HTTP リクエストの送信間隔分布に着目した仮説検定による異常検知

落水壱歩[†1]　堀良彰[†2]　櫻井幸一[†2]

ウェブアプリケーションの普及に伴い，ウェブサーバへの攻撃は多様化してきている．ウェブアクセスに対する異常検知の方法として，ウェブページの遷移を隠れセミマルコフモデルで扱う手法がある．しかしこの方式ではウェブページ間の遷移確率などを算出する必要があり，動的に生成されるページには適用困難である．本研究ではウェブサーバへの不正アクセス検知のために，HTTP リクエスト送信間隔分布の比較による異常検知を提案した．比較には確率分布の同一性を検定するコルモゴロフ・スミルノフ検定を用いた．定常データとそれぞれの模擬攻撃データの送信間隔分布に相違点が生じたことから異常挙動の検出を確認できた．

Anomaly Detection by Hypothesis Testing Focused on HTTP Request Interval Distribution

Kazuho Ochimizu,[†1] Yoshiaki Hori[†2] and Kouichi Sakurai[†2]

The attacks to Web servers are becoming more diverse with popularization of the Web applications. A detection method employing Hidden semi-Markov Model is one of the anomaly based detection measure for illegal access. However, this method has one weakness that this method can not applied to dynamically generated Web pages. In this work, we proposed anomaly based detection method for abnormal access detection by comparison of the the HTTP request interval distribution. For the comparison, we used Kolmogorov-Smirnov test. In this test, we can confirm the detection of anomaly behavior as we can observe the difference of the distribution between normal data and anomaly data.

1. はじめに

近年，ウェブサーバへの攻撃が問題となってきている．ウェブサーバとはHTTP やHTTPS などの通信プロトコルを用いて，HTML 文書やオブジェクトをウェブブラウザなどに提供するプログラム，及びそのプログラムが動作するコンピュータのことである．代表的なプログラムに Apache HTTP Server と呼ばれるものがある[1)]．

今日では様々な情報がウェブを介してやり取りされている．ウェブを用いることでオンラインで買い物を行ったり，動画を見たり，SNS などでコミュニケーションを取り合ったりすることが可能となっている．これはウェブアプリケーションと呼ばれるアプリケーションソフトウェアを用いることで提供されている．ウェブアプリケーションは Java や Perl などの言語で記述されている．

ウェブが普及した理由としてはクライアントがウェブブラウザを用意するだけでサービスを享受できる，ウェブアプリケーションの更新自体が簡単であるなどの点があげられる．またウェブブラウザによって多少の差はあるものの，どの環境においてもほぼ同様のコンテンツを表示させることが可能であるということも利点の一つである．

このようにウェブは便利な技術である．しかし，使用されている技術には多くの場合，脆弱性が存在する．その脆弱性をつくことでウェブサーバに対して攻撃を行い，ウェブサービスを利用困難にしたり，個人の情報を漏えいさせたりすることが可能となってしまう場合がある．そのような攻撃に対してはパッチやモジュールなどでの対応が行われるが，それだけでは日々見つかる脆弱性に対応しきれない．

また，通常，ウェブサービスはHTTP と呼ばれるプロトコルを用いてデータのやり取りを行なっている．しかしやり取りされる文書や画像などのデータが一つのプロトコルだけで転送されているため，通信の内容が正常なのか異常なのか判断するのが困難になっているという問題点もある．

以上のような状況であるため通信内容を観測するのではなく，その通信の振る舞いを観測することで正常なのか異常なのかを判断できる手法が必要となる．

†1 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, School of Engineering, Kyushu University
†2 九州大学大学院システム情報科学研究院情報学部門
Department of Informatics, Faculty of Information Science and Electrical Engineering, Kyushu University

