

ページアクセスの挙動解析に基づいた HTTP-GET Flood 攻撃の検知手法

谷田貝 健 磯原 隆将 笹瀬 巖

慶應義塾大学理工学部情報工学科
223-8522 神奈川県横浜市港北区日吉 3-14-1

{yatagai, isohara, sasase}@sasase.ics.keio.ac.jp

あらまし 近年, Web サーバに対するサービス拒否 (DoS:Denial of Service) 攻撃や分散型サービス拒否 (DDoS:Distributed DoS) 攻撃による被害が深刻となっている. これらの攻撃の中には, 正常な通信プロトコルで多量の通信を発生させる攻撃もあり, サーバ側での正常な通信と攻撃を見分けることが難しく, 既存のセキュリティ対策技術であるウイルス検知システム (VDS:Virus Detection System) や侵入検知システム (IDS:Intrusion Detection System) では検知できないといった問題がある. DoS 攻撃の中でも, HTTP プロトコルの GET メソッドを悪用した HTTP-GET flood 攻撃による被害が発生しており, 実際に HTTP サーバがダウンし, サービスの遅延やサーバの停止といった被害が発生している. そこで本論文では, HTTP サーバのアクセスログから, ページのアクセス挙動を解析することで, 上記のような HTTP-GET flood 攻撃を検知する方式を提案する. これは, HTTP サーバのアクセスログから, 1) コンピュータウイルスなどからの攻撃では, 攻撃端末が同じ振る舞いを持つことに注目し, 共通する閲覧順序を持つクライアントを攻撃クライアントとみなす手法, 2) 本来のクライアントは閲覧ページの情報量が大きい場合に閲覧時間が長くなることに注目して, 情報量に依存しない閲覧時間のアクセスを攻撃とみなす手法の 2 通りから構成される. 実際に運用されている HTTP サーバのアクセスログを用いた評価により, HTTP-GET flood 攻撃を迅速に検知できることを確認し, 提案方式が本来のクライアントと機械的なアクセスを見ける手法として有効であることを示す.

Detection Technique of HTTP-GET Flood Attack Based on Analysis of Page Access Behavior

Takeshi Yatagai Takamasa Isohara Iwao Sasase

Dept. of Info. & Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku, Yokohama Kanagawa,
223-8522, Japan

{yatagai, isohara, sasase}@sasase.ics.keio.ac.jp

Abstract Recently, there are many denial-of-service (DoS) attacks by computer viruses or botnet. DoS attacks to the Web service are called HTTP-GET flood attack and threats of them increase day by day. In this type of attacks, the malicious clients send a large number of HTTP-GET requests to the target HTTP server. Since these HTTP-GET requests have legitimate formats, an intrusion detection system (IDS) can not detect them. In this paper, we propose HTTP-GET flood attack detection techniques based on analysis of page access behavior. We propose two detection algorithms, one is focusing on a browsing order of pages and the other is focusing on a correlation with browsing time to page information size. We implement the proposed scheme and evaluate the attack detection rates, i.e., false positive and false negative. The results show that our techniques can detect the HTTP-GET flood attack effectively.

1 はじめに

近年, コンピュータウイルスやボットネットからの HTTP-GET flood 攻撃に対する脅威が高まっている. 例えば Trojan.Sientok や Gaobot からの HTTP-

GET flood 攻撃が確認されている [1]-[2] が, こうした攻撃クライアントは HTTP プロトコルの GET メソッドを悪用し, 標的となる HTTP サーバに対して大量の GET 要求パケットを送信する [3]-[4]. これらの攻撃パケットは, 正式な HTTP プロトコル

に従って送信されるため、サーバ側では正規クライアントからの GET 要求パケットと攻撃クライアントからの攻撃パケットを識別することができず、全ての要求パケットを正当な要求として処理することになる。そのため、サーバは自身のリソースを使い果たしてしまい、正規クライアントへのサービスの遅延、停止を招く。HTTP-GET flood 攻撃を行う手法は、以下の 4 種類に分類される。

