情報処理学会論文誌 Vol.59 No.2 574–590 (Feb. 2018)

# 多数のWebサイトを対象とした攻撃の共起性に基づく悪性アクセス検知手法とその評価

齊藤 聡美[1,2,a)] 吉岡 克成[3] 松本 勉[3]

受付日 2017年5月10日, 採録日 2017年11月7日

**概要**：インターネットに公開されているWebサイトには，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等が日々到達している．我々は，複数Webサイトのアクセスログから，複数Webサイトに送信された悪意あるリクエストを抽出する手法を提案する．本稿では，Webアプリケーションの脆弱性の探索を目的としたリクエストや悪意あるコード挿入を行うリクエストで，送信されるURIにパターンが存在するリクエストを悪意あるリクエストとして抽出する．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，アクセスログの分析を行う際に適用することを想定し，これらのログにおけるアクセスの送信元IPアドレス，送信先ドメイン，URIの関係性を分析し，しきい値数以上のWebサイトに対して同一のURIを送信した送信元IPアドレスを攻撃元として抽出する．提案手法を，実際のWebホスティングサービスのアクセスログに適用した結果，攻撃元となったIPアドレスを誤検知なく抽出できるしきい値が存在することを示した．さらに，既存のオープンソースのIDS（Intrusion Detection System）およびWAF（Web Application Firewall）ではシグネチャが登録されておらず，検知できない攻撃についても提案手法では検知できる事例を確認した．提案手法は単体では見逃し率が高いため，既存の攻撃検知技術と併用することで効果が期待できる．

**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,a)] Katsunari Yoshioka[3] Tsutomu Matsumoto[3]

Received: May 10, 2017, Accepted: November 7, 2017

***Abstract:*** Websites on the internet accept requests with many kinds of purposes today. For example, normal users for getting contents, search engines for collecting websites properties, attackers for intruding web servers and etc. In this paper, we propose a method for extracting malicious requests from access log collected from multiple websites. We aim to the malicious requests with sending specific URI patterns. Those requests aim to searching vulnerable web applications and inserting malicious codes in those headers. We assume that our method is applied on a website hosting service provider who monitors his websites. Our method analyzes relations among source IP address, destination domain and URI from access log and extracts source IP Addresses who have co-occurrence between destination domains and sent URIs. Those IP addresses sent requests for multiple domains and the URIs are shared among such domains. We apply our method for real access log collected from website hosting service on our university. As a result, our method succeeded extracting malicious source IP addresses without false positives under the specific thresholds. Furthermore, we show malicious requests that cannot be detected by other detecting tools such as IDs (Intrusion Detection System) and WAF (Web Application Firewall). These tools have no signatures that can detect those requests. Our proposal method has high false negative ratio, so we expect that our method performs effectively combined with other detecting systems.

***Keywords:*** Web access log, log analysis and network monitoring

## 1. はじめに

インターネットに公開されているWebサイトには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等が日々到達している．

また，今日では多くの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．たとえば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．特にWordPress [1] やJoomla! [2] は，オープンソースとして公開されているCMSで，個人・組織を問わず多く利用されている．さらにこれらの機能を拡張するためのプラグインも多く存在する．

しかし，こうしたCMSが攻撃の対象となる事例も報告されている．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 [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 [4]．CMSはWebサイトごとに異なる形態・規模で運用されているものの，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ため，その脆弱性は多くのWebサイトに影響を与えうる．

Webサイトに対する悪意あるアクセスの検知手段として，リクエストの文字列が特定のパターンに合致すれば攻撃と判断するパターンマッチングや，普段とかけ離れたデータを含むリクエストを攻撃と判断するアノマリ検知が存在する．これらの技術は，通常，各々のWebサイトに対するリクエストあるいはリクエスト群を対象に攻撃を検知する．しかし，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区別が難しいものや，異常性の確認が難しいものも存在する．たとえば，ゼロデイや発見されたばかりの脆弱性を突く攻撃であれば，検知パターンが対応していなければシグネチャマッチングによる検知はできない．アノマリ検知についても，不特定多数からのアクセスを受付けるWebサイトについては，「正常なリクエスト」の定義が難しい場合がある．

そこで我々は，脆弱性を持つWebアプリケーションの探索や，Webサイトに対して悪意あるコード挿入を行うリクエストを，悪意あるリクエストとして検知することを考える．我々は，複数のWebサイトを監視対象とし，単一の送信元から複数のWebサイトに向けて送信されたリクエストに着目することで，リクエストの異常性を判断するアプローチをとる．脆弱性を持つWebアプリケーションの探索や，悪意あるコード挿入を行うリクエストには，単一の送信元から送信される際にそのURIにパターンが存在するものがある．たとえば，phpMyAdminに対する脆弱性を探索するとみられるリクエストには，下記6種類のURIを複数のWebサイトへ送信するIPアドレスが複数存在する事象が，横浜国立大学で運用しているWebホスティングサービスで観測されている[*1]．

- GET //myadmin/scripts/setup.php HTTP/1.1
- GET /muieblackcat HTTP/1.1
- GET //pma/scripts/setup.php HTTP/1.1
- GET //MyAdmin/scripts/setup.php HTTP/1.1
- GET //phpMyAdmin/scripts/setup.php HTTP/1.1
- GET //phpmyadmin/scripts/setup.php HTTP/1.1

また，ShellShockの脆弱性を突く攻撃をリクエストヘッダに含むリクエストには，下記3種類のURIパターンで到達する攻撃が存在することが報告されている [6]．こちらの事象も横浜国立大学で運用しているWebホスティングサービスで観測されている．

- GET / HTTP/1.0
- GET /Ringing.at.your.dorbell! HTTP/1.0
- GET /Diagnostics.asp HTTP/1.0

本稿では，Webアプリケーションの脆弱性の探索を目的としたリクエストや悪意あるコード挿入を行うリクエストで，送信されるURI群にパターンが存在するリクエストを悪意あるリクエストとして抽出する．正規ユーザならば，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，目的・規模の異なる複数Webサイトに対してまったく同じコンテンツをリクエストすることは稀であると考えられる．よって，単一の送信元から複数Webサイトに対して同一のURIが送信されたならば，URI文字列に特徴を持つものであれば，当該リクエストは正規目的でなく，攻撃目的の可能性が高いと判断できる．

本稿では，複数Webサイトに対するアクセスログを対象として，複数Webサイトに送信された悪意あるリクエストを検知する手法を提案する．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，自身の管理下にあるWebサイトの監視を行うため，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．さらに，本稿で提案する手法は，User-AgentやReferrerといった送信元の身元やリクエストの起源を示す情報，リクエスト中のヘッダ情報は用いない．User-AgentやReferrerの文字列

[1] 横浜国立大学
Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[2] 株式会社富士通研究所
FUJITSU LABORATORIES LTD., Kawasaki, Kanagawa 211–8588, Japan
[3] 横浜国立大学環境情報研究院/横浜国立大学先端科学高等研究院
Graduate School of Environment and Information Sciences and Institute of Advanced Sciences, Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[a)] saito-satomi-zc@ynu.jp

[*1] URI中の文字列「muieblackcat」があるが，これはphpMyAdminの脆弱性スキャナであるMuieblackcatに関係するものと考えられる [5]．

を正規ユーザやクローラを模擬した文字列に設定された場合であっても，提案手法では悪意あるリクエストが抽出できるようにする．

本手法では，一定期間収集したアクセスログにおいて，送信元 IP アドレス，送信先ドメイン，リクエストの対象となった URI の関係性を分析し，複数のドメインに対して同一の URI を送信した送信元 IP アドレスから，しきい値数以上のドメインに対して同一の URI を送信した送信元 IP アドレスを攻撃元として抽出する．本手法により，単一のドメインへのアクセスを分析することでは異常性の判断が難しいリクエストであっても，しきい値を超える種類のドメインに対して，しきい値を超える種類の共通した URI を送信した IP アドレスならば，提案手法では悪性として抽出することが可能である．また Web ホスティングサービス管理者は，抽出された IP アドレスに対して，当該 IP アドレスの送信先となったドメインが割り当てられた Web サイトが被害を受けていないかを検証したり，IP アドレスをブラックリスト化したりすることができる．

提案手法の評価のため，横浜国立大学が運用する Web ホスティングサービスのアクセスログを分析し，検知結果をオープンソースの IDS（Intrusion Detection System）である snort [5] および WAF（Web Application Firewall）の modsec [8] と比較した．その結果，送信先となったドメイン種類数が 91 以上かつ URI 種類数が 4 以上，および送信先となったドメイン種類数が 5 以上かつ URI 種類数が 45 以上にしきい値を設定した場合において，攻撃元となった IP アドレスを誤検知なく抽出できることを示した．さらに，比較対象のツールではシグネチャの登録が間に合わず検知できない攻撃についても，提案手法により検知できる事例を確認した．提案手法は単体では見逃し率が高いため，既存の攻撃検知技術と併用することで効果が期待できる．本稿の貢献は下記の 2 点である．

1. 複数ドメインへの共通の URI 送信という挙動に着目することで，シグネチャ等を必要とせず検知が可能な手法を提案した．
2. 255 サイト規模からなる実際のホスティングサービスのアクセスログを用いて評価を行い，既存の検知ツールでは検知できない攻撃を検知できる例を示した．

本稿の構成は下記のとおりである．2 章で関連研究を紹介し，3 章で提案手法を述べる．4 章で提案手法の評価実験を行った結果を示し，5 章で評価実験の結果に関する議論を行う．6 章でまとめと今後の課題を述べる．

## 2. 関連研究

本章では，Web サイトへの攻撃の分析・検知技術に関する関連研究を紹介する．文献 [9] で Kruegel らは URI の文字列をモデル化し学習することで，Web サイトに対する異常なリクエストを抽出する手法を提案している．文献 [10] では，鐘らは Web アプリケーションの脆弱性や設定ミスに起因する脆弱性を持つ Web サイトの発見を目的としたリクエスト（Web スキャン）を，アクセスログから検知する手法を提案している．この手法では，1 つの送信元 IP アドレスから多数の送信先 IP アドレスへアクセスが発生すること，Web スキャンに該当するリクエストはその URI が文字列として類似することを Web スキャンの特徴としている．しかし，URI が文字列として類似しないリクエストの悪性の有無を判断することが難しい．そのため，文字列としては類似しないものの，送信される URI にパターンが存在するリクエストを抽出できない場合がある．また，文献 [11] では，Sanghyn ら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な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．文献 [12] では，Dusan らはアクセスログから抽出した特徴を自己組織化マップに適用し，悪意ある Web クローラの識別を行う手法を提案している．彼らは，1 セッションあたりのリクエスト発生頻度やリクエスト種類，レスポンスのエラー率等を特徴としている．