本研究ではウェブサーバのアクセスログを解析することで異常検知を行うことを考えた．ログファイルを用いた解析では新しく機器を追加する必要がないため導入しやすいという利点がある．ログの中でも HTTP リクエストの送信間隔に着目して攻撃を検知するために色々な視点から解析を行う．攻撃を検知するために次の 2 点を目標にして本実験を行った．まず 1 つ目として一般ユーザの挙動の共通性を見つけることである．通常，人がウェブサーバにアクセスする場合には個人差によりいろいろなパターンがあり得る．そこで通常状態を定義することが大切である．2 つ目は一般ユーザと攻撃の際の挙動の差を示すことである．これを示すことでこの着目点に基づいた方式で本当に攻撃を検知することが可能になるのかを確かめる．

2. DoS(Denial of Service) 攻撃

DoS 攻撃とはネットワークを通じた攻撃の一つである．攻撃対象のコンピュータやルータに対して，トラフィック量を増やして必要な処理を増大させたり，不正なデータを送ったりすることでネットワークを停止させてしまう攻撃のことである．

前者の攻撃は大量のリクエストを送るだけの単純なものであるが，リクエストの送信後に通信を意図的に中断させることで待機状態のリクエストを大量に作り出し，リソースを消費させる方法もある．後者の攻撃ではアプリケーションの脆弱性を狙うことで少量のリクエストでもサーバに大量のリソースを消費させることができる．最近の例としては Apache Killer と呼ばれる Apache HTTP Server の脆弱性をついた攻撃が有名である[2]．この攻撃はプロセスの肥大化を行い，実メモリを消費させることで OS 自体を機能不全に陥らせる攻撃ツールである．すでに Apache HTTP Server ではこの攻撃に対する対策は取られているが，このような脆弱性が再び見つかった場合にはサーバが脅威にさらされてしまう可能性がある．

DDoS(Distributed Denial of Service) 攻撃

DDoS 攻撃とは DoS 攻撃の一種であり，特定のホストからではなく大量のホストが特定のサーバに対して一斉にリクエストを送信し，サービス不能に陥らせてしまうような攻撃である．このような攻撃を行う際にはボットネットと呼ばれる悪性のネットワークが用いられることが多い．

ボットネットとはマルウェアに感染したコンピュータによって構成される悪性ネットワークのことである．このようなボットネットを構築するようなコンピュータはゾンビクラスタと呼ばれる．このようなゾンビクラスタがボットネットを管理するボットマスターから特定のサーバに対して攻撃を行うように命令を受けると，そのサーバに対して一斉にリクエストを送り込む．サーバは大量のリクエストを処理しきれなくなって停止してしまう．その概要を**図 1** に示す．

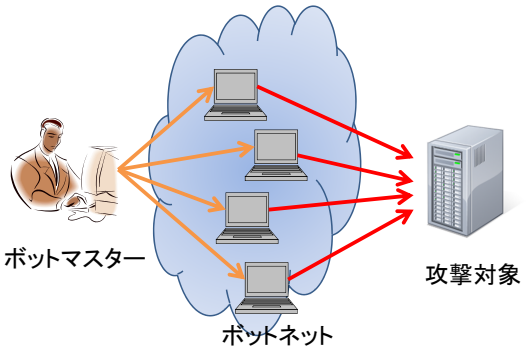

図 1 ボットネットによる攻撃例
Fig. 1 An attack example of a botnet

ボットネットの規模が数千を超えるような場合には特定のホストからの通信を遮断するというような対処が取れない．そのため通常の DoS 攻撃に比べて対処が難しいという問題点がある．

3. 既知の異常検知手法

3.1 侵入検知システム

侵入検知システムとはコンピュータやネットワークなどに対する不正行為を検出し，通知するシステムのことである．このシステムは検査対象を常時観測することで，不正アクセスを検知したり，不正アクセスが行われた場合には通信を遮断したり，管理者に通知したりする．

不正アクセスの対策の最も一般的な方法としてファイアウォールが存在する．ファイアウォールは内部に持つフィルタの設定に従って攻撃と思われる通信を遮断する．ファイアウォールを適切に設定することで既存の攻撃，例えば脆弱性を持っていたアプリケーションが利用していた特定ポート番号への攻撃などを取り除くことができる．