コンピュータウイルスによる攻撃 悪意のユーザが作成したコンピュータウイルスに感染した各端末が、大量の通信を発生させることで、1 対 1 の DoS 攻撃を仕掛ける。コンピュータウイルスによる攻撃では、攻撃先 URL や攻撃間隔、攻撃期間などの攻撃パラメータがウイルスの作成者によって事前に決められている。こうしたウイルスによる攻撃の場合、ウイルスに感染した端末の所有者が駆除するまで攻撃が慢性的に続く場合が多い。最近では P2P ソフトを経由して拡散したウイルスによる DDoS 攻撃の被害が発生している。

ボットによる攻撃 悪意のユーザにより作成されたボットと呼ばれる不正プログラムに感染した端末がボットネットと呼ばれる攻撃ネットワークを形成し、DoS 攻撃を仕掛ける。悪意のユーザはボットネット内の感染端末に対して攻撃を指令することで 1 対 n の DDoS 攻撃を仕掛ける。前述したウイルスによる攻撃との違いとして、ボットによる攻撃では、攻撃者が攻撃パラメータを随時変更できるため、攻撃先となる URL や攻撃間隔に変化をつけることができ、より動的な攻撃が可能である点が挙げられる。[2]。

DoS ツールによる攻撃 DoS 攻撃用のツールがインターネット上に公開されている。匿名掲示板などで協力者を募ることで DDos 攻撃に発展することがあり、突発的な攻撃が発生する。DoS ツールによる攻撃では、攻撃パラメータは攻撃者により任意に決定される。

F5 アタックによる攻撃 Web ブラウザでは、「F5」キーにページの更新機能が割り当てられている。このキーを連打し、必要以上にページのリロードを繰り返すことで HTTP-GET flood 攻撃を仕掛けることができる。F5 アタックは専門的な知識や特別なツールを必要としないため、簡単に攻撃を仕掛けることができる。F5 アタックによる攻撃では、攻撃パラメータは攻撃者により任意に決定される。

現在、ネットワークにおける攻撃への対策として、侵入検知システム (IDS: Intrusion Detection System) が導入されている。IDS はネットワークを流れるパケットと、シグネチャと呼ばれる攻撃パターンを比較することで攻撃検知を行う [5]。しかしながら、HTTP-GET flood 攻撃では、正当な HTTP プ

ロトコルを用いて攻撃が行われるため、IDS での攻撃検知が困難となる問題がある。

また、HTTP-GET flood 攻撃への対策として、トラフィック制御や攻撃検知を行うモジュールが提案されている [6]-[8]。トラフィック制御では HTTP サーバの許容量を超えるようなトラフィックを遮断することにより、サーバの過負荷を回避する。しかし、トラフィック制御を用いた場合、攻撃パケットのみでなく、正規利用者からの要求パケットに対しても制限を加えることとなり、結果的に正規利用者へのサービス障害を招くといった問題点がある。また、HTTP-GET flood 攻撃の検知モジュールでは、閾値判定により攻撃検知手法を行い、1 秒間に同じ URL へのアクセスを繰り返すクライアントを攻撃クライアントとして検知する。しかしながら、この方式では、攻撃パラメータを変更することで容易に検知を回避されてしまい、誤検知率が高くなるといった問題点がある。

そこで、本論文ではクライアントからのページアクセスの挙動解析に基づく HTTP-GET flood 攻撃の検知方式を提案する。本方式は、HTTP サーバのアクセスログから、1) コンピュータウイルスなどからの攻撃では、攻撃端末が同じ振る舞いを持つことに注目し、共通する閲覧順序を持つクライアントを攻撃クライアントとみなす手法、2) 本来のクライアントは閲覧ページの情報量が大きな場合に閲覧時間が長くなることに注目して、情報量に依存しない閲覧時間のアクセスを攻撃とみなす手法の 2 通りから構成される。実際に運用されている HTTP サーバのアクセスログを用いた評価により、HTTP-GET flood 攻撃を迅速に検知できることを確認し、提案方式が本来のクライアントと機械的なアクセスを見ける手法として有効であることを示す。以下、2 章で提案方式について説明し、3 章で特性評価を行う。最後に 4 章でまとめる。

