多数のWebサイトを対象とした攻撃の共起性に基づく検知手法

齊藤 聡美[1,2]　吉岡 克成[1]　松本 勉[1]

概要：近年，Web アプリケーションを利用した Web サイトを対象とした攻撃事例が多数報告されている．本稿では，Web アプリケーションが稼働している複数 Web サイトに対する悪意あるリクエストを，アクセスログから抽出する手法を提案する．提案手法では，複数の Web サイトに対するアクセスログを適用対象とし，リクエスト送信元ホスト・送信先 Web サイト・送信先コンテンツ間の関係性を分析する．これにより，複数の Web サイトに向けて同じコンテンツを要求するリクエストの発生を抽出できる．複数 Web サイト管理者は，こうしたリクエストの発生を検証することで，悪意あるリクエストの発生を突き止めることができる．

キーワード：Web アクセスログ，ログ分析，ネットワーク監視

Detecting Malicious Access based on Co-occurrence among Multiple Websites

Satomi Saito[1,2]　Katsunari Yoshioka[1]　Tsutomu Matsumoto[1]

Abstract: In recent years, websites are often compromised by various cyber attacks. In this paper, we propose a method for extracting requests intended compromising websites under web applications from access log. Our method analyzes relations among source hosts, destination websites and requested contents. For this relation analysis, we can extract multiple requests that shared with different websites. With verifying those requests by website security analysts, they can find out malicious requests occurrence.

Keywords: Web access log, Log analysis and Network monitoring

1. はじめに

近年，多くの Web サイトが Web アプリケーションを利用して運用されている．例えば CMS（Contents Management System，コンテンツ管理システム）では，Web サイト上の管理・編集画面を通して，Web サイトの概観を簡単にカスタマイズしたり，コンテンツを更新したりすることができる．CMS である WordPress[1] や Joomla![2] は，オープンソースとして公開されており，個人・組織を問わず多く利用されている．さらにこれらのアプリケーションの機能を拡張するためのプラグインも多く存在する．

一方で，インターネットに公開されている Web サイトには様々なアクセスが到達する．Web サイトを利用する正規ユーザによるアクセスや検索エンジンによる情報収集目的のアクセス，攻撃を目的とした悪意あるアクセスなどが日々到達している．中でも，CMS が攻撃の対象となった事例が報告されている．あるホスティングサービス上において WordPress を利用した Web サイトが大量に改ざんされた事例（文献 [3]）や，CMS 中のスクリプトの改ざんが継続して発生していることが報告されている (文献 [4])．CMS は Web サイト毎に異なる形態・規模で運用されてい

[1] 横浜国立大学大学院環境情報研究院/先端科学高等研究院
Graduate School of Environment and Information Sciences, Yokohama National University / Institute of Advanced Sciences, Yokohama National University
[2] 株式会社富士通研究所
FUJITSU LABORATORIES LTD.

るものの，アプリケーションは共通のスクリプトから構成されている．そのため，ある攻撃手口が発見されてしまえば，そのアプリケーションを使っている Web サイト全体が攻撃対象となる．

Web サイトに対する悪意あるアクセスの中には，一見通常のアクセスと見分けることが難しいものも存在する．例えば，CMS の設定変更や編集を行うリクエストだけでは，正規ユーザによるリクエストなのか，コンテンツ改ざん等の悪意あるリクエストなのか判断することが難しい．これまで悪意あるリクエストを検知する手段として，リクエスト中の文字列に着目して脆弱性を持つプラグインの存在を調査したり，ライブラリのバージョンを確認したりするリクエストを検知する手段が取られてきた．こうしたリクエストを正規ユーザが送信することは稀であるため，攻撃目的のリクエストとして検知することが可能であった．しかし，正規ユーザも送信するリクエストについては，リクエスト文字列だけでは攻撃か否かを判断することは難しい．

本稿では，Web アプリケーションを対象とした Web サイトに対する悪意あるリクエストを，アクセスログから抽出する手法を提案する．抽出にあたり我々は，複数の異なる Web サイトを攻撃対象とし，Web サイト間で共通のリクエストが送信されることを攻撃の特徴として着目する．Web サイト管理サービスや検索エンジンによるクローリングを除き，正規ユーザは目的・規模の異なる複数 Web サイトに対して，全く同じコンテンツをリクエストすることは考えにくい．こうしたリクエストに，正規ユーザが Web サイト管理のために送信されるリクエストが含まれているならば，抽出したリクエストは Web サイトへ攻撃目的のリクエストであると判断できる．

提案手法は，複数の Web サイトを管理する Web ホスティングサービス管理者が，自身のサービス監視を行うため，アクセスログの分析を行う際に適用することを想定する．提案手法では，複数の Web サイトに対するアクセスログを適用対象とし，アクセスログから，リクエスト送信元ホスト・送信先 Web サイト・送信先コンテンツ間の関係性を分析する．複数の送信元ホストから複数の Web サイトに対してリクエストされた 2 種類のコンテンツを「共起性を持ってリクエストされたコンテンツ」として取り出し，コンテンツをノードとするネットワークグラフを作成する．作成されたネットワークグラフでは，複数の異なる Web サイトを対象として，これらの Web サイト間で共通の名前を持つコンテンツに対して送信されたリクエストが，ノードが密に接続された形状で表現される．このネットワークグラフの特徴を計測することで，攻撃目的のリクエストの候補を効率良く抽出することができる．