Web サイトに対する攻撃を防ぐ手段として，Web サイトに対する攻撃を模擬したリクエストを Web サイトに適用したりすることで，その Web サイトが持つ脆弱性を洗い出す手段が存在する．洗い出した脆弱性を公開前に対処することで，Web サイトに対して実際に攻撃が発生するのを防ぐことができる．文献 [13], [14] では，SQL インジェクションや XSS（クロスサイトスクリプティング）といった Web サイトに存在する脆弱性を検査する手法が提案されている．文献 [15] では，Doupe らは Web サイトの内部状態の遷移を考慮した脆弱性検査手法を提案している．また，こうした脆弱性検査技術の評価を行った研究もある．文献 [16] では，Doupe らは著者らが構築した脆弱性を持つ Web サイトに対して 11 種類のブラックボックス型の脆弱性検査ツールを適用し，脆弱性の検知率を計測している．文献 [17] では，Bau らは 8 種類の脆弱性検査製品に対して既知の脆弱性を突く攻撃の検知率を比較した結果を報告している．しかし，パターンマッチングと同様に，ゼロデイや発見されたばかりの脆弱性を突く攻撃であれば，こうした検査が間に合わない場合が存在する．

サーバ型ハニーポットをインターネット上に設置することで，Web サイトに対する攻撃に関する情報を収集できる．ハニーポットに対するアクセスを精査することで，攻撃リクエストの傾向や，ゼロデイといったこれまで認識されていなかった攻撃を知ることができる．文献 [18] では，John らは脆弱性を持つ Web サイトの情報をインターネットから取得し再現するハニーポットシステムを提案している．文献 [19] では，Yagi らは送信元とインタラクション

を行うことで攻撃側の挙動を観測できるWebハニーポットを提案している．文献[20]で久世らはハニーポットに対する通信ログからHTTPリクエストおよびレスポンスや通信の特徴を抽出し，悪意あるアクセスを識別する手法を提案している．特に，ハニーポットを複数台設置することで，数値的に連続したIPアドレスに対して順に通信要求を行う特徴を取得可能であると述べている．Canaliらは文献[21]で，Webアプリケーションの脆弱性を模擬したWebサイトを多数設置してリクエストを観測している．Canaliらは観測の結果，4段階の攻撃フェーズと13種類の目的に分類できたことを報告している．他にも，インターネット上で入手可能なサーバ型ハニーポットとしてGlastopf[22]やDionea[23]等が存在する．ハニーポットで観測できた攻撃情報をIDS等のシグネチャにフィードバックする技術も提案されている．文献[24]では，Christianらはハニーポットに到達した通信履歴からIDS用のシグネチャを自動生成する手法を提案している．ハニーポットは実際のサービスが運用されていないため，ハニーポットに到達するリクエストをほぼ悪性と見なして，悪意あるリクエストの収集や分析が実施できる．しかし，囮のWebサイトであることが攻撃側に分かってしまうと，攻撃対象から除外され，悪意あるリクエストが観測できなくなる恐れがある．

## 3. 提案手法

本章では，複数Webサイトのアクセスログを対象として，複数Webサイトに送信された悪意あるリクエストを抽出する手法を提案する．

鐘らの手法では，1つの送信元IPアドレスから多数の送信先IPアドレスへアクセスが発生すること，Webスキャンに該当するアクセスはそのURIが文字列として類似することを特徴として，Webアプリケーションの脆弱性や設定ミスを持つWebサイト発見を目的としたリクエスト（Webスキャン）の特徴としている[10]．

しかしこの手法では，URIが文字列として類似しないアクセスの悪性の有無を判断することが難しい．たとえば，横浜国立大学で運用しているWebホスティングサービスの管理下にある91のWebサイトに対して同一のIPアドレスから，ShellShockの脆弱性を突く攻撃をリクエストヘッダ内に含むリクエストが到達していたことが確認されている[6]．このリクエストは下記の3種類のURIで構成されている．

- GET / HTTP/1.0
- GET /Ringing.at.your.dorbell! HTTP/1.0
- GET /Diagnostics.asp HTTP/1.0

これらのURIは，同一の送信元IPアドレスから送信されたこと，複数のWebサイトに対して送信された特徴を持つ．一方で，URIの文字列として見たときに，類似性が高いとはいえない．

これらの悪性RUIは，文字列としては高い類似性を有していないものの，リクエストを受信したWebサイトは，どれも同じURIを受信していたという特徴があった．そこで我々は，1つの送信元IPアドレスから多数のWebサイトへアクセスが発生する特徴を鐘らの手法より継承する．そのうえで，URIの文字列の類似性を比較する代わりに，ある送信元IPアドレスからある送信先Webサイトに対して送信されたURIをグループ化し，そのグループが複数のWebサイトに対して共通して送信されたことを悪性リクエストか否かを判断する特徴とする．なお，提案手法ではURI間の文字列の類似性を比較しないため，送信元IPアドレスが攻撃対象としたWebアプリケーションの種別を分類するには，鐘らの手法等でURIを分類する必要がある．

提案手法では，User-AgentやRefererといった送信元IPアドレスの身元やリクエストの起源を示す情報は入力として用いない．User-AgentやRefererは送信側で任意の文字列を設定できる．そのため提案手法では，User-AgentやRefererの文字列を正規ユーザやクローラを模擬した文字列に設定された場合であっても，提案手法では悪意あるリクエストが抽出できるようにする．

提案手法の全体構成を**図1**に示す．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の管理者が，自身の管理するWebサイト群のセキュリティ監視を行うことを目的として使用することを想定する．このWebホスティングサービスでは，管理下のWebサイト群にはドメインが個別に割り当てられている．提案手法では，複数のWebサイトから取得したアクセスログを入力とし，複数のWebサイトに対して同一のURIを送信した送信元IPアドレスを攻撃と判断する．

提案手法の入力とするアクセスログについて述べる．アクセスログには，ある送信元IPアドレスからWebホスティングサービス管理下のドメインに対して送信されたリクエストが1行のレコードとして記録される．

アクセスログ中のレコードは，送信元IPアドレス，送信先となったドメイン（ドメイン），アクセス受信時刻（受信時刻），ドメインに対して送信されたURI（URI）の4項目から構成される（**表1**）[*2]．

提案手法では，監視対象のWebサイト群に対して発生したアクセスを記録し，一定期間監視した結果得られるアクセスログを入力とする．提案手法の処理手順を述べる．まず，複数送信先ドメインに共通するURIを送信した送信元IPアドレスを抽出する手順を述べる．この抽出手順では，まず複数種類の送信先ドメインにURIを送信した送信元IPアドレスを抽出し，当該送信元IPアドレスが送信した

[*2] 送信元IPアドレスは，NAT配下である場合があり，送信元IPアドレスが同一であっても，実際紐づいているマシンは異なる場合がある．

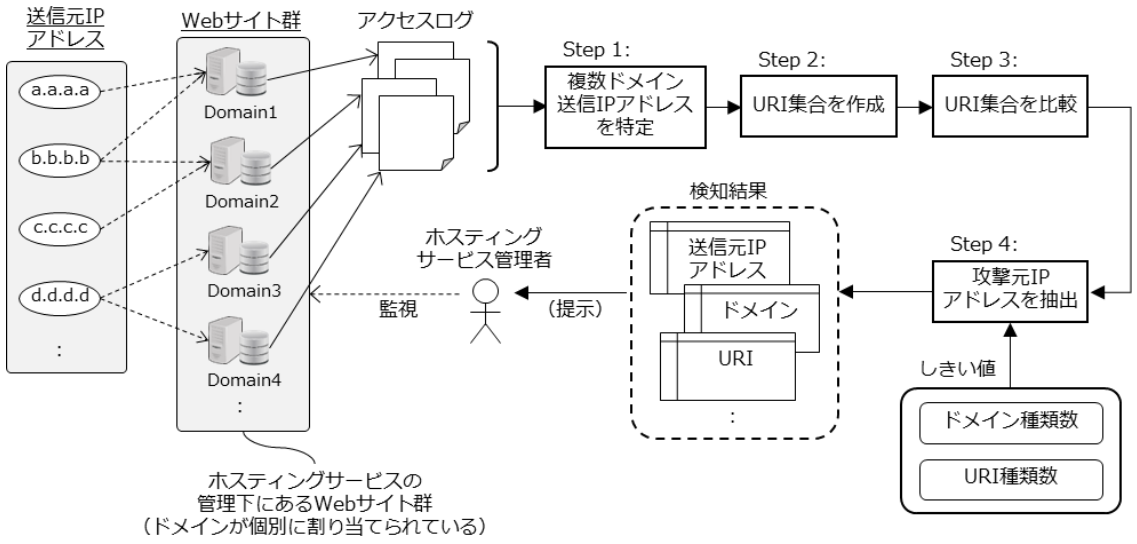



図 1 提案手法の全体構成
**Fig. 1** The whole structure of proposal system.

表 1 入力とするアクセスログ
**Table 1** Input access log.

| 送信元 IP アドレス | ドメイン | 受信時刻 | URI |
|---|---|---|---|
| a.a.a.a | ynu | 4/1 0:00 | GET /index.html |
| a.a.a.a | ynu | 4/1 0:01 | GET /logo.png |
| b.b.b.b | ias | 4/1 0:30 | GET /robots.txt |
| : | : | : | : |

URI が送信先ドメイン間で一致するか否かを検証する．入力のアクセスログから送信元 IP アドレスとドメインの項目を取り出し，送信元 IP アドレスごとにドメイン種類数（送信元 IP アドレスに紐づいたドメイン種類数）を集計する．集計の結果，2 種類以上のドメインに URI を送信した送信元 IP アドレスを抽出し，複数送信先ドメインに URI を送信した送信元 IP アドレスと判断する．次に，抽出した送信元 IP アドレスを含むアクセスログを対象に，各送信元 IP アドレスについて，送信先ドメインに同じ URI を送信したか否かを検証する．各送信元 IP アドレスについて，送信先ドメインごとに URI の集合を作成する．作成した URI 集合を比較し，完全一致した場合に，当該送信元 IP アドレスは 2 種類以上の複数種類の送信先ドメインに対して同一の URI を送信したと判断する．なお，各ドメインに対するリクエストの受信時刻が同時刻の場合，リクエストの順序性を判断することが難しい．そのため，本手法では送信先ドメインが受信した URI の順番は考慮しない．

上述の抽出処理により，複数送信先ドメインに対して同一の URI を送信した送信元 IP アドレスを抽出できる．これらから，悪性の可能性が高い送信元 IP アドレスを絞り込む．提案手法では，送信元 IP アドレスの送信先ドメイン種類数と，送信元 IP アドレスが送信した URI 集合の種類数の，2 つの観点を悪性送信元 IP アドレスの判断基準とする．このドメイン種類数および URI 集合の種類数について，それぞれしきい値を設ける．設けたしきい値を超えるような多種類のドメインに対して，多種類の共通した URI を送信した送信元 IP アドレスは，悪性の可能性が高いと判断する．

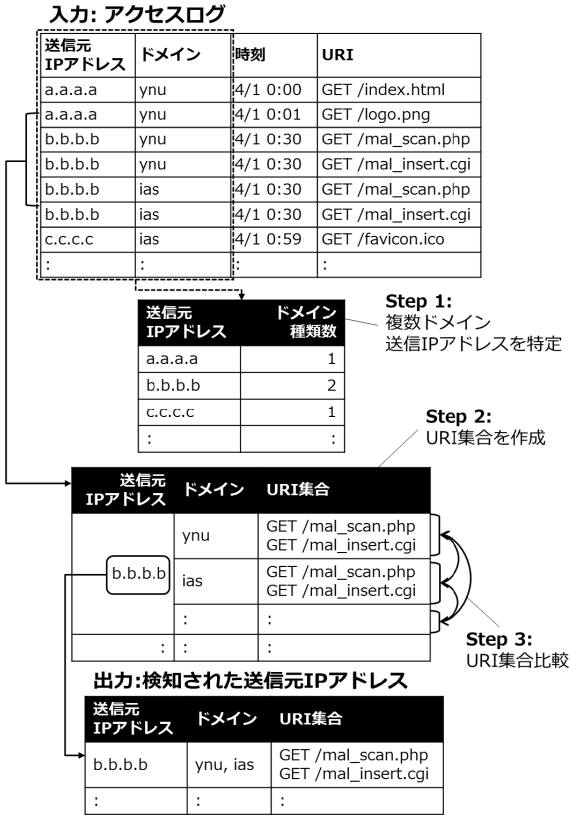



図 2 提案手法における送信元 IP アドレス抽出処理の例
**Fig. 2** An example of extraction in proposal system applying to access log.