2 提案方式

本研究では、HTTP サーバへ向けられた GET メソッドのうち、HTTP-GET flood 攻撃を検知する方法を提案する。提案方式はネットワークのゲートウェイにおいて HTTP サーバへ向けたページアクセスの挙動を解析し、攻撃を検知した場合はゲートウェイの段階で遮断する。提案方式の概念図を図 1 に示す。

2.1 アクセス順序の重複検知方式

ウイルスまたはボットネットに感染した端末による HTTP-GET flood 攻撃では、あらかじめ設定された URL に対して攻撃を行うため、同じウイルスに感染した端末からの攻撃があった場合、同じ振る舞いを行うページへのアクセスが連続的に観測される。この特徴を利用して、Web ページに向けたアクセスのうち、各送信元 IP アドレスについてページの閲覧順序を解析し、共通するアクセス順序を探し出

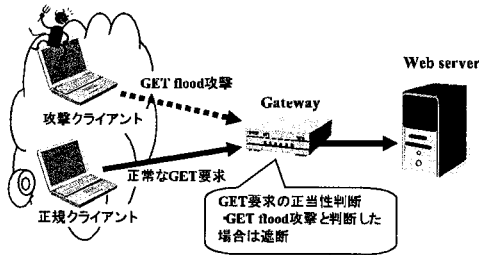


図 1: 提案方式の概念図

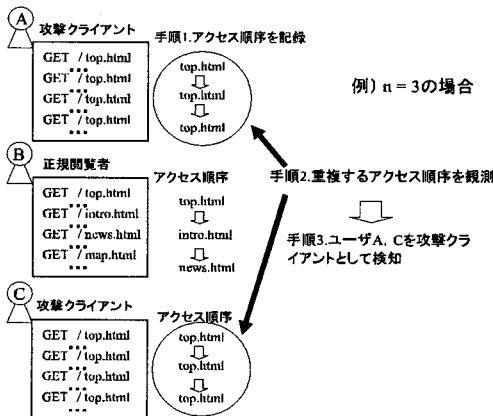


図 2: アクセス順序の重複検知方式の概要

すことで HTTP-GET flood 攻撃の検知を行う、アクセス順序の重複検知方式を提案する。アクセス順序の重複検知方式の概要を図 2 に示す。まず、図 2 の手順 1 に示すように、各送信元 IP アドレスのアクセスログから、一定量の各クライアントのページの閲覧順序を求める。次に図 2 の手順 2 に示すように、共通する閲覧順序を持つクライアントを検索する。同じ閲覧順序を持つクライアントが複数確認された場合、これらクライアントからのアクセスは HTTP-GET flood 攻撃の可能性があると判断し、これら送信元 IP アドレスからの以降のアクセスを遮断する。

2.2 ページ内情報量と閲覧時間の相関に注目する検知方式

正常な通信において Web ページを閲覧する場合、ページ内に含まれる情報量が増えるに従い、閲覧時間も増加すると考えられる。一方、HTTP-GET flood 攻撃はページ内の情報量とは無関係に Web ページにアクセスを行う。そこで、各送信元 IP アドレスについて一定量の閲覧ページ、閲覧時間を記録し、ページ内に含まれる情報量との相関に注目すること