本手法を，横浜国立大学が学内組織や教職員に提供する Web ホスティングサービスより取得できたアクセスログに適用した．その結果，WordPress アプリケーションが稼働している Web サイトに対して，WordPress 管理画面へログインを試みる攻撃事象を抽出できることが確認できた．

本稿の貢献は下記の 2 つである．まず，複数 Web サイトに対するアクセスログを対象として，Web アプリケーションを対象とした Web サイトの攻撃を目的とするリクエストを抽出する手法を提案した．次に，本提案手法を実際の Web ホスティングサービスで取得できたアクセスログに適用した結果，攻撃事象を抽出することができ，手法の有効性を示すことができた．

本稿の構成は次の通りである．第 2 章で関連研究について紹介し，第 3 章でアクセスログ分析手法を提案する．第 4 章で提案手法の評価を行い，第 5 章で提案手法の考察を行う．第 6 章でまとめと今後の課題とする．

2. 関連研究

本章では，悪意あるアクセスの分析・検知技術に関する研究を紹介する．文献 [6] や文献 [7] では，リクエストの持つ文字列の特徴をモデル化し機械学習を適用することで，Web サイトに対する攻撃の検知手法を提案している．文献 [8] では，URI 中の文字列の類似性に基づき，同種のスキャンを検知する手法を提案している．文献 [5] では，ベイズ推定を用いて，ページ遷移の順序性に着目した異常セッションの抽出手法が提案されている．これらの手法により，正規ユーザによる要求が稀な文字列を有するリクエストや複数のリクエストから成るコンテンツアクセスを，攻撃として検知することができる．また，文献 [9] では，Web プロキシログを対象としてリクエスト送信先ホストの種類数や通信先ドメインの評価結果などの特徴を用いてクラスタリングすることで，セキュリティインシデントを検知する手法を提案している．文献 [10] では，ネットワークイベントを対象として，発生したイベントのトリガーとなったイベント同士の関係性を分析することで，顕著に特徴が現れないマルウェアを検知する手法を提案している．

3. 提案手法

3.1 提案手法の流れ

提案手法の全体構成を図 1 に示す．提案手法では，複数の Web サイトに対するアクセスを記録したアクセスログを入力とする．アクセスログの形式を表 1 に示す．入力したアクセスログから，まず送信元ホスト，送信先 Web サイト，リクエスト対象コンテンツ間の関係を分析し，「共起性を持ってリクエストされたコンテンツの組合せ」を抽出する．共起性を持ってリクエストされたコンテンツの組合せは，ある送信元ホストから，異なる Web サイトに対して送信されたリクエストであるものの，同名のコンテンツがリクエストされたペアであるという性質を持つ．

次に，共起性を持ってリクエストされたコンテンツの組合せを，送信元ホストおよび送信先 Web サイトの類似性に

表 1　アクセスログの形式

項目	内容
ホスト (Host)	リクエストの送信元ホスト
Web サイト (Site)	リクエストの送信先 Web サイト
受信時刻	リクエストの受信時刻
コンテンツ (Contents)	リクエスト対象のコンテンツ
ステータス	リクエストに対する応答結果

基づいて分類する．分類されたコンテンツの組合せに対して，コンテンツをノードとするネットワークグラフを作成する．組合せに該当する場合，コンテンツ同士を接続し，共起性を持つ送信元ホスト・送信先 Web サイトを，エッジの属性情報として付与する．作成できたネットワークグラフについて，各ノードが密に接続されているか，エッジがどの送信元ホスト・送信先 Web サイトについて共起性を持つかに着目しグラフの特徴を計算する．密に接続されたノード数が多く，共起性を持つ送信元ホストおよび送信先 Web サイトが多ければ多いほど，多くの送信元ホストが多くの Web サイトに対して，同名のコンテンツを対象とするリクエストを送信していたことがわかる．

提案手法の出力は，コンテンツをノードとするネットワークグラフとグラフの特徴を計算した結果となる．Web ホスティングサービスの管理者は，ネットワークグラフの特徴などから，Web サイトへ攻撃を目的としたリクエストが発生していないかを検証する．

3.2　処理の手順

提案手法の処理手順を述べる．以下では，あるリクエストの送信元ホストを Host，リクエスト送信先 Web サイトを Site，リクエスト対象となったコンテンツを Contents とする．

3.2.1　コンテンツの組合せ抽出

コンテンツの組合せ抽出処理では，Host，Site，Contents 間の関係を分析し，共起性を持ってリクエストされたコンテンツの組合せを抽出する．このとき，アクセスログに存在する Host，Site，Contents の組合せを全て比較する方法では，比較すべき組み合わせ数が爆発するため，処理に時間がかかってしまう．そのため提案手法では，Site，Host のそれぞれの観点から順番に，共起性が存在しないかを検証していく．