提案手法の適用例を**図 2** に示す．図 2 では，まず Step

1 で入力となるアクセスログ中の送信元 IP アドレスとドメインの 2 項目に着目し，送信元 IP アドレスごとに送信先となったドメイン種類数を集計する．この集計の結果，b.b.b.b の送信先ドメイン種類数が 2 種類であったため，この b.b.b.b を複数ドメインに送信した送信元 IP アドレスと判断する．次に，Step 2 の処理では，複数ドメインに送信した送信元 IP アドレスを含むアクセスログを対象として，ドメインごとに URI 集合を作成する．アクセスログより，b.b.b.b は ynu に対しては GET /mal_scan.php および GET /mal_insert.cgi を，ias に対しても GET /mal_scan.php および GET /mal_insert.cgi を送信していた．そのため，b.b.b.b は ynu について URI 集合 { GET /mal_scan.php, GET /mal_insert.cgi} が，ias についても URI 集合 { GET /mal_scan.php, GET /mal_insert.cgi} が作成される．そして Step 3 では，1 つの送信元 IP アドレスについて作成された URI 集合を比較する．Step 2 で b.b.b.b について，ynu について作成できた URI 集合と ias について作成できた URI 集合を比較する．これら 2 種類の URI 集合に含まれる要素は一致するため，b.b.b.b は ynu と ias に対して共通の URI 群（GET /mal_scan.php，GET /mal_insert.cgi）を送信したと判断する．この b.b.b.b の場合では，ドメイン種類数は 2，URI 集合の種類数は 2 となる．

本手法により，検知パターンが存在しなかったり，単一のドメインでは異常性を判断できなかったりする攻撃であっても，しきい値を超える種類のドメインに対して，しきい値を超える種類の共通した URI を送信した送信元 IP アドレスならば，提案手法では悪性として抽出することが可能である．

次章では，提案手法を実際の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，ドメイン種類数および URI 種類数のしきい値を変化させたときの，提案手法の検知精度を計算した結果を報告する．

## 4. 評価実験

本章では，3 章で提案したアクセスログ分析手法を実際のホスティングサービスのアクセスログに適用した結果を報告する．提案手法と既存ツールである IDS の snort [5] と WAF の modsec [8] を比較した．本評価実験では，提案手法のパラメータを変化させながら，誤検知率および攻撃見逃し率を計算し，提案手法がどの程度攻撃を検知できるのかを示す．

### 4.1 適用対象とするアクセスログ

横浜国立大学では，学内組織や教職員を対象とした Web サイトホスティングサービスを運用している．このサービスでは，学内の部局や研究室等の Web サイトが運用されており，学内外からのアクセスをアクセスログに記録している．

表 2 クローラの判断基準
**Table 2** Crawlers domain name list.

| 検索エンジン | ドメイン名における判断基準 |
|---|---|
| Baidu | crawl.baidu.com で終わる． |
| Google | googlebot.com で終わる．<br>あるいは rate-limited-proxy-で<br>始まり google.com で終わる． |
| msn | search.msn.com で終わる． |

適用対象としたのは 2015 年 6 月 26 日から 9 月 23 日の 90 日間に発生したアクセスのうち，検索エンジンによるクローラと判断された送信元 IP アドレスによるアクセスを除外したものである．クローラの判断においては，まず Baidu [25]，Google [26], [27]，msn [28] を対象として，逆引きドメイン名が**表 2** に該当する送信元 IP アドレスをクローラによるものと判断した．

Baidu と msn と同じ URI 集合を送信した送信元 IP アドレスもクローラと判断した．適用対象としたアクセスログの規模は，レコード件数が 44,177,946 件，送信元 IP アドレスが 63,084 種類，ドメイン種類数が 255 種類，URI が 1,062,433 種類であった．

本章の実験では，提案手法により（a）1 日単位で攻撃元を検知する場合，（b）1 時間単位で攻撃元を検知する場合，の 2 通りを実験した．

### 4.2 正解データの作成

本節では，他ツールにより悪意ありと判断されたリクエストを送信した送信元 IP アドレスを抽出する手順について述べる．本手順は 3 つの手順からなる．

**Step 1：他ツールにより悪性として検知されたリクエストを送信した送信元 IP アドレスの抽出**

Step 1 では，URI を既存ツールに適用し，悪性と判断されるリクエストの収集を行った．

まず，実験マシン内でテスト用 Web サイトと，送信対象 URI 一覧に格納されたリクエスト文字列を HTTP リクエストとしてテスト用 Web サイトに対して送信するリクエスト生成器を構築した．構築した Web サイトには文字列「Hello World!」が書かれた index.html のみ設置されている．このリクエスト生成器を用いて，評価対象 URI 一覧へアクセスログ中の URI を登録し，テスト用 Web サイトに対して HTTP リクエストを送信する．テスト環境の構成を**図 3** に示す．リクエスト生成器では，評価対象 URI 一覧に格納された URI に対してホスト名やヘッダ情報を補完し，テスト用 Web サイト（http://localhost/）に対する HTTP リクエストを生成し，送信する．この Web サイトに対するアクセスを，オープンソースの IDS である Snort および WAF の modsec を適用して監視し，これらのツー

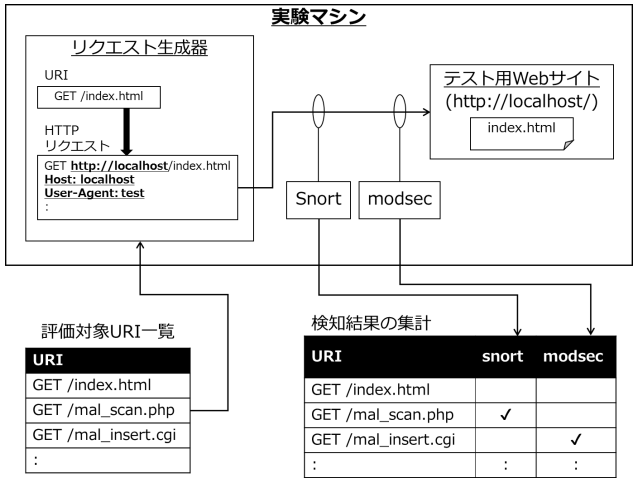



図 3 既存ツールとの比較を行った環境

**Fig. 3** The environment for comparison of other existing tools.

ルが HTTP リクエストを検知するかを記録した[*3,*4]．各ツールのどちらか一方が攻撃と検知したならば，当該 URI を攻撃と判断した．我々は，このテスト環境を用いて適用対象としたアクセスログに記録された URI に対する snort および modsec による検知結果を得た．

**Step 2：独自シグネチャの作成およびマッチング**

Step 1 では Snort および modsec により Web サイトへの攻撃に関する正解データの作成を試みたが，Web サイトへの攻撃は多岐にわたるため，これらの既存ツールにも見逃しが予想される．そこでこれらのツールに加えて，セキュリティサイト等の情報をもとに攻撃パターンを正規表現として手動で書き下した独自シグネチャを作成した．独自シグネチャを付録 A.9 に示す．

**Step 3：悪意ある送信元 IP アドレスの判断**

Step 1 および Step 2 の手順により悪意ありと判断された URI と，送信元 IP アドレスが送信した URI とを比較する．URI がすべて攻撃と判断されたならば，当該送信元 IP アドレスは悪意ある URI を送信したと判断した．

### 4.3 正解データとして抽出できた送信元 IP アドレス

上述の手順に従って正解データを作成した結果，適用対象アクセスログの期間を (a) 1 日とした場合では合計 9,364 種類の送信元 IP アドレスが，(b) 1 時間とした場合では合計 15,967 種類の送信元 IP アドレスが，それぞれ抽出できた．適用期間 (a) と (b) それぞれについて，期間ごとに抽出できた送信元 IP アドレスのドメイン種類数の分布を，(a) と (b) の場合をそれぞれ**図 4**，**図 5** に示す．送信元 IP アドレスの変化および URI 種類数の分布を，(a) の場合を付録 A.1，A.2 に，(b) の場合を付録 A.3，A.4 にそれぞれ示す．

*3 Snort ルールセットの取得日は 2016/12/21，modsec ルールセットの取得日は 2017/2/28 である．

*4 実験の際には，リクエスト文字列を判断対象とするように調整したルールセットを使用した．

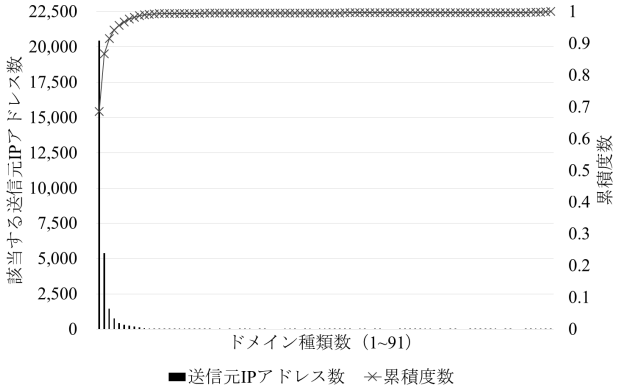



図 4 (a) の場合に正解データと判断した送信元 IP アドレスのドメイン種類数の分布

**Fig. 4** The number of destination domains distribution of correct source IP addresses in the case (a).

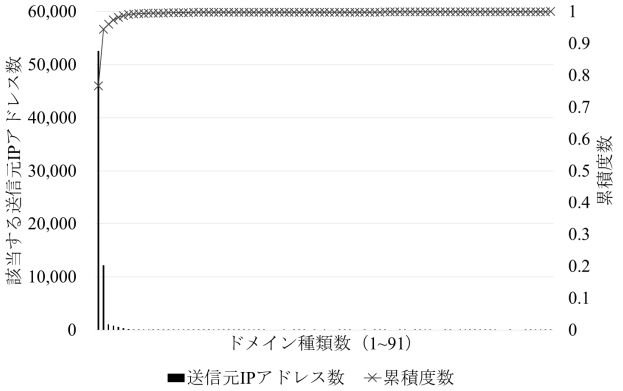



図 5 (b) の場合に正解データと判断した送信元 IP アドレスのドメイン種類数の分布

**Fig. 5** The number of destination domains distribution of correct source IP addresses in the case (b).

### 4.4 提案手法の適用結果抽出できた送信元 IP アドレス

アクセスログを提案手法に適用した結果，適用対象アクセスログの期間を（a）1 日とした場合では合計 9,551 種類の送信元 IP アドレスが，（b）1 時間とした場合では合計 15,361 種類の送信元 IP アドレスが，それぞれ抽出できた．1 章で述べた，文献 [6] で報告された ShellShock の脆弱性を狙った URI 群を送信した IP アドレスも，(a) と (b) それぞれの場合において提案手法で抽出できたことを確認した．**表 3** に，この URI 群の送信元となった IP アドレス種類数と送信先ドメイン種類数を示す．表 3 では，たとえば，(a) の場合に，7 種類の送信元 IP アドレスが 90 種類のドメインに対してこの URI 群を送信していたことを示す．