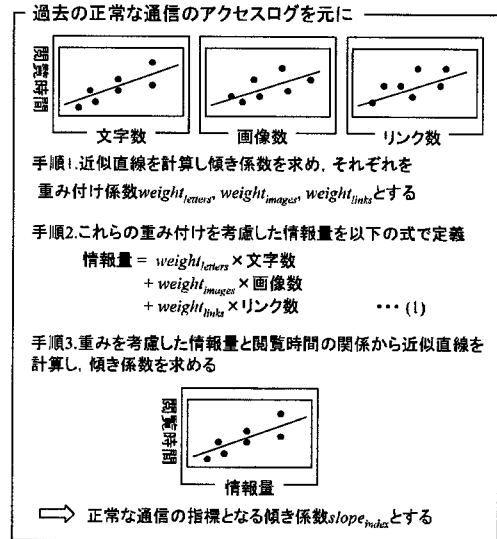


図 3: 正常な通信の指標の算出の流れ

で HTTP-GET flood 攻撃の検知を行う方式を提案する。ページ内の情報の要素としては、文字数、画像数、リンク数を考慮し、各要素による重み付けを行った値を情報量と定義する。本方式では、攻撃の検知に先立ち、過去の正規閲覧者からのアクセスログを解析することで、閲覧時間と情報量の傾き係数を求め、この値を正常な通信の指標 ($slope_{index}$) として用いる。攻撃検知では、HTTP サーバへ向けられたアクセスを送信元 IP アドレスごとに分類し、情報量と閲覧時間の相関を求め、傾き係数を算出し、 $slope_{index}$ よりも傾きが小さいものを攻撃として検知する。以下に提案方式の動作を示す。

2.2.1 正常な通信の指標の算出

正常な通信の指標の算出の流れを図 3 に示す。まず、図 3 の手順 1 に示すように、過去の正規閲覧者からのアクセスログから、閲覧ページと閲覧時間を求め、各閲覧ページに含まれる文字数、画像数、リンク数と閲覧時間との関係を示す。各要素ごとに、最小二乗法を用いて要素と閲覧時間の関係についての近似直線を求め、近似直線の傾きを重み係数 $weight_{characters}$, $weight_{images}$, $weight_{links}$ とする。次に、重みを考慮した情報量を以下の式で定義する。

$$\text{情報量} = weight_{characters} * num_{characters} + weight_{images} * num_{images} + weight_{links} * num_{links} \quad (1)$$

ここで、 $num_{characters}$, num_{images} , num_{links} は各ページ内に含まれる文字数、画像数、リンク数である。最後に、式 (1) で表される情報量と閲覧時間の関係から最小二乗法を用いて近似直線を求め、近似直線の傾きを正常な通信の指標 ($slope_{index}$) とする。