しかしながらファイアウォールですべての攻撃が防げるわけではない．例えば HTTP 通信は通信の際に 80 番ポートをよく利用するが，このポートはファイアウォールでは通常はフィルタリングされることはない．そこで，このポートに対して DoS 攻撃を行った場合などはポート番号ベースでのファイアウォールでは防ぐことはできない．また今日ではウェブアプリケーションの普及に従って，ウェブサーバそのものを直接攻撃するのではなく，ウェブアプリケーションの脆弱性を狙った攻撃が多くなってきている．ウェブアプリケーションは HTTP 通信がベースで動作していることが多く，このような脆弱性への攻撃というのもファイアウォールでは防ぐことができない．

3.2 ブラウジング挙動を基にした異常検知

アノマリベースの異常検知の 1 つの例として，人のブラウジングの挙動を隠れセミマルコフモデルを用いてモデル化する方法がある[3]．隠れセミマルコフモデルの概略を**図 2** に示す．隠れマルコフモデルとは通常のマルコフモデルに加えて以下の 2 つの条件が追加されている．

- 状態遷移の様子を観測することができない．
- 状態遷移間隔に一般分布を用いる．

この手法は隠れセミマルコフモデルにおける状態を一つのウェブページに割り当てる．そして状態遷移確率をページ遷移確率，シンボル出力確率を出力される HTTP リクエストの確率，状態遷移間隔確率をページごとのリクエスト数の確率に割り当てる．このモデルを Forward-Backward アルゴリズムと呼ばれる手法で，事前に取得したトラフィックデータから生成することで，ウェブページの通常状態をモデル化し，異常検知を行う．この手法では複数のパラメタを用いてモデル化を行なっているため，ウェブの挙動をより細かく表現できているといえる．しかし，この手法には JavaScript などで動的にリンクが生成されるようなウェブページの場合には，状態遷移確率などを求めることが出来ず，適用が困難であるという問題点がある．

4. 提 案 手 法

4.1 解 析 手 法

本実験では Perl を用いてアクセスログから HTTP リクエストの送信間隔を算出する．アクセスログ 1 秒単位で時間が記録されいるため，送信間隔が 0 秒であるものも多数存在する．つまり送信間隔が 0 秒であるものは今回の実験においては送信間隔が 1 秒未満であることを示す．計算をする際にはアクセスログに含まれるソース IP アドレスとユーザエージェントの情報をペアとして ID を振る．送信間隔についてはその ID ごとに算出する．ソース IP アドレスだけで ID を割り振ってしまうと，NAT などで構成されているネットワークに対してはすべて同じ端末からの通信とみなしてしまうことになる．そこでより細かく解析を行うためこのような手法を取った．

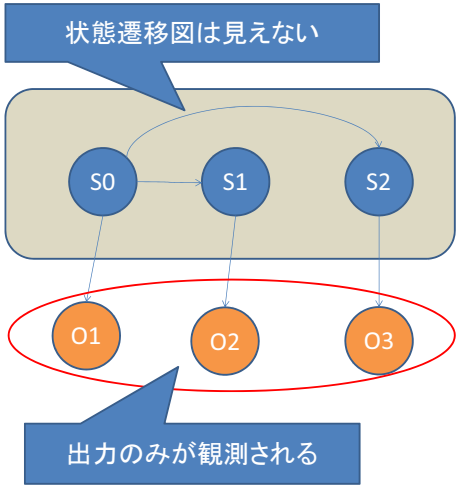

図 2 隠れセミマルコフモデルのブラウジング挙動への割り当て
Fig. 2 Assign a browsing model to Hidden semi Markov Model

HTTP リクエストの送信間隔を計算したあとの解析については R[4] を用いた．R とはオープンソースの統計・解析プログラミング言語である．標準で様々な検定のための関数を含んでおり，解析を行うのには非常に有効な言語である．

