数値を得ることができた．

4.4.3 エントロピーによる判別手法の評価結果

エントロピーによる判別手法でセッショングループ 2 を判別した際の評価結果を表 7 に示す．エントロピーによる判別手法でも Precision が平均 99.72%と高い割合で判別できることがわかった．しかし，正当なアクセスに対して厳しい条件となるので Recall は約 56%にとどまった．

4.5 考察

正当なセッション判別手法の検証実験では，Precision が約 99%，Recall が約 91%という結果になった．Precision は本実験の場合，正当だと判別したセッションのうち実際に正当なセッションであった割合を表すので，正当だと判別したセッションのうち 99%は正しく，攻撃セッションを正当なセッションだと誤検知してしまったものが 1%であったことを示す．Recall は本実験の場合，正当なセッションのうち正しく正当なセッションだと判別できた割合を表すので，正当なセッションのうち 91%を正しく判別し，9%を見逃したということを示す．この結果から，正当なセッション判別手法では，攻撃を正当なものだとする誤検知の割合を低く抑えつつ，正当なセッションを高い割合で判別できるため，IDS の誤検知を軽減し正当なユーザのサービスを担保するという目的を果たすことができると考える．

また，アクセス回数 1 のセッション判別の検証実験では，Precision が約 99.72%，Recall が約 56.22%という結果になった．アクセス回数 1 のセッションは正当なセッションと攻撃セッションで挙動の差を得ることができないが，今回のようなランダムにページを選択してアクセスするような攻撃では，ページアクセスのエントロピーを使用することで高い Precison で正当なセッションを判別できることがわかった．

今回の実験では検証できなかったことがいくつかある．第 1 に，HTTP-GET Flood 攻撃緩和システムでは IDS に攻撃だと検知されたアクセスのアクセスログを用いて正当なセッション判別を行うが，今回判別の対象としたアクセスログは IDS を通していないアクセスのアクセスログであり，IDS を通した場合の精度は測定できていない．

第 2 に，検疫サーバでは一定量のアクセスを受け取り，判別に使用するためのアクセスログをとるが，どれくらいのアクセスを取れば正当なセッション判別が可能か検証できていない．電子商取引で使用されるミッションクリティカルなサーバなど，短時間のサービス停止が大きな損害へ繋がるため，判別に必要な最低限のアクセス量を検証することは重要である．

また，提案した判別手法ではボットネットシミュレータの BoNeSi を使用して攻撃を再現し，攻撃の特徴を学習した．BoNeSi はランダムに選択した Web ページへアクセスするようなボットネットの攻撃を再現しているが，ボットネットからの攻撃は，同じページに繰り返しアクセスするようなものや，あらかじめ決めた順番どおりにアクセスするようなものなど様々なアクセスの仕方が考えられるため，BoNeSi の挙動とかけ離れた挙動を示す攻撃に対して判別できるか検証が必要である．

5. おわりに

本論文では，我々が行ってきた HTTP-GET Flood 攻撃緩和システムの検疫サーバで，正当なアクセスを判別する手法について提案した．正当なアクセスを判別するため，アクセスをセッションというまとまりに分割し，そこから人と攻撃で差が出そうな挙動を示す特徴量を抽出し判別に利用した．複数種類の特徴量から正当なセッションと攻撃セッションの微々たる差を発見するためその判別には機械学習を利用した．

実運用により記録されたアクセスログと HTTP-GET Flood 攻撃シミュレーションツールにより作成したアクセスログを使用して学習フェーズにおいて識別器を学習させ，交差検証により検証実験を行った．その結果，アクセス回数 2 以上のセッ

ションの場合，提案した判別手法で Accuracy が約 95%という精度で判別でき，十分な有効性があることを示した．判定の困難なアクセス回数 1 のセッションについては，ページアクセスのエントロピーを使用することで，Recall はある程度犠牲になるが高い Precision で判別できることがわかった．

今回は，検疫サーバでの正当なアクセス判別手法に焦点を当て研究を行ったが，今後は，提案した判別手法を HTTP-GET Flood 攻撃緩和システムに組み込み，運用時に有効に機能するかや，検疫サーバでどれくらいのアクセスログをとれば十分に判別できるか検証を行う必要がある．また，本研究で使用した攻撃ツールは 1 種類であり，そのツールと挙動の異なる攻撃では判別できない可能性があるため，複数種類の攻撃の挙動を記録したアクセスログを作成し，判別できるか検証が必要である．

参考文献

1) 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).

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

3) Takeshi Yatagai， Takamasa Isohara and Iwao Sasase: Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior, in Proceedings IEEE Pacific Rim Conference on Communications，Computers and Signal Processing， pp.232-235(2007).

4) 高橋朝英， 田口元貴， 小林良太郎， 加藤雅彦: 仮想計算機のリソース制御による HTTP-GET Flood 攻撃対策，電子情報通信学会論文誌 D， Vol.J94-D， No.12, pp.2058-2068(2011).

5) 吉田祥真， 三上烈史， 小林良太郎， 金岡晃， 加藤雅彦: 複数台のおとりマシンによる HTTP-GET Flood 攻撃対策，情報科学技術フォーラム講演論文集，Vol.11， No.4, pp.207-210(2012).

6) Mizuki Watanabe， Ryotaro Kobayashi and Masahiko Kato: HTTP-GET Flood Prevention Method by Dynamically Controlling Multiple Types of Virtual Machine Resources，Journal of Information Processing, Vol.23， No.5， pp.655-663(2015).

7) Dirt Jumper Ver.5 Technical Security Notes， (online), avalable from (https://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/research_DirtJ5.pdf)(accesed 2017-01-26).

8) 小池泰輔， 梅澤猛， 大澤範高: ランダムフォレストアルゴリズムを用いたネットワーク侵入検出システムの性能解析，第 76 回全国大会講演論文集， pp.619-620(2014).

9) Qin Liao， Hong Li， Songlin Kang and Chuchu 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).

10) 小宅宏明，宮地玲奈，川口信隆，重野寛，岡田謙一: 機械学習によるネットワーク IDS の false positive 削減手法，情報処理学会論文誌，Vol.45，No.8，pp.2104-2112(2004).

11) Welcome! - The Apache HTTP Server Project, (online)， avalable from (https://httpd.apache.org/)(accesed 2017-01-17).

12) 犯罪の国際化及び組織化並びに情報処理の高度化に対処するための刑法等の一部を改正する法律案，(オンライン)， 入手先 (http://www.moj.go.jp/content/000001552.pdf)(参照 2017-01-17).

13) GitHub - Markus-Go/bonesi: BoNeSi - the DDoS Botnet Simulator，(online)， avalable from (https://github.com/markus-go/bonesi)(accesed 2017-01-17).

14) scikit-learn: machine learning in Python, (online)， avalable from (http://scikit-learn.org/stable/)(accesed 2017-01-17).

15) Choosing the right estimator scikit-learn 0.18.1 documentation, (online)， avalable from (http://scikit-learn.org/stable/tutorial/machine_learning_map/)(accesed 2017-01-26).