適用期間 (a) と (b) それぞれについて，期間ごとに抽出できた送信元 IP アドレスのドメイン種類数種類数の分布を，(a) と (b) の場合をそれぞれ**図 6**，**図 7** に示す．送信元 IP アドレスの変化および URI 種類数の分布を，(a) の場合を付録 A.5，A.6 に，(b) の場合を付録 A.7，A.8 にそれぞれ示す．

正解データとして抽出できた送信元 IP アドレスと提案

**表 3** (a) と (b) の場合において，提案手法で抽出できた ShellShock の脆弱性を狙った URI 群に紐づく送信元 IP アドレスと送信先ドメイン

**Table 3** Source IP addresses and destination domains connected to the extracted ShellShock included URI group extracted by proposal method in case (a) and (b).

| | 送信元 IP アドレス種類数 | 送信先ドメイン種類数 |
|---|---|---|
| (a) | 7 | 90 |
| | 1 | 4 |
| | 1 | 91 |
| (b) | 6 | 90 |
| | 2 | 4 |
| | 1 | 91 |
| | 1 | 87 |

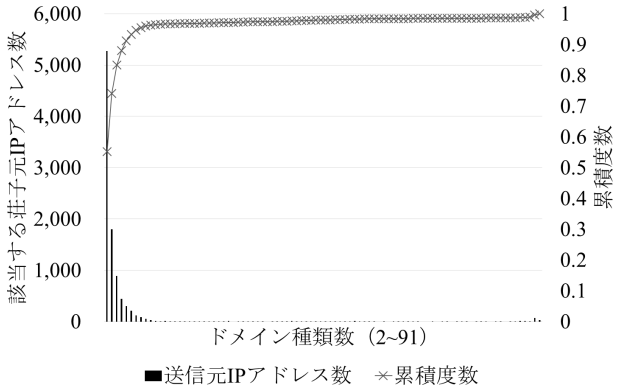



**図 6** (a) の場合に提案手法が抽出した送信元 IP アドレスのドメイン種類数の分布

**Fig. 6** The number of destination domains distribution of source IP addresses extracted by proposal method in the case (a).

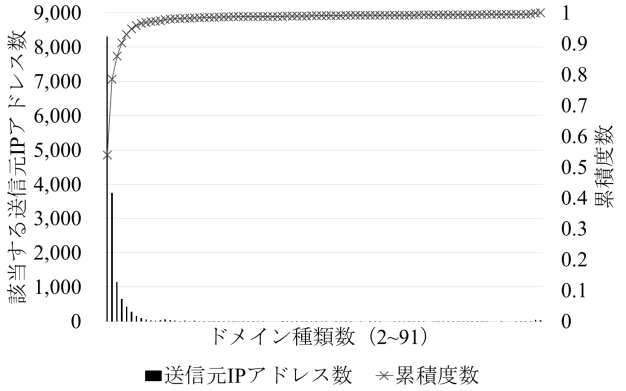



**図 7** (b) の場合に提案手法が抽出した送信元 IP アドレスのドメイン種類数の分布

**Fig. 7** The number of destination domains distribution of source IP addresses extracted by proposal method in the case (b).

**表 4** 評価対象となった送信元 IP アドレスの基本統計量

**Table 4** Standard statistics for correct and proposal method extracted source IP addresses in case (a) and (b).

| | | | 最小値 | 最大値 | 平均値 |
|---|---|---|---|---|---|
| 単位期間当たりに抽出できた送信元 IP アドレス数 | (a) | 提案手法 | 30 | 454 | 106.12 |
| | | 正解データ | 17 | 560 | 104.04 |
| | (b) | 提案手法 | 2 | 26 | 7.17 |
| | | 正解データ | 1 | 117 | 7.87 |
| ドメイン種類数 | (a) | 提案手法 | 2 | 91 | 4.97 |
| | | 正解データ | 2 | 91 | 4.37 |
| | (b) | 提案手法 | 2 | 91 | 3.98 |
| | | 正解データ | 2 | 91 | 3.36 |
| URI 種類数 | (a) | 提案手法 | 1 | 304 | 1.33 |
| | | 正解データ | 1 | 96 | 1.78 |
| | (b) | 提案手法 | 1 | 337 | 1.27 |
| | | 正解データ | 1 | 96 | 2.17 |

手法の適用結果抽出できた送信元 IP アドレスを対象に，単位期間あたりの送信元 IP アドレス数，ドメイン種類数，URI 種類数について，最小値，最大値および平均値を比較した結果を**表 4** に示す．まず，提案手法で抽出できた送信元 IP アドレスと正解データとして抽出した送信元 IP アドレスを比較する．単位期間あたりに抽出できた送信元 IP アドレス数については，(a) と (b) のどちらの場合でも，提案手法よりも正解データの方が多かった．また，URI 種類数についても，提案手法よりも正解データの方が最大値・平均値ともに多かった．URI 種類数の分布に着目すると，特に (b) の場合に，正解データでは 2〜3 種類の URI を送信した送信元 IP アドレスが全体の約 19.4%存在した．これに対し，提案手法では 2〜3 種類の URI を送信した送信元 IP アドレスが全体の約 78.5%であった．このことから，1 種類のみのドメインに URI を送信した送信元 IP アドレスが多く，それらが送信した URI 種類数も多かったことが分かる．図 4 および図 5 に示したドメイン種類数の分布からも，URI が共通か否かにかかわらず，2 種類以上のドメインに URI を送信した送信元 IP アドレスは正解データ全体と比較すると少ないことが分かる．次に，提案手法で抽出できた送信元 IP アドレスについて，(a) と (b) の場合を比較すると，ドメイン種類数の平均値の差は約 0.99 であった．ドメイン種類数の分布を比較すると，(a) よりも (b) の場合の方が，Web サイト種類数が 2 だった送信元 IP アドレスが占める割合が大きかった．一方で，URI 種類数に関しては，平均値の差は約 0.06 であった．このことから，複数種類のドメインに対して共通した URI を送信した送信元 IP アドレスの多くは，1 時間以内に URI を送信し終えていたといえる．

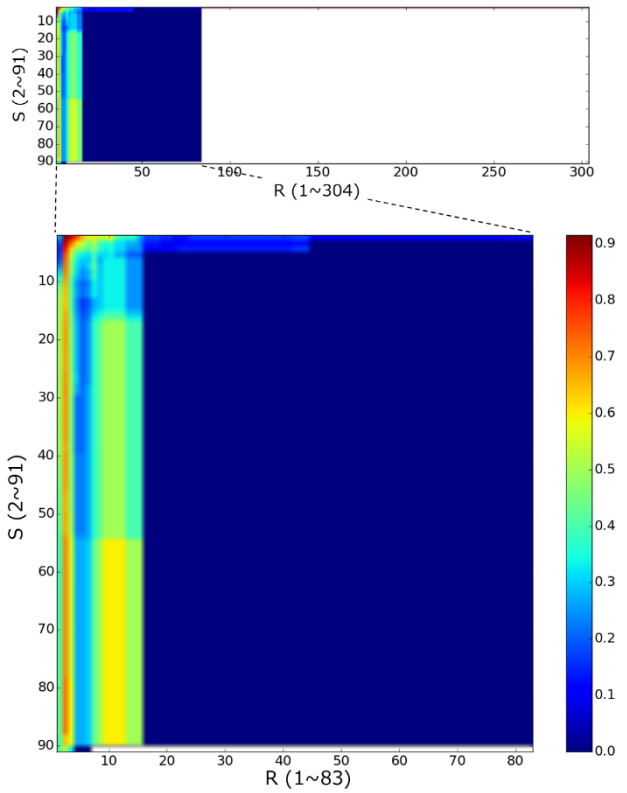



図 8 (a) の場合における FP の変化
**Fig. 8** False positive ratio in case (a).

### 4.5 FP および FN の計算結果

提案手法を適用した結果について，ドメイン種類数のしきい値 S および URI 種類数のしきい値 R を変化させたときの誤検知率（False Positive，FP）と見逃し率（False Negative，FN）を，それぞれヒートマップ形式で可視化した．FP および FN は次式により計算した．

- FP：(Snort，modsec，独自シグネチャのいずれかの手段が検知しなかったものの，提案手法が検知した送信元 IP アドレス数)/(提案手法が検知した送信元 IP アドレス数)
- FN：(Snort，modsec，独自シグネチャのいずれかの手段が検知したものの，提案手法が検知しなかった送信元 IP アドレス数)/(Snort，modsec，独自シグネチャのいずれかの手段が検知した送信元 IP アドレス数)

(a) 1 日で区切った場合における FP の変化を**図 8** に，FN の変化を**図 9** に，(b) 1 時間で区切った場合における FP の変化を**図 10** に，FN の変化を**図 11** に，それぞれ示す．ただし，ヒートマップ内の白色の領域は，該当する送信元 IP アドレスが存在しなかったことを示す．

ヒートマップ中の色の分布から，(a) と (b) のどちらの場合においても，総じて誤検知率は小さいものの，見逃し率は高い結果となった．FP の変化に関しては，(a) と (b) の両方において，S も R もどちらも小さい値に設定した場合には，FP が大きくなった．FN の変化に関しては，(a) と (b) の両方において R が 84 以上のときには見逃し率が

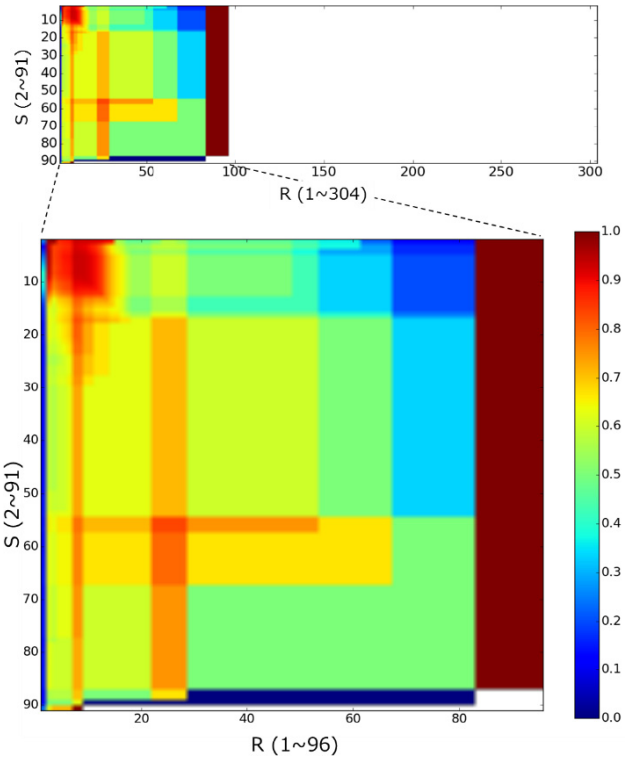



図 9 (a) の場合における FN の変化
**Fig. 9** False negative ratio in case (a).

