クライアントのHTTPリクエスト送信動作に着目したボット検知手法

藤　竜成[a)]・山場　久昭[b)]・油田　健太郎[c)]・岡崎　直宣[d)]

On a Bot Detection Method Focused on HTTP Request Transmission Behavior of Clients

Ryusei FUJI, Hisaaki YAMABA, Kentaro ABURADA, Naonobu OKAZAKI

Abstract

Attack detection comes to be more important because distributed denial of service (DDoS) attacks cause serious disruption to web servers. In addition, we have to achieve a further task to tell a new phenomenon called flash events and DDoS attacks apart. In recent years, the emergence of flash events, which are spikes in network traffic from legitimate users, has been seen frequently. Flash events are often recognized as DDoS attacks by mistake because of their large amount of traffic and numerous clients. We need to react to them in a different way, though their behaviors are similar. To realize this, we must identify the factors of them. In this paper, we propose a bot detection system that focuses on the HTTP request transmission behavior of clients to identify increases in traffic and clients. In an evaluation experiment, we evaluated the detection accuracy using datasets of flash events and DDoS attacks.

Keywords: bot similarity, DDoS attack, flash event

1. はじめに

DDoS(Distributed Denial-of-Service) 攻撃は Web サービスに対して大量の不正なトラフィックを送りつけることにより Web サービスのサービス提供を妨害する攻撃で深刻な被害を引き起こす。DDoS 攻撃の規模や頻度が著しく増加している [1] ことから、DDoS 攻撃の検知システムの重要性が増している。一般に、DDoS 攻撃はボットネットと呼ばれるボットに感染したコンピュータによって構成されるネットワークを通じて実行される。ボットはボットネット管理者の攻撃命令を受信することにより Web サービスに対して DDoS 攻撃を実行する。

近年、人間によって引き起こされるフラッシュイベントと呼ばれる DDoS 攻撃に類似した現象が発生している。典型的な DDoS 攻撃とフラッシュイベントの概略図を図 1 に示す。フラッシュイベントの実例としては、2016 年 2 月にインドの Ringing Bells 社によって発表された格安スマートフォン「Freedom 251」の予約受付が Web サイト上で開始された。しかし、その格安スマートフォンを求め何百万の人々が同時に Web サイトにアクセスしたためサーバのクラッシュを引き起こした [2]。

DDoS 攻撃は Web サービスを提供しているサーバを過負荷状態に追い込む可能性がある。また、DDoS 攻撃と同様にフラッシュイベントもサーバを過負荷状態に追い込む可能性がある。DDoS 攻撃を引き起こす要因はボットであるため、Web サービスへのアクセスを禁止する必要があるが、フラッシュイベントを引き起こす要因は人間であるため、アクセスを禁止するのではなくサーバのスケールアウトやスケールアップ等を実施し、可用性の低下を防ぐよう措置を講じる必要がある。フラッシュイベントは大量のトラフィックやクライアントが観測される点で DDoS 攻撃と類似している。そのため、到達している全体のトラフィックだけに着目する場合フラッシュイベントを DDoS 攻撃であると誤検知する可能性がある。また、DDoS 攻撃とフラッシュイベントの識別を行わず、高可用性を優先するとコストの増加に繋がる。なぜならば、本来、サービスの提供が不要な DDoS 攻撃に対してもサービスを提供するからである。よって、DDoS 攻撃とフラッシュイベントの識別は重要であり、DDoS 攻撃とフラッシュイベントの検知後において、それぞれ別の対応を行う必要がある。そのために、大量のトラフィックやクライアントの要因が何であるかを特定しなければならない。

そこで、DDoS 攻撃時にはボットを検知し、フラッシュイベント時には人間を発見する検知システムが必要となる。

一般に DDoS 攻撃時においてボットはサーバやネットワーク等のリソースの枯渇を目的として大量のトラフィックを継続的に送りつける。それに対して、フラッシュイベント時において人間は、ある特定の行動を目的とするため、多くの場合、大量のトラフィックを送りつけることはない。また、同じボットネットに属するボットは、攻撃を実行するためのプログラムが予めインストールされている。このプログラムはボットネット内では同一であると考えられる。そのため、ボット同士の挙動が類似することが考えられる [3]。そこで本研究ではクライアントの HTTP リクエストの送信動作に着目したボッ