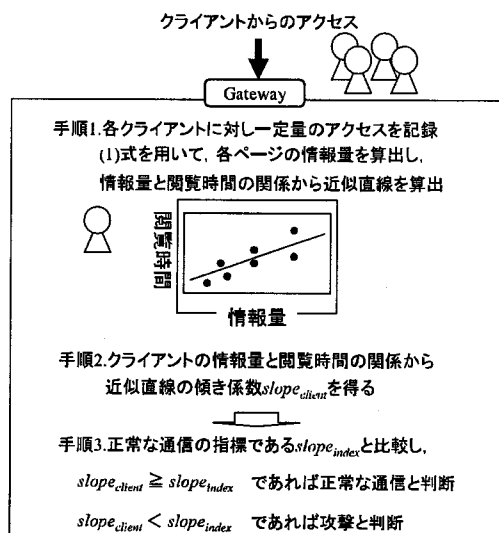


図 4: ページ内情報量と閲覧時間の相関に注目する検知方式の概要

2.2.2 攻撃検知の動作

ページ内情報量と閲覧時間の相関に注目した攻撃検知の概要を図 4 に示す。まず、図 4 の手順 1 に示すように、各送信元 IP アドレスについてのアクセスログから閲覧したページと閲覧時間を求める。次に図 4 の手順 2 に示すように、クライアントごとに式 (1) を用いて各ページ内に含まれる情報量を計算し、閲覧時間との関係から最小二乗法を用いて近似直線を計算し、近似直線の傾き係数 ($slope_{client}$) を求める。求めた $slope_{client}$ と $slope_{index}$ を比較し、 $slope_{client} \geq slope_{index}$ の場合は正常な通信と判断し、 $slope_{client} < slope_{index}$ の場合は攻撃として検知する。しかし、記録したアクセスがすべて 1 種類の URL へのアクセスであった場合は、情報量と閲覧時間の相関が取れないため、例外とみなす。記録したアクセスが全て 1 種類の URL へのアクセスであった場合、それらのアクセス時間の平均値を求める。平均値が正規閲覧者からのアクセス時間の最小値よりも小さい場合、その送信元 IP アドレスからのアクセスを HTTP-GET flood 攻撃として検知する。

3 評価

本節では、アクセス順序の重複検知方式、ページ内情報量と閲覧時間に注目した検知方法を用いた場合の誤検知率の評価を行う。誤検知としては以下に示す 2 種類について評価する。

- False Positive: 正常な通信を HTTP-GET flood 攻撃と判断してしまう

表 1: 評価用トラフィックデータの概要

	正常通信	GETflood 攻撃	合計
送信元 IP アドレス数	164 個	9 個	173 個
要求数	2429 回	28110 回	30539 回

- False Negative: HTTP-GET flood 攻撃を正常な通信と判断してしまう

3.1 評価データ

本評価ではトラフィックデータとして、筆者の管理下にある実運用中の HTTP サーバに対して、実際に HTTP-GET flood 攻撃を仕掛けた際のトラフィックを Wireshark [9] を用いてキャプチャリングしたデータを用いる。トラフィックデータの概要を表 1 に示す。攻撃を仕掛けるウイルスとして、約 0.03 秒間隔で GET 要求を行う Trojan.Sientok.A, 約 0.25 秒間隔で GET 要求を行う Netsky.Q, 約 0.137 秒間隔で GET 要求を行う BlueCode.Worm の 3 種類のウイルスを用いた [10]。また、F5 アタックのトラフィックは著者自身が F5 ボタンを連打することで再現し、DoS ツールは初期設定である 0.20 秒間隔、ボットのトラフィックは 5 つの URL に対し 5 秒間隔でのアクセスを繰り返すトラフィックを発生させた。

3.2 アクセス順序の重複検知方式に対する評価

本評価では、攻撃検知に用いるアクセス数を変化させた場合の誤検知率を評価した。図 5 にアクセス順序の重複検知方式を用いた場合の誤検知率を示す。図 5 より、4 つ以上のアクセスを記録した場合、False Positive は約 1% 程度の値をとり、False Negative は 22% 近くの値をとっていることがわかる。False Positive は、同じ閲覧順序を持つクライアントが複数いたために引き起こされた誤検知であった。本評価で用いたホームページでは、大学での講義用資料配布ページが用意されており、資料を閲覧を目的としたクライアントは同じ閲覧順序を持つ結果になってしまった。また、False Negative の内訳は、F5 アタックや DoS ツールによる攻撃を検知できなかったことによる誤検知であった。F5 アタックや DoS ツールを用いた攻撃の場合、攻撃者が攻撃先となる URL を任意に決定できるため、共通するページの閲覧順序に注目した方式では攻撃を検知することができなかった。以上のことから、アクセス順序の重複検知方式では、正規クライアントを攻撃と誤検知する確率を低く抑えることができるが、攻撃を見逃す確率も高くなってしまったことがわかった。