解析においては主に以下の 2 つのことを行った．

- HTTP リクエストの送信間隔の平均，分散の算出
- コルモゴロフ・スミルノフ検定による HTTP リクエスト送信間隔分布の比較

4.2 コルモゴロフ・スミルノフ検定

コルモゴロフ・スミルノフ検定とは統計学における仮説検定の一種である[5]．帰無仮説，対立仮説はそれぞれ以下のように定義されている．

- 帰無仮説：2 つの標本の分布は等しい
- 対立仮説：2 つの標本の分布は異なる

検定内容としては有限個の標本に基づいて 2 つの母集団の確率分布が異なるかどうかを判定する．検定においては累積分布関数を調べる．累積分布関数とは確率変数 X が x 以下の値をとる確率であり，以下の式で表現される．

$$F(x) = P(X \leq x) \quad (1)$$

帰無仮説が棄却されない場合において 2 つの母集団の確率分布が等しいと仮定できる．

ここで検定を行う際に注目すべき点として p-value と呼ばれる値がある．p-value とは帰無仮説が正しい時に，観測されたデータ以上に極端な例が得られる確率のことである．つまり p-value を見ることで帰無仮説の信憑性を測ることができる．p-value が小さい場合には 2 つの標本の母集団の確率分布は異なるということができる．p-value が大きいような場合には 2 つの標本の母集団の確率分布は近いということが出来る．

p-value の大小の基準として有意水準と呼ばれるものがある．今回の実験においては有意水準を 5 % に設定して検定を行った．つまり，p-value が 0.05 よりも小さい場合には 2 つの標本の母集団の確率分布は異なるといえ，大きい場合には近いといえる．研究の目的が 2 つの確率分布の同一性を発見することであったため，有意水準をこの値に定めた．

4.3 研究室のウェブサーバのログと模擬攻撃のログとの比較

実験では研究室のウェブサーバのログと模擬攻撃のログとについて比較を行った．今回は模擬攻撃として Apache Bench と呼ばれるウェブサーバの負荷テストツールによる攻撃と，人の手によってウェブページのリロード作業を繰り返す，F5 アタックと呼ばれる攻撃を行った．

4.3.1 グラフを用いた比較

研究室のウェブサーバのログと模擬攻撃（F5 アタック，Apache Bench）のログのそれぞれでグラフを書き，視覚的な差を比較した．グラフを表示させる際には HTTP リクエストの送信間隔を横軸に，その送信間隔で送られているリクエストが何個あるかという情報を縦軸にした．**図 3** は通常のウェブサーバのログのグラフを示している．**図 4** と**図 5** はそれぞれ Apache Bench と F5 アタックを行った際のグラフを示している．

グラフを見てみると違いは存在することがわかる．研究室のウェブサーバのログにおいては送信間隔が 1〜5 秒の間での広く，なだらかに分布しているのがわかる．それに対して他の 2 つの異常データについては値が 0 あたりのみに分布しているのがわかる．これは一般の人がウェブサーバアクセスした場合にはコンテンツの種類などの様々な要因によってページ遷移の時間が前後に左右するためであると考えられる．それに対して Apache Bench や F5 アタックの結果というのは負荷をかけることが目的であるためにページの内容にかかわらず大量のリクエストが短い期間で大量に送られてくるため通常データの場合とは異なって値がほとんど 0 に集中して分布していると考えられる．

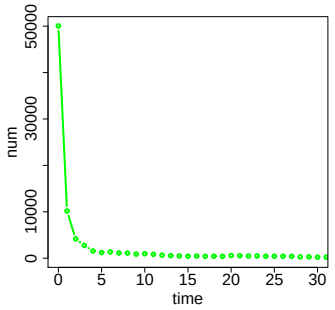

図 3 研究室のウェブサーバのログの送信間隔のグラフ
Fig. 3 HTTP request interval of our lab

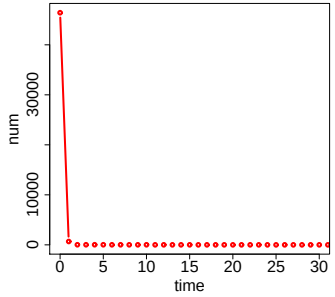

図 4 Apache Bench を用いた場合のウェブサーバのログの送信間隔のグラフ
Fig. 4 HTTP request interval of Apache Bench