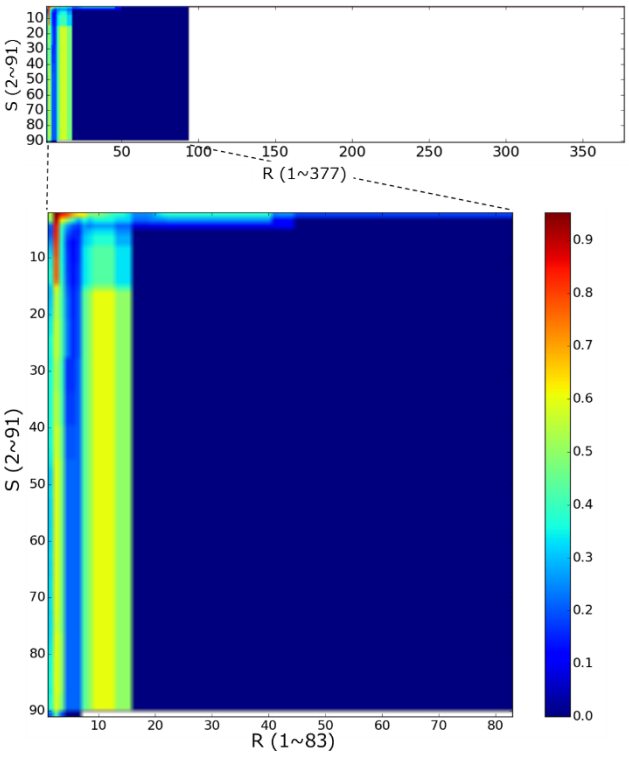



図 10 (b) の場合における FP の変化
**Fig. 10** False positive ratio in case (b).

1 となった．これは，提案手法では URI 種類数が 84 以上であった送信元 IP アドレスを抽出できなかったことを示している．

次に，(a) と (b) の場合において FP および FN がそれぞれ 0 となる S と R の範囲を**表 5** に示す．特に FP が 0 と

表 5 (a) および (b) の場合における FP = 0，FN = 0 となる S と R の範囲
**Table 5** S and R ranges when FP = 0 and FN = 0 in case (a) and (b).

| | | S の範囲 | R の範囲 | FP | | FN | |
|---|---|---|---|---|---|---|---|
| | | | | 最小値 | 最大値 | 最小値 | 最大値 |
| (a) | FP=0 | S=91 | 4≦R≦6 | 0 | 0 | 0.714 | 0.714 |
| | | 5≦S≦90 | 16≦R≦44 | 0 | 0 | 0 | 0.833 |
| | | 3≦S≦90 | 45≦R≦83 | 0 | 0 | 0 | 0.75 |
| | FN=0 | S=90 | 9≦R≦28 | 0 | 0.6 | 0 | 0 |
| | | 88≦S≦90 | 29≦R≦83 | 0 | 0 | 0 | 0 |
| (b) | FP=0 | S=91 | 4≦R≦6 | 0 | 0 | 0.333 | 0.429 |
| | | 5≦S≦90 | 16≦R≦44 | 0 | 0 | 0 | 0.857 |
| | | 3≦S≦90 | 45≦R≦83 | 0 | 0 | 0 | 0.8 |
| | FN=0 | S=90 | 9≦R≦28 | 0 | 0.6 | 0 | 0 |
| | | 88≦S≦90 | 29≦R≦83 | 0 | 0 | 0 | 0 |

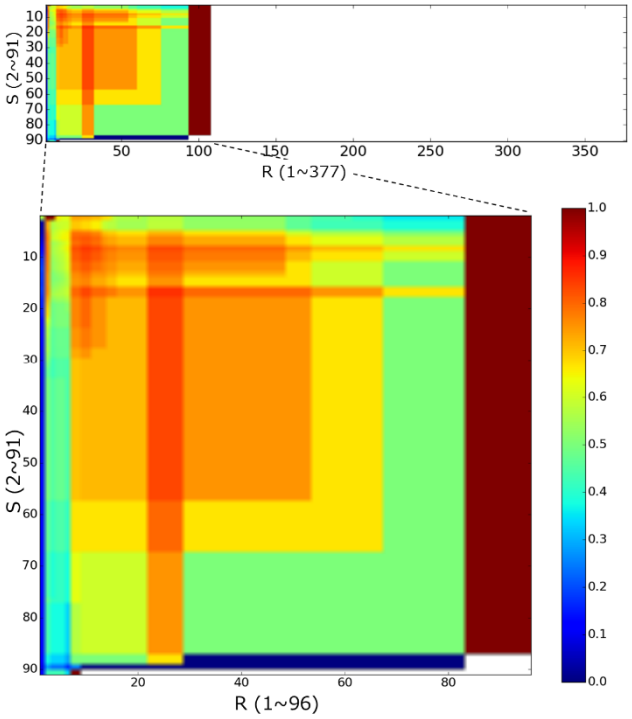



図 11 (b) の場合における FN の変化
**Fig. 11** False negative ratio in case (b).

なった範囲は，(a) と (b) の場合どちらも同じ範囲で，次の 3 種類であった．

- **領域 $FP_1$**：S = 91，4 ≦ R ≦ 6
- **領域 $FP_2$**：5 ≦ S ≦ 90，16 ≦ R ≦ 44
- **領域 $FP_3$**：3 ≦ S ≦ 90，45 ≦ R ≦ 83

特に領域 $FP_2$ および領域 $FP_3$ については，88 ≦ S ≦ 90，29 ≦ R ≦ 83 の範囲においては，FP だけでなく，FN も 0 となった．

したがって，攻撃を誤検知なしに検知可能な範囲として，下記の 2 種類の範囲が設定可能であるといえる．まず，送信対象のリクエスト種類数が少なくても，ドメイン種類数が多かった範囲である．本評価実験では，領域 $FP_1$ である，ドメイン種類数が 91 以上かつ URI 種類数が 4 以上に該当する．次に，ドメイン種類数が少なくても，送信対象の URI 種類数が多かった範囲である．本評価実験では，領域 $FP_2$ および領域 $FP_3$ が重なる領域である，ドメイン種類数が 5 以上かつ URI 種類数が 45 以上に該当する．

表 6 FP = 0 の領域における独自シグネチャにより悪性と判断できた送信元 IP アドレスの割合
**Table 6** Source IP address ratio detected by original signatures in case of no false positives.

| | 領域 | 独自シグネチャ含有送信元 IP アドレスの割合 |
|---|---|---|
| (a) | 領域 $FP_1$ | 0 |
| | 領域 $FP_2$ | 1 |
| | 領域 $FP_3$ | 0 |
| (b) | 領域 $FP_1$ | 0 |
| | 領域 $FP_2$ | 0.5 |
| | 領域 $FP_3$ | 0 |

領域 $FP_1$: S=91, 4≦R≦6
領域 $FP_2$: 5≦S≦90, 16≦R≦44
領域 $FP_3$: 3≦S≦90, 45≦R≦83

FP = 0 となった領域について，独自シグネチャにより悪性と判断できた送信元 IP アドレス（独自シグネチャ含有送信元 IP アドレス）の割合を，**表 6** に示す．この表から，(a) と (b) どちらの場合も，領域 $FP_1$ および領域 $FP_3$ は既存ツールで検知できた攻撃のみが含まれていた．一方，領域 $FP_2$ には，既存ツールでは検知できなかった，独自シグネチャでのみ検知できた攻撃も含まれていた．この表の結果から，FP = 0 となるしきい値を設定したとき，既存のツールでは検知できなかった攻撃を，提案手法では検知で

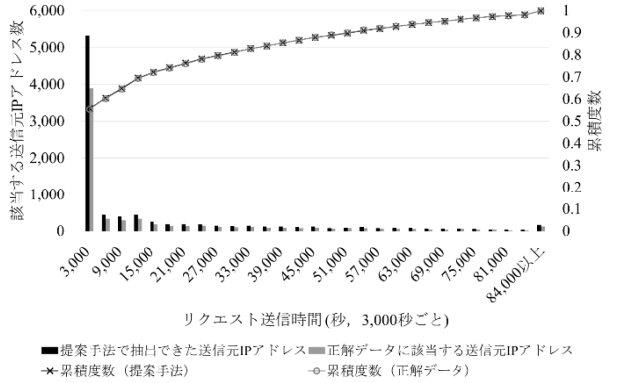



**図 12** (a) の場合におけるリクエスト送信時間と送信元 IP アドレスの分布

**Fig. 12** The number of sending periods distribution of source IP addresses in case (a).

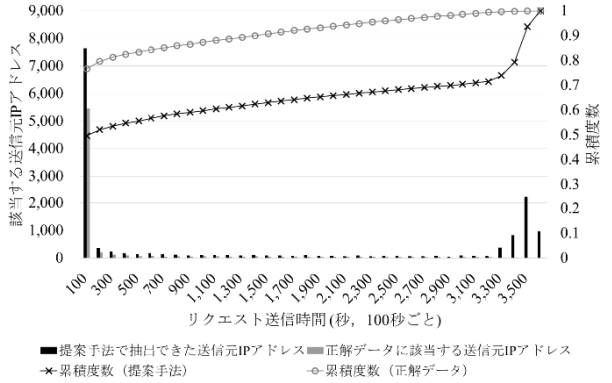



**図 13** (b) の場合におけるリクエスト送信時間と送信元 IP アドレスの分布

**Fig. 13** The number of sending periods distribution of source IP addresses in case (b).

きたことが確認できた．

独自シグネチャでのみ検知できた攻撃について述べる．領域 $FP_2$ には 2 種類の攻撃を行った送信元 IP アドレスが含まれていた．まず，Asset Manager を通して任意のファイルのアップロードが可能な assetmanager モジュールを見つけるため [29]，assetmanager.asp，assetmanager.aspx のように拡張子を変えながら URI を送信した送信元 IP アドレスである．次に，FCKeditor を通して任意のファイルのアップロードが可能な connecter モジュールについて，connector.asp，connector.aspx のように拡張子を変えながら URI を送信した送信元 IP アドレスである．Snort には拡張子 “.asp” を含む URI を検知するシグネチャが存在したため，これらの送信元 IP アドレスが送信した URI の中には Snort でも攻撃と判断した URI が含まれていた．しかしそれ以外の拡張子を含む URI は攻撃と判断されなかった．また，connector モジュールに対して URI を送信した送信元 IP アドレスには，Snort がマルウェアによるファイルのアップロード試行を行うために攻撃と検知される URI を含む送信元 IP アドレスも存在した．以上の結果から，作成したシグネチャに不足がある場合であっても，提案手法では悪意ある URI を送信した IP アドレスを抽出できることを確認できた．