[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

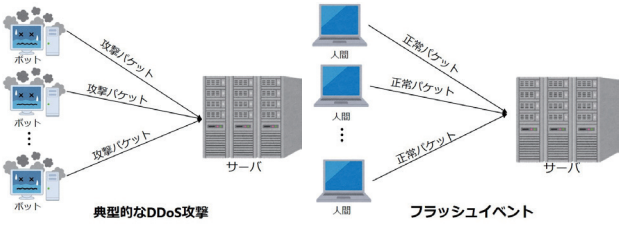

図 1. 典型的な DDoS 攻撃とフラッシュイベント

ト検知手法を提案する。本手法は以下の観点によってボットの検知を行う。

1. クライアント毎の単位時間当たりのリクエスト量
2. Web ページに対するリクエスト送信間隔の類似性
3. 大量のリクエスト送信の継続性

本手法によりクライアント毎に人間或いはボットかの識別が可能となる。すなわち、DDoS 攻撃時にはボットを検知し、フラッシュイベント時にはフラッシュイベントのトラフィックを DDoS 攻撃と誤検知せず、人間を発見することが可能となる。評価実験では検知精度について実験を行い本手法の有用性について議論する。

2. 関連研究

文献 [4] で小島らは、パケットが持つ送信元 IP アドレスや宛先ポート番号等の 9 つのヘッダ情報からエントロピー系列を求め、小島らが提案する多次元マハラノビス距離法 (EMMM) によりトラフィックが異常 (攻撃) であるか否かを判定している。しかし提案された手法は、フラッシュイベントの発生を考慮していないためフラッシュイベントを異常 (攻撃) とみなしてしまい、誤った判断を招く可能性がある。

文献 [5] で Bhatia は、単位時間当たりのパケット数やサーバの CPU 使用率等の特徴とこれらの変化を検知する技術を用いて、到達しているトラフィックが DDoS 攻撃或いはフラッシュイベントかを判定している。しかし提案された手法は、クライアント単位での判定が出来ず、DDoS 攻撃発生時に人間のアクセスまでも攻撃とみなしてしまう可能性がある。

文献 [6] で Saravanan らは、flash crowd 攻撃と呼ばれるフラッシュイベント間に実行される DDoS 攻撃に対応するため、フローの類似性、アクセスしたページ、クライアントの正当性等を用いてクライアントがボットか人間かを判定している。しかし DDoS 攻撃が発生した際にフローの類似度の計算量が爆発的に増加する問題やフローの類似性によるボットの検知をかい潜るような攻撃が行われた際にボットを正しく検知することが出来ない問題がある。

3. 提案手法

本手法では、サーバに到達するトラフィックを単位時間 U_T 毎に解析しサーバにアクセスしているクライアントがボットであるか否かを判定する。提案手法のボット検知処理の流れを図 2 に示す。

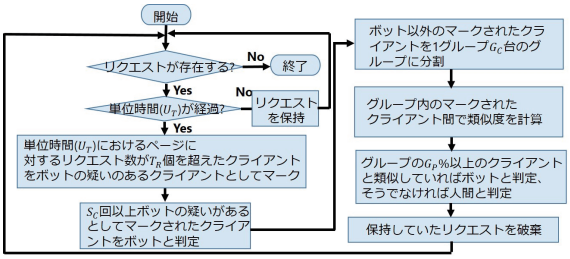

図 2. 提案手法のボット検知処理の流れ

3.1 クライント毎の単位時間当たりのリクエスト量

一般に DDoS 攻撃時においてボットは、サーバやネットワーク等のリソースの枯渇を目的として大量のリクエストをサーバへ送信する。そこで、あるクライアントが単位時間 U_T 内に C_{RN} 回 Web ページに対するリクエストを送信したとすると、式 (1) を満たす時、このクライアントをボットの疑いのあるクライアントとしてマークする。ここで、T_R はクライント毎の単位時間当たりのリクエスト量に関するしきい値を表す。

$$C_{RN} \geq T_R \tag{1}$$

3.2 Web ページに対するリクエスト送信間隔の類似性

一般に DDoS 攻撃はボットネットを通じて実行され、同じボットネットに属するボットは攻撃を実行するためのプログラムが予めインストールされている。このプログラムはボットネット内では同一であるためボット同士の挙動が類似することが考えられる。そこで、マークされたボットの疑いのある全てのクライアントにおいて、Web ページに対するリクエストの送信間隔の確率分布を求め、ボットの疑いのあるクライアント間で確率分布の類似度を計算する。今回は Web ページのファイルの拡張子を htm ファイルと html ファイルとし、それらのファイルに対するリクエストの送信間隔を用いる。これは htm ファイルや html ファイルが一般に Web ページを表現するためのファイルとして用いられており、Web ページに対するリクエストは、クライアントが意図しないと送信されないためである。また、Web ページ以外のリソースに対するリクエストを送信間隔の計算に含めた場合、リクエスト送信間隔の類似性を適切に表現できないと考えたため、今回は Web ページ以外のリソースに対するリクエストは類似度の計算には使用しないことにした。リクエスト送信間隔の単位は秒で、秒以下の値は切り捨てる。