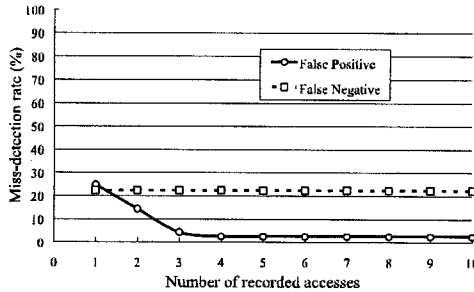


図 5: アクセス順序の重複検知方式における誤検知率

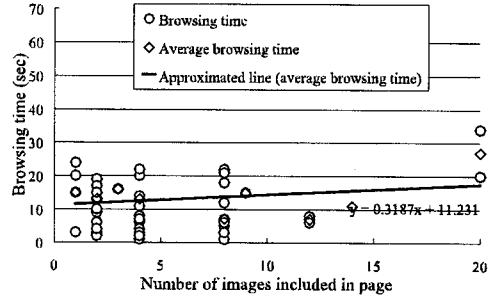


図 7: ページ内に含まれる画像数と閲覧時間の関係

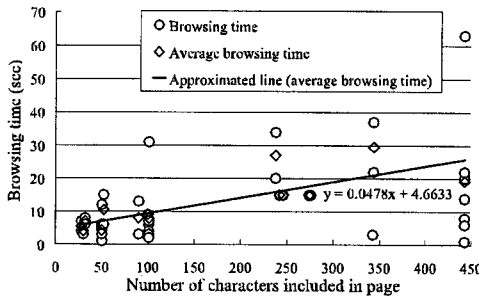


図 6: ページ内に含まれる文字数と閲覧時間の関係

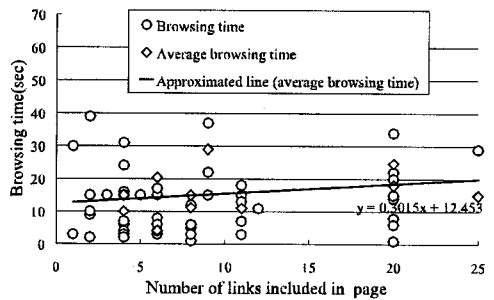


図 8: ページ内に含まれるリンク数と閲覧時間の関係

3.3 ページ内情報量と閲覧時間の相関に注目する検知方式に対する評価

3.3.1 正常な通信の指標の算出

本評価では、1週間分のアクセスログを解析することで $slope_{index}$ を算出した。ページ内に含まれる文字数、画像数、リンク数と閲覧時間との関係を示したグラフをそれぞれ図 6, 7, 8 に示す。

これらの図より、近似直線が全て右上がりになっているため、文字数、画像数、リンク数と閲覧時間には相関があると考えられる。また、これらの近似直線の傾き係数がそれぞれ 0.0436, 0.3187, 0.4705 であることから、式 (1) は以下になる。

$$\begin{aligned} \text{情報量} = & 0.0436 * num_{characters} \\ & + 0.3187 * num_{images} \\ & + 0.4705 * num_{links} \end{aligned} \quad (2)$$

以降、本研究では情報量を式 (2) で定義し、式 (2) で計算される情報量と閲覧時間の関係から、正常な通信の指標 $slope_{index}$ を求める。図 9 に式 (2) を用いて算出した重み付き情報量と閲覧時間の関係を示す。

図 9 より、近似直線の傾きが 1.0664 であることから、 $slope_{index} = 1.0664$ とし、以降この値を正常な通信の指標として用いる。

3.3.2 攻撃検知における誤検知率の評価

図 10 にページ内情報量と閲覧時間の相関に注目する検知方式を用いた場合の誤検知率を示す。図 10 を見ると、5 つ以上の GET 要求を解析した場合、False Negative は 0% をとり、全ての攻撃を検知できていることがわかる。しかし、False Positive は 9% 近くの値になっており、正常な通信のうち 1 割近くを遮断することになる。遮断する正常な通信は、短時間で複数のページを閲覧したことにより情報量と閲覧時間の相関が弱まった場合や、相関は見られるものの $slope_{index}$ よりも低い値をとったクライアントからの通信を攻撃と誤検知したためである。以上のことから、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、高い確率で HTTP-GET flood 攻撃を検知できるが、正規クライアントも多く誤検知されてしまうことがわかった。