まず入力のアクセスログから，Contents 毎に複数の Site に対してリクエスト対象となった Site を集計する（集計 1）．集計 1 の結果から，複数の Site に対してリクエストが送信された Contents を，共起性を持ってリクエストされたコンテンツの候補として，以降の処理対象とする．次に，処理対象とする Contents について，これらの Contents をリクエストした Host によってリクエストされた別の Contents を集計する（集計 2）．そして，集計 2 によってリストアップされた Contents について，共起性を持つ組み合わせとなり得るかを検証する．集計 1，集計 2 の例を図 2 に示す．集計 2 で得られた Contents とそれらをリクエストした Host によってリクエストされた Contents の組合せについて，リクエストが存在した Host と Site の組合せを列挙する．もしも Host と Site の組合せ数が 1 つしかなかった場合，それらの Contents の組合せには共起性が存在しないと判断する．

上述の処理を実行し，集計 2 で示した Contents とそれをリクエストした Host によってリクエストされた Contents の組合せと，それらの Contents 組合せに存在する Host と Site の組合せを示した表を出力する．

3.2.2　ネットワークグラフの分析

ネットワークの分析処理では，3.2.1 で述べた処理によって出力されたコンテンツの組合せ抽出により出力された Contents の組合せについて，各組合せに対応する Host と Site の組合せの類似度に基づいて分類する．

提案手法では，WordPress のような多くの Web サイトで利用されている Web アプリケーションへ攻撃を目的としたリクエストを抽出対象としている．そのため，Web サイトが異なっても同名のコンテンツがリクエストされることを攻撃の特徴としている．しかし，プラグインの導入や無効とした機能の存在等により，同じ意図を持った悪意あるリクエストであっても，対象となる Contents が変化する場合も考えられる．そこで，各 Contents の組合せに対応する Host と Site の組合せが類似しているものを同じグループとして分類する．これにより，Site により Contents が変化する場合でも，同じ意図を持ったリクエストとして判断できる．

ネットワークグラフの作成手順を述べる．分類された Content の組合せのグループについて，Contents をノードとするネットワークグラフを作成する．次に，組合せとして存在する Contents 同士をエッジとして接続する．さらに，エッジに属性情報として，Host と Site の組合せを付与する（図 3）．作成できたネットワークグラフの特徴を分析し，悪意あるリクエストが含まれるネットワークグラフの優先順位づけを行う．

優先順位づけに有効な特徴として，次の 3 点を用いる．

- 互いに密に接続されたノード数（*Clique*）
- ネットワークグラフのエッジに含まれる Site と Host の組合せの平均（*Ave*）
- ネットワークグラフのエッジに含まれる Site と Host の組合せの標準偏差（*Std*）

作成できたネットワークグラフについて上述の特徴を計算する．*Clique* および *Ave* が大きければ大きいほど，より多数の複数 Host により複数 Web サイトに対して，同名の多数のコンテンツがリクエストされたと判断できる．特にこれらに該当するネットワークグラフから，互いに密に接続された箇所に該当する Host，Site，Contents を含むリク

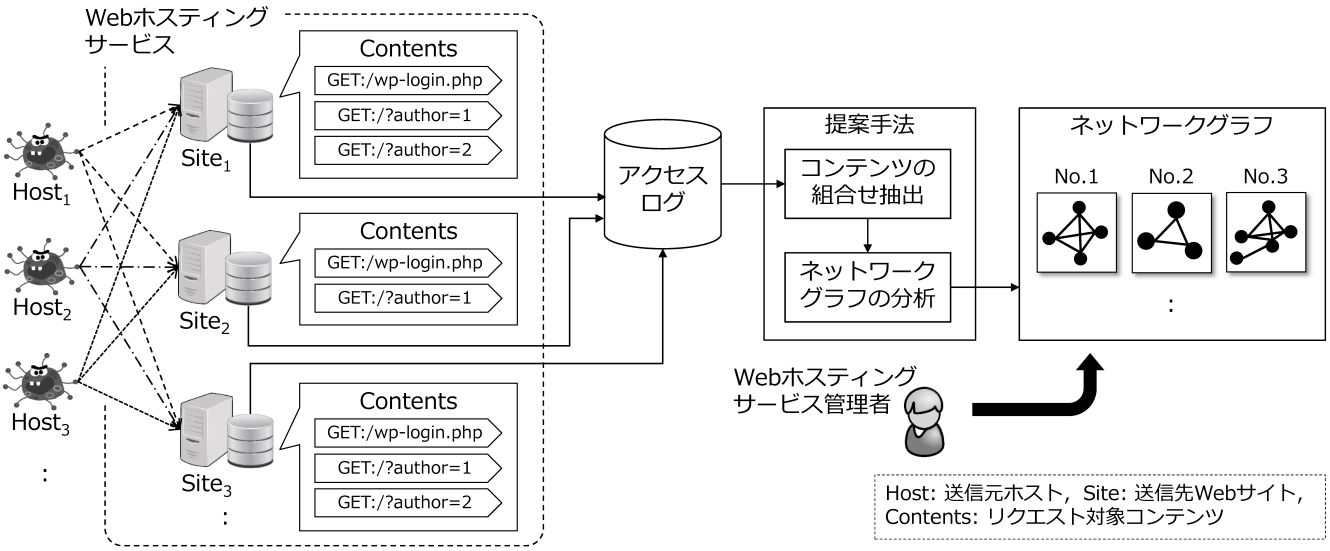
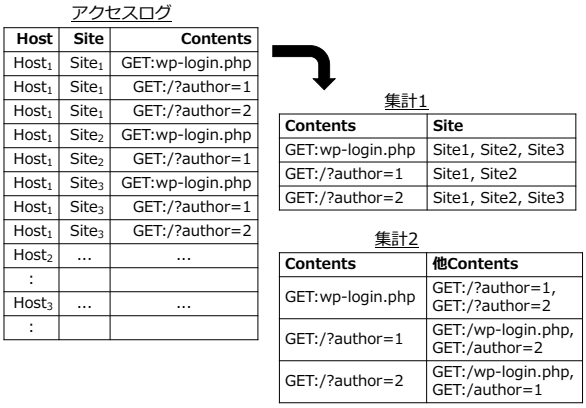