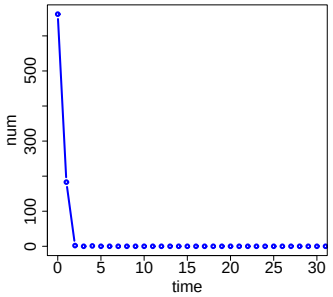

図 5 F5 アタックを行った場合のウェブサーバのログの送信間隔のグラフ
Fig. 5 HTTP request interval of F5 attack

4.3.2 基本統計量を用いた比較

データを数値的に比較するためにログデータの総リクエスト数と平均，分散を算出した．**表 1** に 3 つのデータについての結果を示す．

リクエスト送信間隔の平均については明確な差が見られる．研究室のウェブサーバのログ

表 1 研究室のウェブサーバのログと模擬攻撃のログの基本統計量
Table 1 Basic statistic of our lab's Web server log and simulated attack log

ログの種類	総リクエスト数	平均	分散
研究室	91901	7.033645	247.046
Apache Bench	47320	0.09139899	3.124489
F5 アタック	851	0.4559342	16.09306

の場合には送信間隔の平均が非常に大きくなっている．これは一般のユーザはページの内容を読んだりする時間が必要であるため，リクエストの送信間隔は広範囲に及ぶためだと考えられる．それに対して模擬攻撃のログについては平均はそれぞれ 1 秒を切っており，研究室のウェブサーバのログと比べるとかなり小さくなっている．Apache Bench については F5 アタックよりも更に小さくなっている．これは Apache Bench というソフトウェアでは全く人の手によらずプログラム的な動作でのみアクセスを行うため非常に短期間に大量のリクエストを送っており，リクエストの送信間隔が 0 秒に集中して分布しているためだと考えられる．

リクエスト送信間隔の分散についても，平均の場合とほぼ同様の結果が得られている．

4.3.3 仮説検定による比較

4.2 で述べたコルモゴロフ・スミルノフ検定を用いてそれぞれのデータごとに仮説検定を行った．3 つのデータ (研究室のログ，Apache Bench，F5 アタック) についてそれぞれの組に対して検定を行った．結果としてはすべて p-value は有意水準 5 % を大きく下回ったため，比較した 3 つのデータの母集団の確率分布はすべて異なったと言える．この結果の理由としては 4.3.1，4.3.2 において述べたような差が見られたためであると考えられる．

今回の実験においてはいろいろな攻撃が混ざったログデータではなく，通常のデータとそれ以外のデータとが明確に分かれていたため，解析結果が顕著に現れたと考えられる．本来の解析においては様々なデータが入り混じったログを解析する必要があるため，このような差を明確に算出するのは難しいと考えられる．そのため，次の項目で異常検知を行うための研究室のウェブサーバのログのみに焦点を当てた解析を行う．

4.4 研究室のウェブサーバログの 1 日ごとの比較

研究室のウェブサーバのログと模擬攻撃のログの比較によって通常のユーザのデータと攻撃が行われたデータの間には差があることが確認された．しかし実際に異常検知を行うためには通常ユーザの挙動というものを正確に定義できなくてはならない．人の挙動にはばらつきがあるため同じサーバのログであっても日時や天候などによってデータに差が生まれる．そこでこの実験では 11 月 1 日から 11 月 30 日までのログデータを 1 日ごとに分割して，その各々に対して統計的解析・検定を行う．

4.4.1 グラフを用いた比較

各日のデータに対して送信間隔横軸に，その送信間隔のリクエスト数を縦軸にしてグラフを作成した．得られた 30 のグラフのうち目で確認できる範囲では 2 つのパターンが見られた．2 つのパターンの例として 11 月 13 日と 11 月 19 日のグラフを**図 6** と**図 7** に示す．