4 おわりに

本研究では、HTTP サーバに対する各クライアントからのアクセスを解析することで、HTTP-GET flood 攻撃を検知する手法を提案した。提案方式では、ページの閲覧順序が重複するようなアクセスを攻撃と検知する閲覧順序の重複検知方式と、閲覧時間とページ内に含まれる情報量との相関の解析によ

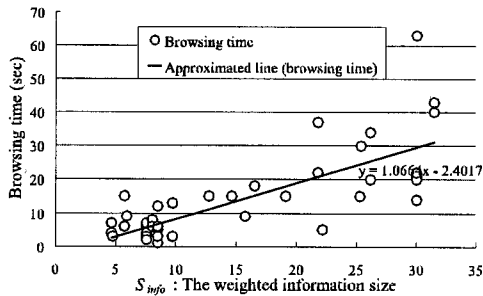


図 9: 重み付けを行った情報量と閲覧時間の関係

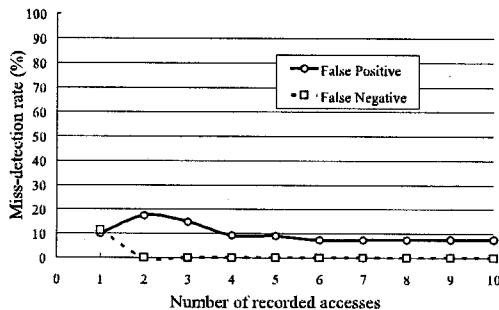


図 10: ページ内情報量と閲覧時間の相関に注目する検知方式における誤検知率

り攻撃を検知するページ内情報量と閲覧時間に注目した検知方式の2通りの方式でHTTP-GET flood攻撃の検知を行った。また、実運用されているホームページのアクセスログを用いた誤検知率の評価から、閲覧順序の重複検知方式では4つ以上のアクセス解析により正常な通信の誤検知を抑えながら、約78%の攻撃を検知でき、特にクライアントへのサービスを優先させる場合に有効であることがわかった。また、ページ内情報量と閲覧時間に注目した検知方式では5つ以上のアクセスを解析する場合で正常な通信の誤検知を約9%に抑えながら、全ての攻撃を検知することができ、ページ内情報量と閲覧時間の相関に注目した検知方式は、攻撃検知を優先させる場合に有効であることがわかった。

謝辞

本研究を進めるにあたり、有益な助言を頂いたKDDI研究所の竹森敬祐氏に、深く感謝する。

参考文献

- [1] http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2006-012815-0103-99
- [2] 分析レポート Gabot, 警察庁
http://www.cyberpolice.go.jp/detect/pdf/report_gabot.pdf
- [3] S.Byers, A. D. Robin and D. Korman, "Defending Against an Internet-Based Attack on Physical World", *ACM Transactions on Internet Technology*, vol.4 No.3, August 2004, Page 239-254.
- [4] Juan M. Estevez-Tapiador, Pedro Garcia-Teodoro and Jesus E. Diaz-Verdejo, "Detection of Web-based attacks through Markovian protocol parsing", *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium*, 27-30 June.2005, Page 457-462
- [5] Chie Ishida, Yutaka Arakawa, Iwao Sasase and Keisuke Takemori, "Forecast techniques for Predicting Increase or Decrease of Attacks Using Bayesian Inference", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, August 2005, Page 1088-1192.
- [6] Apache, mod_access_limit,
<http://mm.apache.or.jp/pipermail/apache-users/2002-January/000879.html>
- [7] Apache, mod_bwshare,
<http://www.topology.org/src/bwshare/README.html>
- [8] Apache, mod_evasive,
http://www.zdziarski.com/projects/mod_evasive/
- [9] Wireshark,
<http://www.wireshark.org/>
- [10] Wei Zhou Lu, and Shun Zheng Yu, "An HTTP Flooding Detection Method Based on Browser Behavior", *IEEE Computational Intelligence and Security, 2006 International Conference on*, Volume 2, Nov.2006, Page 1151-1154.