図 1 提案手法の全体構成



Host	Site	Contents
$Host_1$	$Site_1$	GET:wp-login.php
$Host_1$	$Site_1$	GET:/?author=1
$Host_1$	$Site_1$	GET:/?author=2
$Host_1$	$Site_2$	GET:wp-login.php
$Host_1$	$Site_2$	GET:/?author=1
$Host_1$	$Site_3$	GET:wp-login.php
$Host_1$	$Site_3$	GET:/?author=1
$Host_1$	$Site_3$	GET:/?author=2
$Host_2$	...	...
:		
$Host_3$	...	...
:		

Contents	Site
GET:wp-login.php	Site1, Site2, Site3
GET:/?author=1	Site1, Site2
GET:/?author=2	Site1, Site2, Site3

Contents	他Contents
GET:wp-login.php	GET:/?author=1, GET:/?author=2
GET:/?author=1	GET:/wp-login.php, GET:/author=2
GET:/?author=2	GET:/wp-login.php, GET:/author=1

図 2 コンテンツの組合せ抽出における集計処理

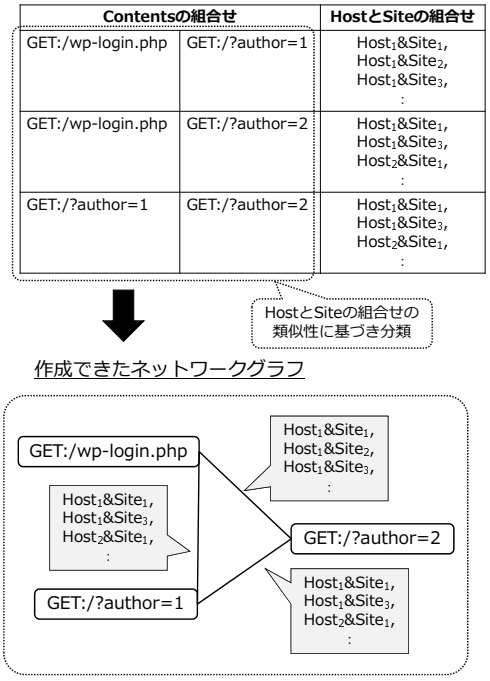


図 3 コンテンツの組合せ抽出処理結果

エストをアクセスログから抽出することで，悪意あるリクエストを特定することができる．

また，*Std* を参照することで，悪意あるリクエストと特定した Contents がどの Host および Site について共起していたのか，その範囲や強弱を判断することができる．

3.3 提案手法の効果

提案手法による効果は次の 3 点である．まず，コンテンツの組合せ抽出処理では，Host および Site について，共起性を持ってリクエストされた Contents を探し出す．これにより，複数 Host から複数 Site に対して送信された複数 Contents から構成される悪意あるリクエストを，互いに関係性を持つ Contents として判断することができる．

次にネットワークグラフの分析処理時に，共起性を持つ Host と Site の組合せの類似性に基づき，共起性を持つ Contents の組合せを分類する．これにより，リクエスト対象となる Contents が変化する場合でも，同じ意図を持ったリクエストを同じグループとして分類することができる．

4. 評価実験

提案手法の有効性を検証するため，我々は提案手法を実装し，横浜国立大学 Web ホスティングサービスより取得できたアクセスログに適用した．

横浜国立大学 Web ホスティングサービスは横浜国立大

表 2　適用対象アクセスログの上位 10 種類のコンテンツ

コンテンツ	レコード件数
GET:	181,022
POST:wp-login.php	163,811
GET:style.php	100,537
GET:box_img_matrix	73,088
GET:superfish_settings.js.php?ver=1.0	52,704
GET:fontawesome-webfont.eot?	42,098
POST:admin-ajax.php	19,248
GET:fontawesome-webfont.woff?v=4.3.0	18,707
HEAD:	10,610
POST:xmlrpc.php	5,318