図 6 は図 7 に対して送信リクエスト数が全体的に多いことが見て取れる．また図 7 ではリクエスト送信間隔が 0 秒のみにピークが存在しているのに対して，図 6 では 0 秒以外にも幾つかピークが存在する．このような差が生まれた原因を調べるため 11 月 13 日のログデータの中身を確認した．確認した結果，11 月 13 日にはウェブクローラによるアクセスが大量に行われていることがわかった．ウェブクローラとは全文検索型サーチエンジンの検索のためのデータベースを作成する際に用いられる，ウェブページ回収プログラムのことである．つまりこのウェブクローラが実験ウェブサーバ上にあるウェブページをすべて取得していたためグラフが大きく遷移していたと考えられる．このウェブクローラは他にも 11 月 26 日，27 日にも再びアクセスしており，それらの日のグラフも 11 月 13 日のデータと同じように送信間隔が 0 秒以外の所にもピークをが見られた．

つまり，グラフにおける解析で 2 つのパターンが生まれたのは，ウェブクローラという通常のユーザとは関係のないアクセスが発生したためである．このウェブクローラによるアクセスがあった 3 日分 (13 日，26 日，27 日) のデータは，通常ユーザの挙動からは離れているという結果が得られた．

4.4.2 基本統計量を用いた比較

各日のデータに対して解析のために総リクエスト数とリクエスト送信間隔の平均，分散を算出した．結果を**表 2** にまとめている．

まず，各日の解析結果の全体的な変化を見てみる．総リクエスト数については 2000 から 10000 の間を変化していることがわかる．また平均については 2 から 18 程度，分散については 70 から 360 の間で変化している．

次に表中で特徴的な値をとっているものについて考える．主に特徴的なパターンを持っているのは 1 日，13 日，26 日，27 日の 4 日分である．この 4 日を除くデータについては平均はおおよそ 5 以下，分散はおおよそ 210 以下という値をとっている．それに対して 1 日は平均，分散が若干大きくなっているのがわかる．また 13 日，26 日，27 日については平均はすべて 10 を超えていて分散についても 280 を超えている．これは明らかに異なると言

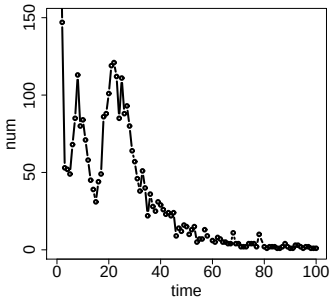

図 6 11 月 13 日の送信間隔のグラフ
Fig. 6 HTTP request interval at 13 Nov.

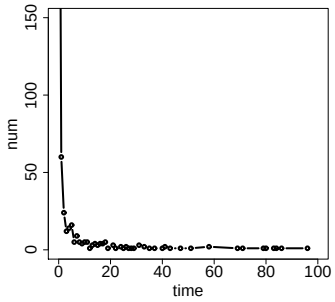

図 7 11 月 19 日の送信間隔のグラフ
Fig. 7 HTTP request interval at 19 Nov.

える．13 日，26 日，27 日のデータについてはグラフでの比較でも述べたようにウェブクローラが確認されており，そのため他の日のデータからは異なった値をとっていることがわかる．1 日のデータについては本研究室のウェブサーバ移行作業によって，2 日移行は古いサーバへのアクセスが現在の実験で用いているサーバにリダイレクトされるようになった．1 日のデータは送信間隔が 1 秒以下のものが他のデータに比べて少なかったが，これは埋め込みオブジェクトの多い私たちの研究室のウェブページが，2 日移行から正式なウェブページとしてアクセスされ始めたためだと考えられる．つまりこの差は攻撃やウェブクローラによるものではなくウェブサーバ移行による，本ウェブページの利用者の変化によるものであったと考えられる．