例えばあるクライアントの Web ページに対するリクエストの送信間隔として $x_1, x_2, ..., x_n$ が観測されたとすると、式 (2) により Web ページに対するリクエスト送信間隔の確率分布 $p(x)$ を求める。

$$p(x) = \frac{cnt(x)}{n} \tag{2}$$

ここで $cnt(x)$ は送信間隔 x が観測された回数である。

ボットの疑いのある全てのクライアントにおいて Web ページに対するリクエスト送信間隔の確率分布を求めた後、確率分布間の類似度の計算を行う。確率分布間の類似度の計算に

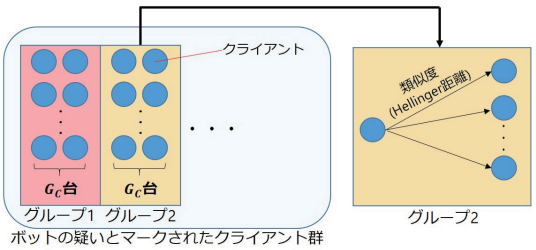

図 3. グループ内での類似度の計算

は Hellinger 距離を用いる。Hellinger 距離は式 (3) で定義される。

$$D_H(p(x), q(x)) = \frac{\sqrt{\sum_x (\sqrt{p(x)} - \sqrt{q(x)})^2}}{\sqrt{2}} \tag{3}$$

ここで $D_H(p(x), q(x))$ が取りうる値の範囲は $0 \leq D_H(p(x), q(x)) \leq 1$ である。確率分布 $p(x)$,$q(x)$ が非常に近い確率分布である時、$D_H(p(x), q(x))$ は 0 に近づき、確率分布 $p(x)$,$q(x)$ が全く異なる確率分布である時、$D_H(p(x), q(x))$ は 1 に近づく。本研究では、式 (4) を満たした際にこのクライアント同士は類似していると判定する。ここで、T_H は Web ページに対するリクエスト送信間隔の確率分布間の類似度に関するしきい値を表す。

$$D_H(p(x), q(x)) \leq T_H \tag{4}$$

Saravanan らの手法では、DDoS 攻撃が発生した際にボットの疑いがあるとしてマークされたクライアント間の全ての組み合わせでフローの類似度の計算を行っている。しかし、ボットの台数が増加すると爆発的に計算量が増加する問題がある。そこで本手法ではマークされたボットの疑いのあるクライアント群を 1 グループ当たり G_C 台に分割し、そのグループ内の全ての組み合わせで確率分布間の類似度の計算を実行するようにした (図 3)。そして、グループ内の G_P% 以上のクライアントと類似していると判定されればこのクライアントをボットと判定し以降のアクセスを禁止する.

3.3 大量のリクエスト送信の継続性

基本的なボットの検知はクライアント毎の単位時間当たりのリクエスト量と Web ページに対するリクエスト送信間隔の類似性を用いて行う。しかし攻撃者がリクエストの送信間隔をランダムにするなど、Web ページに対するリクエスト送信間隔の類似性による検知をかい潜るような攻撃を意図的に行った場合、クライント毎の単位時間当たりのリクエスト量と Web ページに対するリクエスト送信間隔の類似性だけではボットを正しく検知することが出来ない。そこで大量のリクエストを継続して送信するクライアントをボットと判定することにした。例えばあるクライアントが C_{SC} 回ボットの疑いのあるクライアントとしてマークされたとすると、式 (5) を満たしたときそのクライアントをボットと判定し、以降のアクセスを禁止する。

$$S_C \leq C_{SC} \tag{5}$$

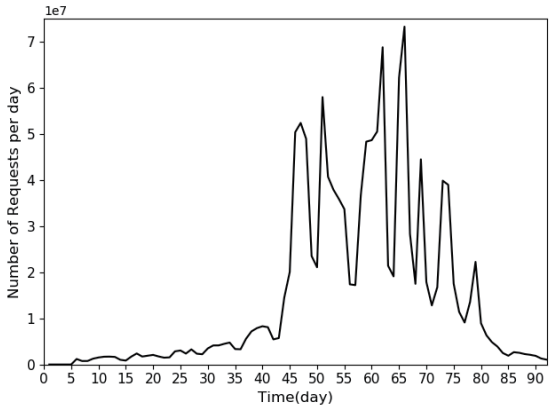

図 4. 1998 FIFA World Cup のリクエスト量の推移

3.4 しきい値 T_R、T_H の決定

クライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R は、通常時のサーバのログを用いて決定する。最初に各単位時間毎にユーザ 1 人あたりのリクエスト数の平均を求め、それらの平均を μ_R とする。同様に各単位時間毎にユーザ 1 人あたりのリクエスト数の標準偏差を求め、それらの平均を σ_R とする。次に式 (6) により、しきい値 T_R を決定する。