学が学内組織や教職員に提供する Web ホスティングサービスであり，学内の部局や研究室等の Web サイトを運用を行っている．アクセスログには，これらの Web サイトに対する学内外からのアクセスが記録されている．

4.1　実験手順

本実験では，横浜国立大学 Web ホスティングサービスにて WordPress が稼働している 18 の Web サイト（Site）に対するアクセスログを適用対象とした．本実験では，リクエストメソッドが異なれば，同じコンテンツをリクエストしていても，異なるコンテンツ (Contents) とした．例えば，GET メソッドでリクエストされた index.php と POST メソッドでリクエストされた index.php は，異なるコンテンツであると判断する．適用対象としたアクセスログは 2015 年 7 月 1 日から 7 月 30 日の 1 ヶ月間に発生したリクエストで，レコード件数は 4,204,965 件，送信元 IP アドレス (Host) の種類数は 34,648，コンテンツの種類数は 19,079 であった．画像，JavaScript，CSS(カスケードスタイルシート) を除いた，コンテンツ上位 10 を表 2 に示す．

さらに，作成できたネットワークグラフの優先順位づけを行うため，*Clieque*，*Ave*，*Std* を計算した．これらの 3 次元ベクトルについて正規化を行った上でノルムを計算し，優先スコアとした．

4.2　実験結果

このレコードに対し提案手法を適用した結果，555 のネットワークグラフが作成できた．これらのネットワークグラフについて優先スコアを計算した．優先スコアの分布を図 4 に示す．ここから，優先スコアは 0.5 から 2 の区間に偏っていることがわかる．

優先スコアの上位 6 に該当するネットワークグラフを図 5 に，各ネットワークグラフに含まれるコンテンツ，IP アドレス，Web サイトの種類数を表 3 に，それぞれ示す．図 5 のネットワークグラフにおけるエッジの太さは，コンテンツ間に関係がある IP アドレスと Web サイトの組合せの数に比例する．

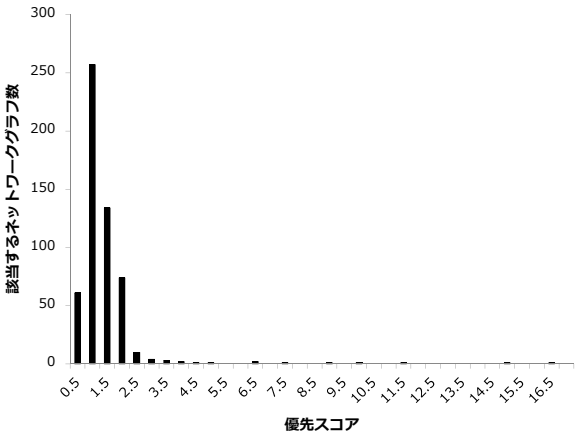


図 4　優先スコアの分布

表 3　上位 6 のネットワークグラフに属するリクエストの種類数および優先スコア

	(1)	(2)	(3)	(4)	(5)	(6)
コンテンツ	21	16	87	3	3	203
IP アドレス	48	44	513	10	8	340
Web サイト	11	8	18	8	8	18
優先スコア	16.27	14.64	11.23	9.63	8.87	7.11

4.3　優先スコア上位ネットワークグラフの分析

図 5 に示したネットワークグラフに含まれるリクエストを分析し，悪意あるリクエストの有無を検証する．

(1)(2) は，WordPress 管理画面にログインを試みるリクエストが集まったものと考えられる．(1)(2) に含まれる IP アドレスは約 73%が，Web サイトは約 91%が共通だった．ネットワークグラフ中心に存在する「?author=1」「?author=2」等のコンテンツは，WordPress に登録されているユーザ名を取得することが可能なリクエストである[*1]．「GET: wp-login.php」は WordPress ログイン画面を表示するために送信されるリクエスト対象のコンテンツである．ログイン画面上でユーザ名とパスワードを入力しログインを行う場合，「POST: wp-login.php」に対してリクエストが送信される．(1)(2) のグラフでは，このような WordPress 管理画面へのログインに関するコンテンツが，互いに共起性を持つものとして接続されている．

(3)(6) は，ID の割り振られたコンテンツに対するリクエストが集まったものと考えられる．(1)(2) とは異なり，(3)(6) のグラフには複数のコンテンツが互いに接続されている形状が確認できない．さらに，(3) に属する Host の約 59%が学内 IP アドレスで，(6) に属する Host の約 47%が検索エンジンのクローラであった．よって，(3)(6) に属するリクエストは悪意のないリクエストであると考えられる．