4.4.3 仮説検定による比較

各日のデータのすべての組み合わせに対してコルモゴロフ・スミルノフ検定を行った．そして各日に対してコルモゴロフ・スミルノフ検定を行った結果，一致すると判定された日の合計を計算した．その結果を**図 8** に示す．

検定の結果として，分布が一致する場合とそうでない場合とで二極化が起きていることわかった．一致している場合には一致数が 10 個付近に分布しているのに対して一致していない場合には 0 個から 2 個の間に分布しているのが確認できた．一致数が 0 個のものについては一般的な挙動と異なる挙動が見られることがわかる．平均，分散の項目で述べた 1 日，13 日，26 日，27 日は一致数が 0 個となっており特別なことが起きていたことがわかる．確かにこの日にはサーバの移行やウェブクローラによるアクセスなどの通常とは異なる活動が

表 2 通常データの日毎の基本統計量
Table 2 Basic statistic of each day for normal data

日付	リクエスト	平均	分散	日付	リクエスト	平均	分散
11/01	4916	7.770079	249.873	11/16	7294	4.20697	146.8923
11/02	6486	3.732022	126.2854	11/17	5582	3.730054	127.1732
11/03	4716	3.491379	127.883	11/18	4307	3.447641	113.9538
11/04	4052	2.292098	69.72499	11/19	2107	3.83871	132.6616
11/05	3491	3.787807	136.3448	11/20	3078	3.746589	150.3416
11/06	3317	4.629914	153.7113	11/21	5050	3.294696	98.91226
11/07	5466	2.92711	92.2551	11/22	3582	4.095019	157.5247
11/08	6707	2.713786	94.8201	11/23	2365	2.640713	75.66516
11/09	7462	5.15387	172.6261	11/24	3630	3.189367	96.97249
11/10	4460	3.159885	107.3116	11/25	5160	5.263027	208.194
11/11	6408	2.358083	73.55055	11/26	6916	17.76283	285.067
11/12	6196	2.495065	78.77273	11/27	6388	13.59378	359.3224
11/13	9594	15.38061	307.8812	11/28	6374	5.638314	180.7112
11/14	6459	3.902909	138.0674	11/29	6866	4.327097	145.9127
11/15	6171	3.433846	105.2792	11/30	6955	4.992427	179.8878

見られた．つまり，この検定の結果を比較することで通常とは異なる挙動を数値化することができているといえる．

また今述べた一致数が少ないものとは逆に一致数が多いものについて考察する．一致数が多いグループについてはだいたい 9 個以上の日が一致している．その中でも 19 日は最も一致数が多く 17 日分のデータがこの日のデータと一致している．このことから 19 日が今回観測した中の通常の人の挙動で一番平均的な動きをしていたのではないかと考えられる．

4.5 定常性を持つデータによる異常検知

4.3 において研究室のウェブサーバのログと模擬攻撃を行った際の差を示した．この比較の際には研究室のウェブサーバのログについてはすべて通常であるという仮定のもとで検知を行った．しかし，4.4 で示したとおり，1 ヶ月のデータについても日毎に分割してみるとデータが変化しているのが分かる．その中にはウェブクローラなどの定常状態とはいえないようなものも含まれており，検知を行う上では適切であるとは言えない．そこで，ここでは 4.4 で述べた定常性を持つと考えられるデータと模擬攻撃のデータを比較する．定常性を持つと考えられるデータの中で最も一致数の多かった 19 日のデータを，1 ヶ月の中で 1 番定常的であると仮定する．

検定を行った結果，p-value は Apache Bench と F5 アタックの両方について有意水準である 5 % を大きく下回っており 19 日のデータとは確率分布が異なっていることが示され