### 4.6 リクエストの収集に必要な時間

本節では，提案手法が送信元 IP アドレスの悪性の有無を判断するため，リクエストの収集に必要な時間を評価する．適用対象アクセスログの期間 (a) および (b) の場合において，提案手法で抽出できた送信元 IP アドレスがアクセスログに出現した最後の時刻から最初の時刻を引いた値をリクエスト送信時間として計算した．

リクエスト送信時間の分布を，(a) の場合，(b) の場合をそれぞれ**図 12**，**図 13** に示す．各図中の黒色の棒は提案手法で抽出できた送信元 IP アドレスを，灰色の棒は提案手法で抽出でき，かつ正解データに該当する送信元 IP アドレスを示す．また，(a)，(b) それぞれの場合において累積度数が 50%，75%，95%以上の下限となるリクエスト送信時間を**表 7** に示す．表 7 では，たとえば，(a) の場合に提案手法で抽出できた送信元 IP アドレスが累積度数 50%以上となる，つまり送信元 IP アドレスの占める割合が半分以上となる下限は，リクエスト送信時間が 3,500 秒であることを示す．

**表 7** (a) および (b) の場合における累積度数が 50%，75%，95%以上の下限となるリクエスト送信時間

**Table 7** The lower sending periods for cumulative frequency 50%, 75% and 95% in case (a) and (b).

| | | 累積度数 | | |
|---|---|---|---|---|
| | | 50%以上 | 75%以上 | 95%以上 |
| (a) | 提案手法で抽出できた | 3,000 秒 | 21,000 秒 | 69,000 秒 |
| | 正解データに該当 | 3,000 秒 | 21,000 秒 | 69,000 秒 |
| (b) | 提案手法で抽出できた | 500 秒 | 3,400 秒 | 3,600 秒 |
| | 正解データに該当 | 100 秒 | 100 秒 | 2,300 秒 |

まず (a) の場合について，リクエスト送信時間が 3,000 秒以下であった送信元 IP アドレスは，提案手法で抽出できたものが約 55.8%，正解データに該当するものが約 55.5%であった．どちらの送信元 IP アドレスも半数強が，リクエスト送信時間が 3,000 秒以下であったといえる．正解データに該当する送信元 IP アドレスを 95%以上含むのは，リクエスト送信時間が 69,000 秒以下の範囲である．

次に (b) の場合について，リクエスト送信時間が 100 秒以下だった送信元 IP アドレスは，提案手法で抽出できたものが約 49.7%，正解データに該当するものが約 76.7%であった．さらに，提案手法で抽出できた送信元 IP アドレスは 3,400 秒より大きく 3,500 秒以下の範囲にも多く集中して

いた．正解データに該当する送信元 IP アドレスを 95%以上含むのは，リクエスト送信時間が 2,300 秒以下の範囲である．

なお，(a) の場合においてリクエスト送信時間が長かった送信元 IP アドレスには，下記のように，Web ホスティングサービス下の Web サイトにリクエストが断続的に到達したものが存在した．

| # | 送信元 IP アドレス | ドメイン | 時刻 | URI |
|---|---|---|---|---|
| 1 | x.x.x.x | domainA | 00:24:44 | POST /xmlrpc.php |
| 2 | x.x.x.x | domainB | 04:36:51 | POST /xmlrpc.php |
| 3 | x.x.x.x | domainC | 04:36:58 | POST /xmlrpc.php |
| : | | : | : | : |

この事例では，設定された Web サイト一覧に対して順番にリクエストを送信しており，その一覧に今回評価対象とした Web ホスティングサービス管理下の Web サイトが含まれていたと考えられる．これらのリクエストを送信した IP アドレスを提案手法で判断できるのは，1 行目のリクエストと，それが到達してから 4 時間 12 分後に到達した 2 行目のリクエストを読み込んだ時点となる．あるいは，数時間情報を保持しない場合であっても，2 行目のリクエストと，それが到達してから 7 秒後に到達した 3 行目のリクエストを読み込んだ時点でも，提案手法では IP アドレスの判断が可能である．ただしこの場合，1 行目のリクエストは提案手法では判断の対象とはならない．

### 4.7 提案手法で観測すべきドメイン種類数

本節では，提案手法適用時に観測すべきドメインの規模について評価する．提案手法では観測対象とするドメインの種類が多ければ多いほど，より高い精度で悪意あるリクエストを送信した IP アドレスを抽出できる可能性が大きくなる．しかし，ドメインの種類が多くなれば提案手法が読み込むべきアクセスログの量や処理内で行われる URI 集合のドメイン間の比較回数が増加するため，提案手法の処理に必要なメモリや時間も増加する．

適用対象アクセスログの期間 (a) および (b) の場合において，送信先となったドメインそれぞれについて，提案手法で抽出できた送信元 IP アドレス数を，提案手法で抽出できた送信元 IP アドレスおよび提案手法で抽出でき，かつ正解データに該当する送信元 IP アドレスについてそれぞれ数え上げた．数え上げた結果を，提案手法で抽出できた送信元 IP アドレスについて昇順で並べ替えた結果を**図 14**，**図 15** に示す．各図中の黒色三角形のマーカは提案手法で抽出できた送信元 IP アドレスを，灰色四角形のマーカは提案手法で抽出でき，かつ正解データに該当する送信元 IP アドレスを示す．

まず (a) の場合について，提案手法で抽出できた送信元

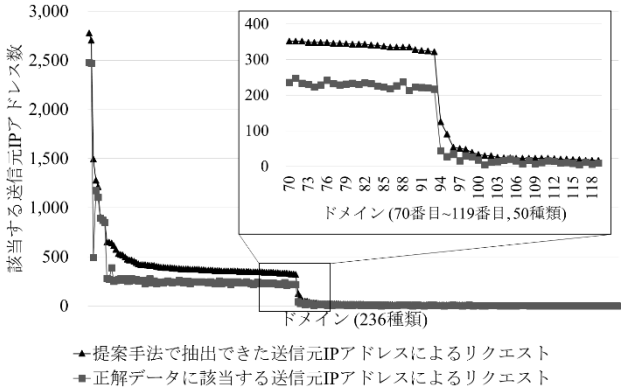



**図 14** (a) の場合におけるドメインにリクエストを送信した IP アドレス数の分布

**Fig. 14** The number of source IP address distribution of destination domains in case (a).

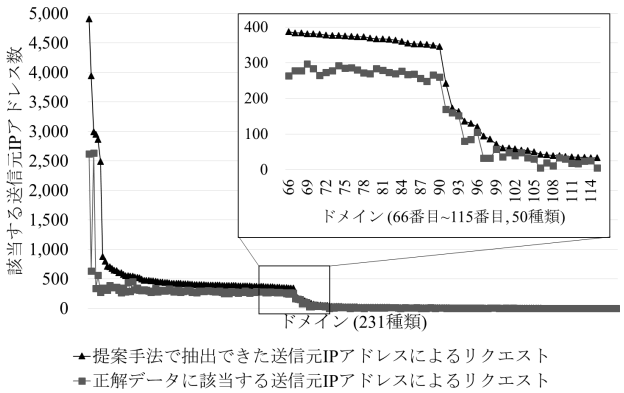



**図 15** (b) の場合におけるドメインにリクエストを送信した IP アドレス数の分布

**Fig. 15** The number of source IP address distribution of destination domains in case (b).

IP アドレスの送信先ドメインを数え上げた結果，上位 5 種類のドメインは 1,000 を超える IP アドレスから URI が送信されていた．さらに，上位 94 番目と 95 番目のドメインを送信先とした IP アドレス数には 196 の差があり，それ以降のドメインには 1 個の送信元 IP アドレスのみが紐づく状況となっている．正解データに該当する送信元 IP アドレスの場合についても，同様の箇所で折れ線グラフが落ち込んでいる．

次に (b) の場合について，提案手法で抽出できた送信元 IP アドレスの送信先ドメインを数え上げた結果，上位 5 種類のドメインは 2,400 を超える IP アドレスから URI が送信されていた．さらに，上位 90 番目と 91 番目のドメインを送信先とした IP アドレス数には 104 の差があり，それ以降のドメインには 1 個の送信元 IP アドレスのみが紐づく状況となっている．正解データに該当する送信元 IP アドレスの場合についても，同様の傾向がある．

以上より，本実験の対象とした Web ホスティングサービス管理下の Web サイト（ドメイン）は 255 種類であったが，提案手法に関与したドメインは (a) の場合で 236 種

類，(b) の場合で 231 種類であった．つまり，実際は Web ホスティングサービス管理下のドメインすべてに悪意あるリクエストが送信されていなかった．さらに，(a) の場合では 95 番目，(b) の場合では 91 番目以降のドメインに対して URI を送信した IP アドレス数は，それら以前と比較すると 200 規模の差が開いていた．このことから，これらのドメインは観測対象から除外するといった，観測すべきドメインの選定も可能である．

## 5. 議論

### 5.1 提案手法が誤検知と判断する URI

本節では，誤検知に該当した送信元 IP アドレスについて事例をあげ，提案手法が誤検知と判断する URI について考察する．

誤検知に該当した送信元 IP アドレスが送信した URI 集合を集計し，該当する送信元 IP アドレスが多かった URI 集合の上位 5 を**表 8** に示す．ただし，表中の「null」は URI の値がログに存在しなかったことを示す．表 8 中の，#4 については文献 [30] より Microsoft Internet Explorer 11 等によるサムネイル画像の要求を含むリクエスト，#5 については Web サイトの情報を収集するクローラによるリクエストにより送信された URI である可能性が高いと判断できる[*5]．また，#3 については，複数の Web サイトを人間が閲覧するためにアクセスしたケースも含まれると考えられる．本実験の評価対象は学内の部局や研究室等の Web サイトに対するアクセスログである．そのため，数種類程度ならば人間が手動でトップページをアクセスして回った，といったシナリオも考えられる．