$$T_R = \mu_R + \alpha_R * \sigma_R \tag{6}$$

ここで、α_R が取り得る値の範囲は $0 \leq \alpha_R$ であり、α_R を大きくするにしたがい、クライアントがボットの疑いのあるクライアントとしてマークされにくくなる.

Web ページに対するリクエスト送信間隔の確率分布間の類似度に関するしきい値 T_H は DDoS 攻撃時のトラフィックを用いて式 (7) により決定する。

$$T_H = \mu_H + \alpha_H * \sigma_H \tag{7}$$

ここで μ_H は DDoS 攻撃時のボット間における Web ページに対するリクエスト送信間隔の確率分布の類似度の平均を表し、σ_H は DDoS 攻撃時のボット間における Web ページに対するリクエスト送信間隔の確率分布の類似度の標準偏差を表す。ここで、α_H が取り得る値の範囲は $0 \leq \alpha_H$ であり、α_H を大きくするにしたがい、クライアント同士が類似していると判定され易くなる。α_R と α_H を式 (6) と式 (7) にそれぞれ設けたのは、ネットワークやサーバ等の状態を考慮してしきい値 T_R と T_H を柔軟に決定出来るようにするためである。

4. 評価

提案手法の検知精度を実験によって評価し、本手法の有用性について議論する. 実験データとしてフラッシュイベント時のデータセットと DDoS 攻撃時のデータセットを用いる。

4.1 フラッシュイベント時のデータセット

フラッシュイベント時の実験用のデータセットとして 1998 FIFA World Cup[7] を用いる。1998 FIFA World Cup は 92 日間のデータセットであり、フラッシュイベント時のデータセットとして広く利用されている。このデータセットは各地に設置されたサーバが受信した 1,352,804,107 個の HTTP リ

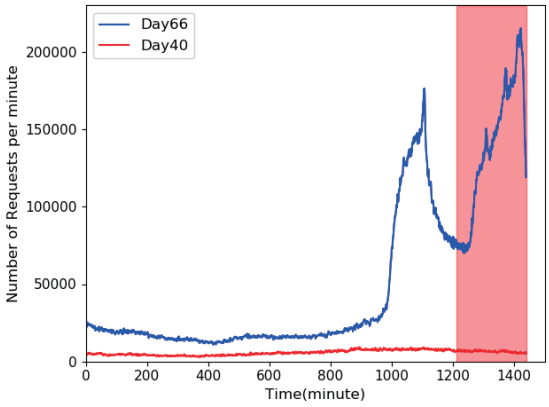

図 5. 40 日目と 66 日目におけるリクエスト量の推移

クエストを含んでいる。このデータセットの 92 日間におけるリクエスト数の推移を図 4 に示す。横軸が時間 (日) で、縦軸がサーバが受信したリクエスト数を表す。評価実験には 1998 FIFA World Cup 66 日目の約 230 分間を用いた (図 5 の赤い領域)。これは図 4 から読み取れるように 66 日目に最大のリクエスト数が観測されており、うち実験で利用した約 230 分間のデータに、1 分間あたりの最大のリクエスト数と急激なリクエスト数の増加が観測されていたためである。この約 230 分間に 63,337 台のクライアントがサーバにアクセスしており、約 200 万個の Web ページに対するリクエストが含まれていた。

4.2 DDoS 攻撃時のデータセット

DDoS 攻撃時の実験用のデータセットとして Bonesi[8] と呼ばれるボットネットシミュレータを利用して DDoS 攻撃を行った際に取得したトラフィックを用いる。DDoS 攻撃を行うボットの台数は 30,000 台とし DDoS 攻撃のトラフィックを 60 秒間取得した。取得したトラフィックの中には、Web ページに対するリクエストが約 90 万個含まれていた。また、1998 FIFA World Cup に含まれている htm ファイルや html ファイルをボットがアクセスする Web ページとして設定した。

本実験において各パラメータの値は $U_T = 60$、$S_C = 3$、$G_C = 10$、$G_P = 60$、$T_R = 4$、$T_H = 0.3$、とした。ここで U_T、S_C、G_C、G_P は経験的に定めた。また、T_R の値は通常時のサーバのログとして 1998 FIFA World Cup の 40 日目を用いて計算した。40 日目を用いたのは図 5 からフラッシュイベントが発生しておらず、通常時のサーバのログとして利用できると考えたためである。T_H の値は Bonesi によって DDoS 攻撃を行った際に取得したトラフィックからランダムに選択した 10 台のボットのトラフィックを用いて計算した。