(4)(5) は，3 種類のコンテンツ「GET:」「GET:Diagnostics.asp」「GET:Ringing.at.your.dorbell!」

[*1] アプリケーションの変更やプラグインの追加により，ユーザ名が取得できないように設定することも可能である．

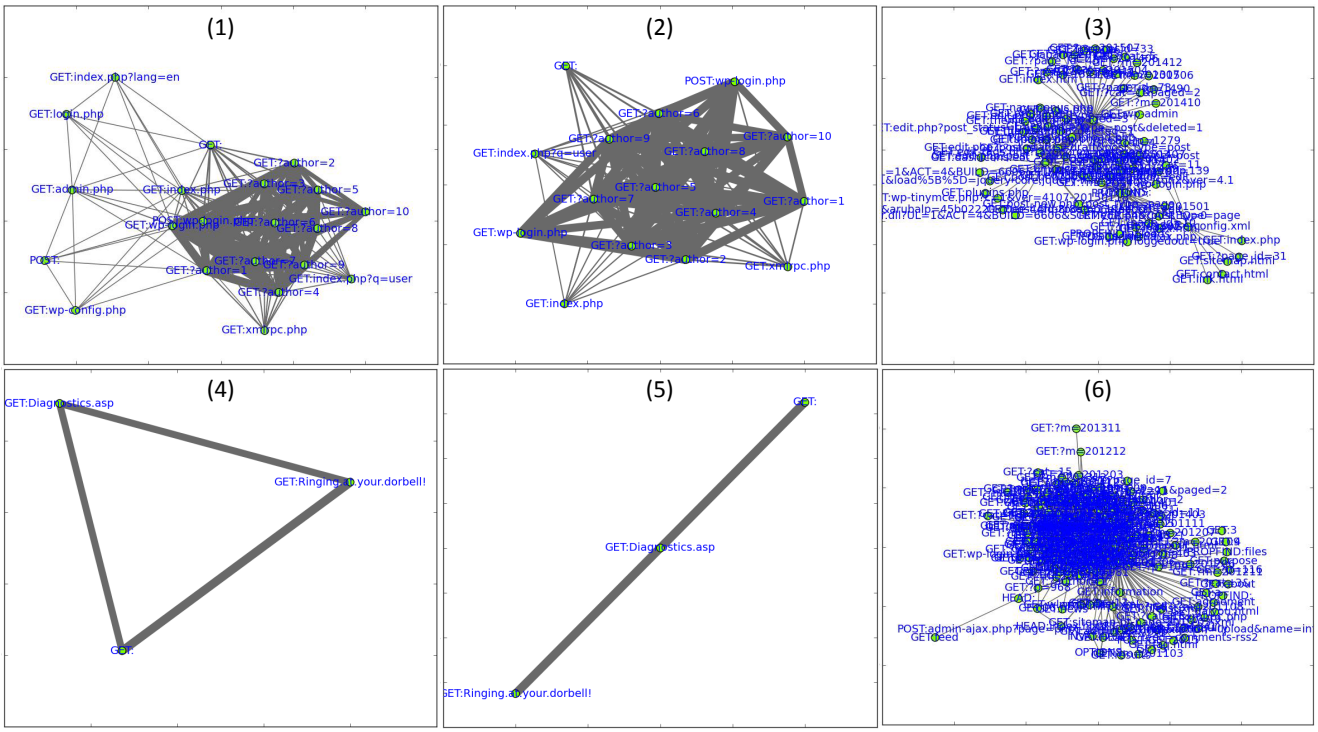


図 5　スコア上位 6 のネットワークグラフ

から構成されるネットワークグラフである．この 3 種類のコンテンツに対するリクエストは他でも観測されており，リクエストヘッダに悪意ある文字列が挿入されていたとの報告もある (文献 [11])．よって，(4)(5) に属するリクエストは既知の悪意あるリクエストであると考えられる．

4.4　WordPress 管理画面へのログインアクセスの検証

図 5 の (1)(2) のネットワークグラフに属するリクエストについて，WordPress に対するログインを試みるリクエストをさらに分析する．

まず，(1)(2) のネットワークグラフに属する 48 の IP アドレスの AS を集計した結果，最も多く含まれていたのは Chinanet であった (38 IP アドレス)．次に，これら Chinanet を AS とする IP アドレスを含むレコードを抽出した．抽出したレコードは 90,830 件あり，6 つの Web サイトに対するリクエストが記録されていた．

これらのレコードについて，送信元 IP アドレス毎のリクエスト推移を図 6 に示す．この図から，送信元 IP アドレスは概ね同じ順番で 6 つの Web サイトに対しリクエストを送信していたことがわかる．Web サイト毎の各コンテンツに対するアクセス推移を図 7 に示す．いずれの Web サイトに対しても，同名のコンテンツがリクエストされていたことがわかる．

ここで，1IP アドレスから 1 Web サイトに対するリクエストを記録したレコードの一部を表 4 に示す．この IP アドレスはある Web サイトに対し，「GET:wp-login.php」を 2 回リクエストしたのち，1 から 10 までインクリメントしながらリクエスト「?author=」を送信していた．そののち，11:34:51 から 11:35:11 の間に「POST:wp-login.php」を 60 回送信していた．これらの挙動から，一連のアクセスは，リクエスト送信対象とした Web サイトに対して WordPress 管理画面へログインを試みるアクセスであると推測できる．「POST:wp-login.php」の送信回数は IP アドレスおよび Web サイトにより異なっていたものの，この傾向は本節にて抽出したレコード全てが該当した．