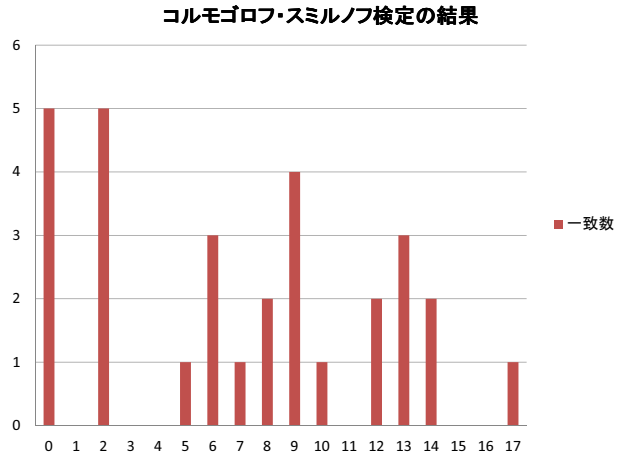

図 8 2011 年 11 月の各日ごとの分布の同一性の検定結果
Fig. 8 The result of the KS test for each day in Nov, 2011

た．今回の実験においては 4.3 においても同様の結果が得られている．この理由としては今回，攻撃として用いたデータは模擬攻撃であり，攻撃の存在を気づかせないような工夫はしていなかったためであると考えられる．

この実験では模擬攻撃のデータがあまり一般的ではないとはいえ，ウェブサーバのログから定常性を持つと考えられるデータを抽出して，そのデータを元に比較を行えば攻撃検知が可能になるということを示した．

5. 結　　論

本研究ではウェブサーバへの不正アクセス検知を行うために，HTTP リクエスト送信間隔分布の比較による異常検知を提案した．グラフと基本統計量，仮説検定を用いることで，ウェブサーバのアクセスログの解析を行った．グラフによる解析では人の目で差を確認する方法であるため，変化が大きな場合には特徴を把握しやすいが，細かい変化については無視することになる．基本統計量と仮説検定を用いた解析では具体的に数値が算出されるため機械的に処理できるが，共通点や相違点がどこにあるのかは具体的にはわからない．このようにそれぞれのメリット・デメリットがあるがそれらを組み合わせることで，ひとつの視点にとらわれない解析を行うことができた．今回，この研究の結果から具体的には以下の２つの項目について確認できた．

- 一般ユーザの挙動は常に変化しているが定常性も存在する．
- 定常性を持つデータと模擬攻撃とでは挙動に差が生まれるため異常検知が可能になる．

この結果から，HTTP リクエストの送信間隔分布を観測することで限定的ではあるが異常検知を行えることを示した．これによりウェブサーバのアクセスログは異常検知を行う上では有用な情報であることがわかった．

今回の実験において検証出来なかった事と，他の視点からの解析を行うための手法について，これらを今後の課題として挙げる．

(1) 研究室のウェブサーバのログの仮説検定による解析において共通点の少ないものについての考察を行う
(2) HTTP リクエストをステータスコード毎に分けてから解析を行う
(3) 解析する間隔をもっと短くし，どの程度まで情報量を減らしても特徴が捉えられるかを調査する

(1) について取り組んでいくことで，今回の実験で得られなかった特徴を発見することにつながると考えられる．また (2) や (3) といった今回の解析とは別の手法でアクセスログを解析することで，異常検知に有効な情報を見つけることにつながるのではないかと考えられる．

謝辞　本研究の一部は，日本学術振興会 科学研究費補助金 基盤研究 C (21500078) による補助のもとで行われた．

参　考　文　献

1) Welcome! - The Apache HTTP Server Project. http://httpd.apache.org/.
2) 「Apache Killer」脆弱性への対応を補強した「Apache HTTP Server 2.2.21」- SourceForge.JP Magazine : オープンソースの話題満載. http://sourceforge.jp/magazine/11/09/15/1020230.
3) Xie, Y. and Yu, S.-Z.: A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors, *IEEE/ACM TRANSACTIONS ON NETWORKING*, Vol.17, No.1, pp.54–65 (2009).
4) The R Project for Statistical Computing. http://www.r-project.org/.
5) J.Crawley, M.: 統計学 : R を用いた入門書，共立出版 (2008).