4.3 提案手法の検知精度

提案手法の検知精度の指標として Detection Rate(DR) と False Positive Rate(FPR) を用いる。DR はボットをボットとして正しく検知した割合を示し、式 (8) で定義される。また、FPR は人間をボットとして誤って検知した割合を示し、式 (9) で定義される。DR の評価には Bonesi を用いて作成した DDoS 攻撃時のデータセットを用い、FPR の評価にはフラッシュイベント時のデータセットである 1998 FIFA World Cup 66 日目の約 230 分間を用いた.

表 1. 各記号の定義

記号	定義
True Positive(TP)	ボットをボットとして検知した数
True Negative(TN)	人間を人間として判定した数
False Positive(FP)	人間をボットとして検知した数
False Negative(FN)	ボットを人間として判定した数

表 2. 提案手法の検知精度

DR	FPR
0.93	0.04

$$DR = \frac{TP}{TP + FN} \tag{8}$$

$$FPR = \frac{FP}{TN + FP} \tag{9}$$

式 (8)、式 (9) における各記号の定義を表 1 に示す。

提案手法の検知精度を表 2 に示す。表 2 から提案手法の DR は 0.93 で FPR は 0.04 となっており、十分な検知精度を持っていると考えられる。今回の実験では 60 秒間の DDoS 攻撃のデータセットを用いたため、DR の値はボットの 93%がクライアントの Web ページに対するリクエスト送信間隔の類似性の観点によってボットと検知されたことを意味する。また、今回の実験では $S_C = 3$ と設定しているため、180 秒以上の DDoS 攻撃のデータセットを用いると提案手法の DR は 1 に近づいていくことが考えられる。提案手法の FPR は 0.04 であるが、これは人間がボットの疑いとマークされた回数が $S_C = 3$ 以上となったため、ボットと判定されたからである。

5. まとめ

本研究ではクライアントの HTTP リクエストの送信動作に着目したボット検知手法の提案を行った。本研究の提案手法により、DDoS 攻撃時にはボットを検知できフラッシュイベント時にはトラフィックを DDoS 攻撃であると誤検知せずに人間を発見することが可能となる。評価実験ではフラッシュイベント時と DDoS 攻撃時のデータセットを用いて検知精度を評価した。評価実験の結果、提案手法は十分な検知精度を持っていることが分かった。今後の課題としては、今回の実験ではパラメータ U_T、S_C、G_C、G_P を経験的に定めたが、これらのパラメータを適切に設定する仕組みが必要である。また、現在の検知システムではクライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R をぎりぎり下回るように DDoS 攻撃を実行された場合、ボットを正しく検知することが出来ない。このような DDoS 攻撃を検知するための特徴の選択や仕組みが必要である。

参考文献

1) “DDoS 攻撃規模は 5 年で 12 倍に増加”, ZDNet Japan available at:https://japan.zdnet.com/article/35096332/ (2017) (accessed 2018/02/05).

2) “Freedom 251 website down for second day”, The Hindu available at:http://www.thehindu.com/sci-tech/technology/gadgets/

freedom-251-website-down-for-second-day/article8257501.ece (2016) (accessed 2018/02/05).

3) D. Acarali, M. Rajarajan, N. Komninos, and I. Herwono: Survey of approaches and features for the identification of HTTP-based botnet traffic, Journal of Network and Computer Applications 76 pp.1-15 (2016).

4) 小島 俊輔、中嶋 卓雄、末吉 敏則: エントロピーベースのマハラノビス距離による高速な異常検知手法, 情報処理学会論文誌 Vol.52 No.2 pp.656-668 (2011).

5) S. Bhatia: Ensemble-based model for DDoS attack detection and flash event separation, Future Technologies Conference (2016).

6) R. Saravanan, Y. Shanmuganathan, and Y. Planichamy: Behavior-based detection of application layer distributed denial of service attacks during flash events, Turkish Journal of Electrical Engineering & Computer Sciences 24 pp.510-523 (2016).

7) 1998 World Cup Web Site Access Logs available at:http://ita.ee.lbl.gov/html/contrib/WorldCup.html(accessed 2018/02/05).

8) GitHub - Markus-Go/bonesi: BoNeSi - the DDoS Botnet Simulator available at:https://github.com/Markus-Go/bonesi(accessed 2018/02/05).