この結果から，特定の IP アドレス群が特定の Web サイトを対象として，ログインを試みるリクエストが断続的に発生していたことがわかった[*2]．

5.　議論

本章では，提案手法の適用により，悪意あるリクエストのさらに効率的な抽出に向けた議論を行う．

第 4 章で報告した評価実験により，図 5 の (1)(2) のネットワークグラフに属する IP アドレス，Web サイト，コンテンツを手掛かりとして，WordPress 管理画面に対しログインを試みるアクセスが Web サイトをまたいで発生していたことを突き止めた．しかし，図 5 で示すネットワークグラフには，「GET:index.php?lang=en」などといった WordPress 管理画面へのログインに直接関連のないコンテ

[*2] 現在，横浜国立大学 Web ホスティングサービスでは，WordPress 管理画面へのアクセスに対してフィルタリングを行うなどの対策を取っている．

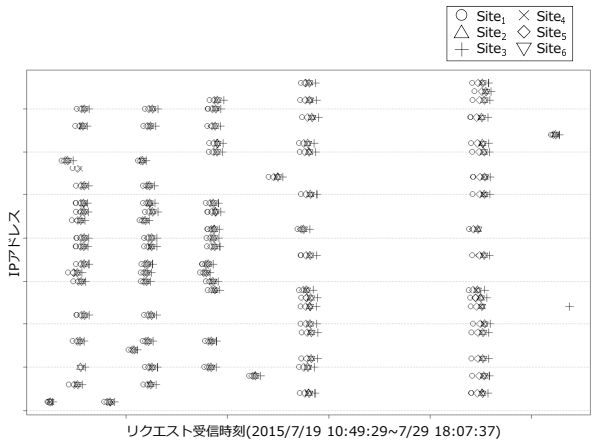


図 6 各送信元 IP アドレスのアクセス推移

表 4 WordPress 管理画面に対するログインアクセスの例

受信時刻	コンテンツ	ステータス
7/19 11:34:48	GET:wp-login.php	200
7/19 11:34:48	GET:wp-login.php	200
7/19 11:34:48	GET:?author=1	200
7/19 11:34:48	GET:?author=2	404
:	:	:
7/19 11:34:51	GET:?author=9	200
7/19 11:34:51	GET:?author=10	200
7/19 11:34:51	POST:wp-login.php	200
7/19 11:34:51	POST:wp-login.php	200
7/19 11:34:52	POST:wp-login.php	200
:	:	:
7/19 11:35:11	POST:wp-login.php	200

ンツも含まれている．また，4.4 節で分析対象としなかった IP アドレスも，WordPress 管理画面に対するアクセスを行っていた．Web サイトに対するアクセスログから悪意あるリクエストを抽出するには，今回人手で行った検証作業も可能な限り自動化する必要がある．抽出結果を攻撃元 IP アドレスのブラックリストや IDS（Intrusion Detection System，侵入検知装置）に適用可能なシグネチャ形式で出力することができれば，分析から対策のプロセスを自律的に行うことが可能となる．

第 4 章では，ネットワークグラフの精査により悪意あるリクエストを効率的に抽出できるように，*Clique*，*Ave*，*Std* の 3 種類の特徴に着目した優先スコアを計算した．しかし，図 5 には悪意のないリクエストを含むネットワークグラフも含まれており，このネットワークグラフの精査は不要であった．そこで，悪意あるアクセスが含まれる可能性が高い，精査する価値があるネットワークグラフが上位に挙がってくるような優先スコアの計算アルゴリズムを構築する必要がある．

6. まとめと今後の課題

本稿では，Web アプリケーションを対象とした Web サイトに対する悪意あるリクエストを，アクセスログから抽出する手法を提案した．提案手法を実際のアクセスログに対して適用した結果，既知の攻撃に加えて，WordPress が稼働している Web アプリケーションの管理画面へログインを試みる攻撃が，複数サイトに対して発生した事象を抽出できた．

今後の課題として，出力されたネットワークグラフから攻撃に該当するリクエストを特定し，IDS にシグネチャとして登録する処理を追加する．これにより，分析結果を次の検知に活用するサイクルを確立する．また，評価実験では WordPress が稼働している Web サイトに関するアクセスログを対象としたものであった．提案手法の有効性を示すため，対象範囲を広げ全ての Web サイトに対するアクセスログを対象に提案手法を適用することも必要である．

謝辞 本研究の一部は文部科学省国立大学改革強化推進事業の支援を受けて行われました．

本研究にあたり，アクセスログを提供頂きました横浜国立大学情報基盤センターに深く感謝致します．

参考文献

[1] https://wordpress.org/, ”Blog Tool, Publishing Platform, and CMS - WordPress.”

[2] https://www.joomla.org/, ”Joomla! The CMS Trusted By Millions for their Websites.”

[3] https://lolipop.jp/info/news/4149/, ”第三者によるユーザーサイトの改ざん被害に関するご報告 - 2013 年 08 月 29 日 10 時 57 分 / 新着情報 / お知らせ - レンタルサーバーならロリポップ！.” last visited 2016/8/1.