以上より，提案手法の適用結果，誤検知となる URI は主に下記の 3 種類の場合であると考えられる．

(1) Web ブラウザ等送信元側の環境で自動的に送信されるリクエスト
(2) 複数 Web サイトの情報収集を目的とするリクエスト
(3) 複数 Web サイト間で共通の名前を持つページを要求するリクエスト

こうした誤検知に対して，(1) および (3) の場合に関しては，提案手法の処理の対象外とする URI をホワイトリスト化によって，誤検知を減らせる場合がある．

まず，(1) の場合に関しては，送信元側の環境で自動的に送信される URI であることが明らかになれば，これらの URI をホワイトリストに追加できる．次に，(3) の場合に関しては，今回の実験で利用したアクセスログは大学内の Web サイトを運用する Web ホスティングサービスであった．そのため，たとえば学内の部局の Web サイトにアクセスした際に，大学のロゴ画像や部局間で共通して利用されるレイアウトに関する JavaScript や CSS ファイルをリクエストしたとみられるケースがあった．こうしたコンテンツをリクエストする際に送信される URI も，ホワイトリスト化が可能であると考えられる．

ただし，上述でホワイトリスト化した URI が Web サイトの脆弱性発見に有用であることが分かった等，攻撃に用いられることが分かった場合に，攻撃目的で URI が送信されたとしても，提案手法では処理の対象外となり，攻撃事象の抽出ができない．また，今回の Web ホスティングサービスのようにサービス利用者や利用目的が限定されていない，不特定多数の利用者が各々異なる目的でサービスを利用する形態の Web ホスティングサービスでは，特に (3) の場合のホワイトリスト化は難しいと考えられる．

**表 8** 誤検知に該当する送信元 IP アドレスが送信した URI 集合上位 5
**Table 8** Top 5 request sets that false positive source IP addresses sent.

| # | URI 集合 | 送信元 IP アドレス数 | Web サイト種類数 | | |
|---|---|---|---|---|---|
| | | | 最小値 | 最大値 | 平均値 |
| 1 | GET:/favicon.ico | 1,005 | 2 | 14 | 2.33 |
| 2 | HEAD:/ | 258 | 2 | 91 | 8.36 |
| 3 | GET:/<br>GET:/favicon.ico | 200 | 2 | 10 | 2.82 |
| 4 | null<br>GET:/browserconfig.xml | 179 | 2 | 4 | 2.27 |
| 5 | GET:/<br>GET:/robots.txt<br>GET:/sitemap.xml | 77 | 2 | 9 | 2.44 |

[*5] robots.txt や sitemap.xml は，Web サイトの情報を収集するために攻撃者が利用する場合も考えられるため，クローラの挙動として除外せず，本稿では提案手法の適用対象とした．

### 5.2 提案手法の限界

提案手法の限界について考察する．提案手法では，2 以上の複数種類のドメインに対して，共通の URI を送信した送信元 IP アドレスを悪意ありと判断する．そのため，1 種類のドメインにしか URI を送信しなかった送信元 IP アドレス，複数種類のドメインに送信した URI が共通でなかった送信元 IP アドレスは，提案手法では抽出の対象から除外される．

よって，提案手法では抽出が難しい悪性 IP アドレスは，下記の 2 種類であるといえる．

(1) 特定の 1 種類の Web サイトに対して悪意あるリクエストを送信した送信元 IP アドレス
(2) 複数種類の Web サイトに対してそれぞれ異なる悪意あるリクエストを送信した送信元 IP アドレス

提案手法を拡張することで，(2) に該当する送信元 IP アドレスも悪性と判断できる場合がある．提案手法では，Web サイトごとに送信した URI 集合を比較する際に，各 URI 集合が完全に一致した送信元 IP アドレスを抽出して

いた．この処理を，完全一致ではなく，類似度を計算したうえであるしきい値以上の類似度を持つ送信元 IP アドレスを抽出するような処理に拡張することで，同一の URI を送信していなくても，ある程度共通した URI 群を送信していた送信元 IP アドレスも抽出結果に含めることが可能である．ただしこのような手段をとった場合であっても，Web サイトごとにまったく異なる URI を送信した送信元 IP アドレスは抽出の対象とすることはできないといった限界が存在する．たとえば Web サイトによって GET リクエスト中のパラメータを変化させながら URI を送信した送信元 IP アドレスは，同じ意図を持った URI を送信したとしても，提案手法では抽出できないという限界がある．

## 6. まとめと今後の課題

本稿では，複数ドメインに対するアクセスログを対象として，複数ドメインに送信された悪意あるリクエストを抽出する手法を提案した．提案手法では，一定期間収集したアクセスログから，送信元 IP アドレス，送信先ドメイン，URI の関係性を分析し，複数のドメインに対して同一の URI を送信した IP アドレスを抽出する．本手法により，単一のドメインへのアクセスを分析することでは異常性の判断が難しいリクエストであっても，しきい値を超える種類のドメインに対して，しきい値を超える種類の共通した URI を送信した送信元 IP アドレスならば，提案手法では悪性として抽出することが可能である．

提案手法を，実際の 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結果，攻撃元となった IP アドレスを誤検知なく抽出できるしきい値が存在することを示した．さらに，既存のオープンソースの IDS および WAF ではシグネチャが登録されておらず検知できない攻撃についても検知できる事例を確認できた．

今後の課題として，リクエスト群の時系列的な関係性にも着目することで，たとえばあるホストからスキャンを受けた後に別のホストから攻撃を受けるといった，複数端末を連携させた攻撃事象の抽出があげられる．

**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いただきました横浜国立大学情報基盤センターに深く感謝いたします．

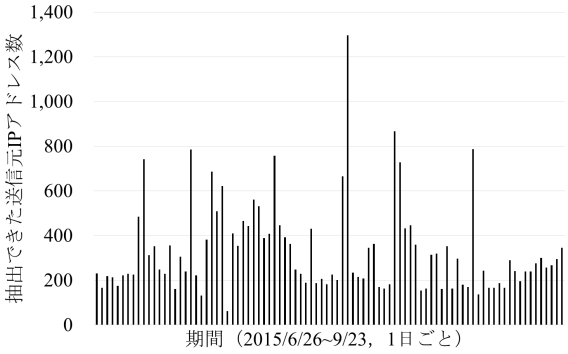
### 参考文献

[1] Blog Tool, Publishing Platform, and CMS - WordPress, available from ⟨https://wordpress.org/⟩ (accessed 2016-11-14).
[2] Joomla! The CMS Trusted By Millions for their Websites, available from ⟨https://www.joomla.org/⟩ (accessed 2016-08-09).
[3] 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！，入手先 ⟨https://lolipop.jp/info/news/4149/⟩（参照 2016-08-01）．
[4] 改ざんの標的となる CMS 内の PHP ファイル（2016-02-25），入手先 ⟨http://www.jpcert.or.jp/magazine/acreport-cms.html⟩（参照 2016-08-01）．
[5] Muieblackcat setup.php Web Scanner/Robot, available from ⟨http://eromang.zataz.com/2011/08/14/suc027-muieblackcat-setup-php-web-scanner-robot/⟩ (accessed 2017-08-30).
[6] New, in your face, malware attacks me:/Ringing.at.your.dorbell!-Dog Is My Copilot, available from ⟨http://www.skepticism.us/2015/05/new-in-your-face-malware-attacks-me-ringing-at-your-dorbell/⟩ (accessed 2017-08-23).
[7] Snort - Network Intrusion Detection & Prevention System, available from ⟨https://www.snort.org/⟩ (accessed 2017-04-10).
[8] ModSecurity: Open Source Web Application Firewall, available from ⟨https://modsecurity.org/⟩ (accessed 2017-04-10).
[9] Kruegel, C., Vigna, G. and Robertson, W.: A multi-model approach to the detection of web-based attacks, *Computer Networks*, Vol.48, No.5, pp.717–738 (2005).
[10] 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).
[11] Sanghyun, C. and Cha, S.: SAD: Web session anomaly detection based on parameter estimation, *Computers & Security*, Vol.23, No.4, pp.312–319 (2004).
[12] Dusan, S., Vlajic, N. and An, A.: Detection of malicious and non-malicious website visitors using unsupervised neural network learning, *Applied Soft Computing*, Vol.13, No.1, pp.698–708 (2013).
[13] Kieyzun, A., Guo, P.J., Jayaraman, K. and Ernst, M.D.: Automatic creation of SQL injection and cross-site scripting attacks, *2009 IEEE 31st International Conference on Software Engineering*, IEEE (2009).
[14] Bisht, P., Hinrichs, T., Skrupsky, N., Bobrowicz, R. and Venkatakrishnan, V.N.: NoTamper: Automatic black-box detection of parameter tampering opportunities in web applications, *Proc. 17th ACM Conference on Computer and Communications Security*, ACM (2010).
[15] Doupe, A., Cavedon, L., Kruegel, C. and Vigna, G.: Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner, *Presented as part of the 21st USENIX Security Symposium* (*USENIX Security 12*) (2012).
[16] Doupe, A., Cova, M. and Vigna, G.: Why Johnny can't pentest: An analysis of black-box web vulnerability scanners, *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2010).
[17] Bau, J., Bursztein, E., Gupta, D. and Mitchell, J.: State of the art: Automated black-box web application vulnerability testing, *2010 IEEE Symposium on Security and Privacy*, IEEE (2010).
[18] John, J.P., Yu, F., Xie, Y., Krishnamurthy, A. and Abadi, M.: Heat-seeking honeypots: Design and experience, *Proc. 20th International Conference on World Wide Web*, ACM (2011).
[19] Yagi, T., Tanimoto, N. and Hariu, T.: Intelligent high-interaction web honeypots based on URL conversion scheme, *IEICE Trans. Communications*, Vol.94, No.5, pp.1339–1347 (2011).
[20] 久世尚美，石倉 秀，八木 毅，千葉大紀，村田正幸：複数のハニーポットにおいて観測された情報に基づく通信

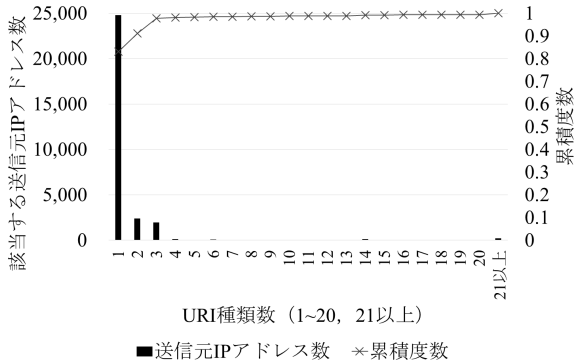
のネットワーク上の特徴を考慮したぜい弱性スキャン識別，信学技報，Vol.115, No.488, ICSS2015-55, pp.47–52 (2016).
[21] Canali, D. and Balzarotti, D.: Behind the scenes of online attacks: An analysis of exploitation behaviors on the web, *Proc. Network and Distributed System Security Symposium* (*NDSS*) (2013).
[22] GitHub - mushorg/glastopf: Web Application Honeypot, available from ⟨https://github.com/mushorg/glastopf⟩ (accessed 2017-04-10).
[23] dionaea, catches bugs, available from ⟨http://dionaea.carnivore.it/⟩ (accessed 2016-06-15).
[24] Christian, K. and Crowcroft, J.: Honeycomb: Creating intrusion detection signatures using honeypots, *ACM SIGCOMM Computer Communication Review*, Vol.34, No.1, pp.51–56 (2004).
[25] 百度一下，就知道，入手先 ⟨http://www.baidu.com/⟩.
[26] Google, available from ⟨https://www.google.co.jp/⟩.
[27] Googlebot かどうかの確認—Search Console ヘルプ，入手先 ⟨https://support.google.com/webmasters/answer/80553?hl=ja⟩（参照 2017-04-10）.
[28] MSN Japan - Hotmail, Outlook.com, Skype, OneDrive, Bing, available from ⟨http://www.msn.com/ja-jp/⟩.
[29] Asset Manager 1.0 - Arbitrary File Upload, available from ⟨https://www.exploit-db.com/exploits/12133/⟩ (accessed 2017-08-31).
[30] IE11 での Web サイト用カスタムタイルの作成（Windows），入手先 ⟨https://msdn.microsoft.com/ja-jp/library/dn455106(v=vs.85).aspx⟩（参照 2017-04-19）.