[4] http://www.jpcert.or.jp/magazine/acreport-cms.html, ”改ざんの標的となる CMS 内の PHP ファイル (2016-02-25).” last visited 2016/8/1.

[5] Cho, Sanghyun, and Sungdeok Cha. ”SAD: web session anomaly detection based on parameter estimation.” Computers & Security 23.4 (2004): 312-319.

[6] Kruegel Christopher, Giovanni Vigna, and William Robertson. ”A multi-model approach to the detection of web-based attacks.” Computer Networks 48.5 (2005): 717-738.

[7] Zhong Yang, Hiroshi Asakura, Hiroki Takakura, and Yoshihito Oshima. ”Detecting Malicious Inputs of Web Application Parameters Using Character Class Sequences.” Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual. Vol. 2. IEEE, 2015.

[8] 鐘場, 折原慎吾, 谷川真樹, 嶋田創, 村瀬勉, 高倉弘喜, 大嶋嘉人, ”URI の共起性に基づく Web スキャンの実態調査.” 信学技報 IEICE Technical Report, ICSS2015-51 (2016-03), 2016.

[9] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. ”Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks.” Proceedings of the 29th Annual Computer Security Applications Conference, 2013.

[10] Zhang Hao, Danfeng (Daphne) Yao, and Naren Ramakrishnan. ”Detection of stealthy malware activities with traffic causality and scalable triggering relation discovery.” Proceedings of the 9th ACM symposium on Infor-

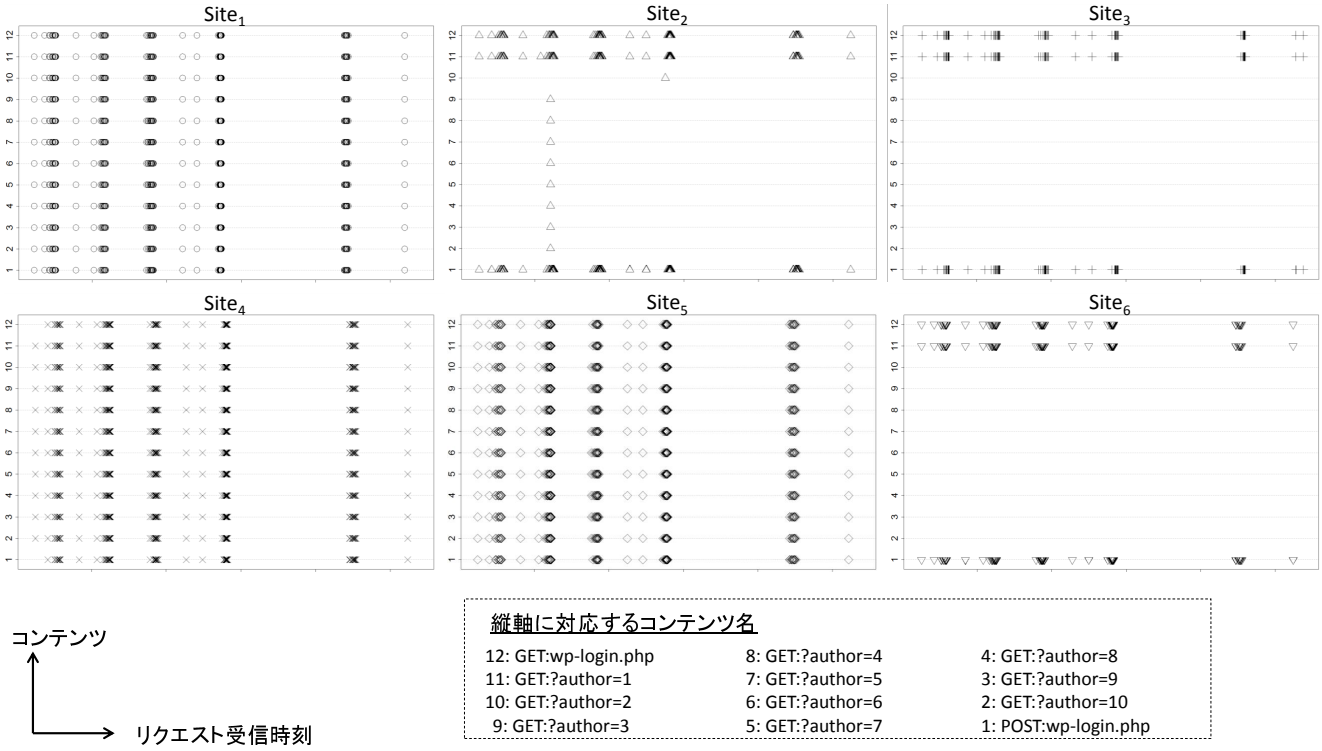


図 7 Web サイト毎の各コンテンツに対するアクセス推移

mation, computer and communications security, 2014.
[11] http://www.skepticism.us/2015/05/new-in-your-face-malware-attacks-me-ringing-at-your-dorbell/, ”New, in your face, malware attacks me: /Ringing.at.your.dorbell!-Dog Is My Copilot.” last visited 2016/8/9.