## 付　　録

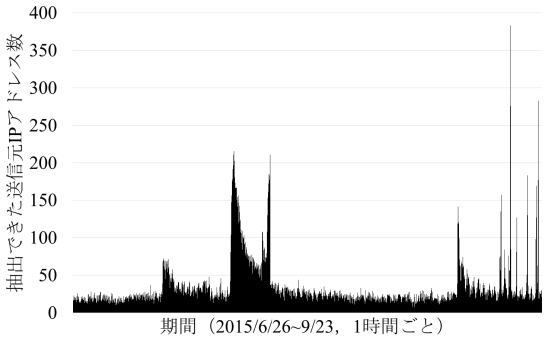
### A.1 (a) の場合に期間ごとに正解データと判断した送信元 IP アドレス



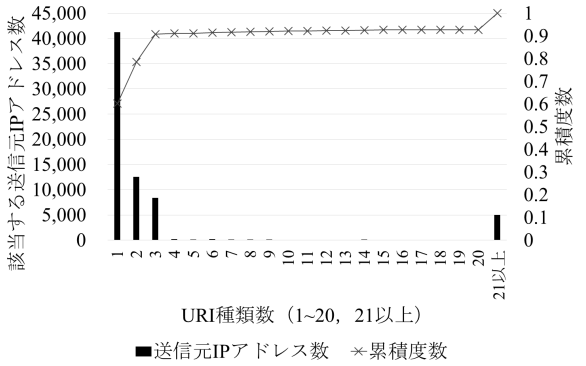

### A.2 (a) の場合に正解データと判断した送信元 IP アドレスの URI 種類数の分布



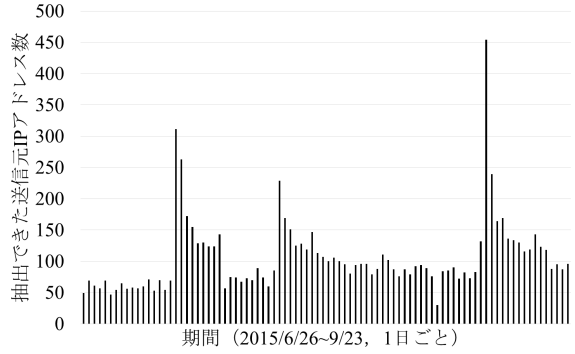

### A.3 (b) の場合に期間ごとに正解データと判断した送信元 IP アドレス



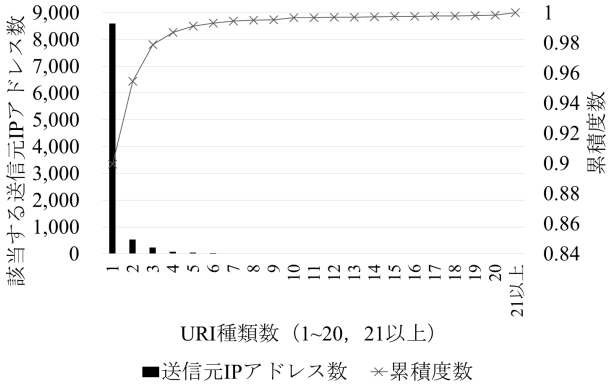

### A.4 (b) の場合に正解データと判断した送信元 IP アドレスの URI 種類数の分布



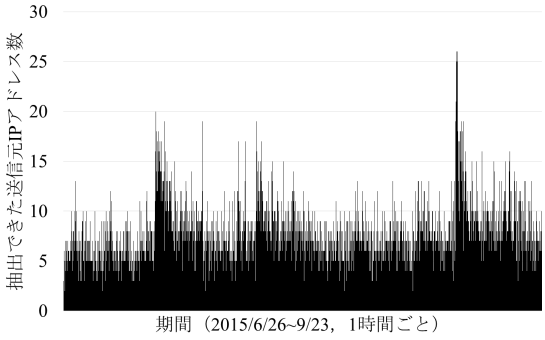

### A.5 (a) の場合に期間ごとに提案手法が抽出した送信元 IP アドレス



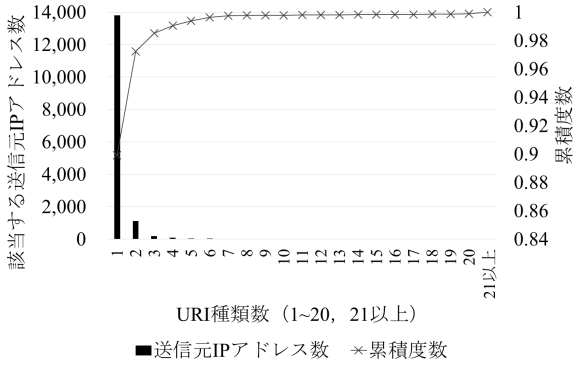

### A.6 (a) の場合に提案手法が抽出した送信元 IP アドレスの URI 種類数の分布



### A.7 (b) の場合に期間ごとに提案手法が抽出した送信元 IP アドレス




### A.8 (b) の場合に提案手法が抽出した送信元 IP アドレスの URI 種類数




### A.9 独自シグネチャの一覧

**Original signatures list**

| カテゴリ | 正規表現 |
|---|---|
| CMS | .*¥/wp-login.php.* |
| | .*wp-content.*¥.php |
| | .*wp-admin.*¥.php |
| | .*¥/xmlrpc.php |
| | .*¥/administrator¥/.*joomla.* |
| | .*¥/assetmanager.aspx |
| | .*soapCaller.bs |
| | .*open=1&arrs1¥[¥]=99&arrs1¥[¥]=102&arrs1¥[¥]=103&arrs1¥[¥]=95&arrs1¥[¥].* |
| FCKeditor | .*fckeditor.*¥/.+¥.html |
| | .*fckeditor.*¥/.+¥.php |
| | .*fckeditor.*¥/.+¥.aspx |
| | .*ckeditor.*¥/.+¥.php |
| | .*editor.*¥/.+¥.html |
| | .*editor.*¥/.+¥.php |
| | .*editor.*¥/.+¥.aspx |
| | .*¥/fckeditor¥/.*¥/$ |
| | .*¥/fckeditor.* |
| | .*¥/editor¥/.*¥/$ |
| | .*¥/editor.* |

| | |
|---|---|
| 不正中継ホストの探索 | ^CONNECT.+:25$ |
| | ^CONNECT.+:80$ |
| | ^CONNECT.+:443$ |
| | .+:http:¥/¥/.+ |
| | .+:https:¥/¥/.+ |
| IoT 機器の存在確認 | .*:¥/rom-0$ |
| | .*:¥/tmUnblock¥.cgi$ |
| | .*:¥/onvif¥/snapshot$ |
| | .*:¥/rtpd.cgi |
| Struts | .*¥/login¥.action$ |
| | .*¥/LoginPage¥.do$ |
| MongoDB | .*moadmin¥.php$ |
| SQL injection を含む | .*%20union%20select%20.* |
| | .*union¥/¥*¥*¥/select¥/¥*¥*¥/.*¥/¥*¥*¥/from¥/¥*¥*¥/.* |
| PHP-CGI 攻撃 | .*:¥/cgi-bin¥/php |
| 改ざんサイトの存在確認 | .*:¥/nyet.+$ |
| ShellShock 脆弱性 | .*:¥/Ringing¥.at¥.your¥.dorbell!$ |

**齊藤 聡美** （正会員）

2010 年横浜国立大学工学部電子情報工学科早期卒業．2012 年横浜国立大学大学院環境情報学府情報メディア環境学専攻情報メディア学コース博士課程前期修了．同年富士通株式会社および株式会社富士通研究所入社．現在，セキュリティ研究所サイバーセキュリティプロジェクトに所属し，ネットワークセキュリティ分析・検知技術の研究開発に従事．2015 年 4 月より横浜国立大学大学院環境情報学府博士課程後期在籍．2009 年度電気学会電気学術女性活動奨励賞，受賞．2014 年マルチメディア，分散，協調とモバイルシンポジウム（DICOMO2014）優秀プレゼンテーション賞・優秀論文賞受賞．

**吉岡 克成** （正会員）

2005 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期修了．博士（工学）．同年 4 月独立行政法人情報通信研究機構で研究員として勤務．2007 年 12 月より横浜国立大学学際プロジェクト研究センター特任教員（助教）．2011 年 4 月横浜国立大学大学院環境情報研究員准教授．マルウェア解析やネットワーク攻撃観測・検知等のネットワークセキュリティの研究に従事．2009 年文部科学大臣表彰・科学技術賞（研究部門），2016 年産学官連携功労者表彰総務大臣賞受賞．

**松本 勉** （正会員）

1986 年 3 月東京大学大学院工学系研究科電子工学専攻博士課程修了，工学博士．同年 4 月横浜国立大学講師．2001 年 4 月より同大学院環境情報研究院教授．先端科学高等研究院情報・物理セキュリティ研究ユニット代表を兼務．暗号アルゴリズム・プロトコル，ネットワーク/ソフトウェア/ハードウェアセキュリティ，バイオメトリクス，人工物メトリクス，計測セキュリティ，自動車セキュリティ等の「情報・物理セキュリティ」の研究教育に 1981 年より従事．1982 年にオープンな学術的暗号研究を目指した「明るい暗号研究会」を 4 名で創設．2005～2010 年国際暗号学会 IACR 理事．CRYPTREC 暗号技術検討会座長．日本学術会議連携会員．産業技術総合研究所研究顧問．第 32 回電子情報通信学会業績賞，第 5 回ドコモ・モバイル・サイエンス賞，第 4 回情報セキュリティ文化賞，2010 年文部科学大臣表彰・科学技術賞（研究部門）受賞．