アクセスパターンに基づく 攻撃対象 Web アプリケーション発見手法の提案

黒木 琴海[1,a)]　鐘本 楊[1]　青木 一史[1]　三好 潤[1]

概要：攻撃者は Web サーバに対して攻撃を行う前に，攻撃可能な Web サーバを探すスキャンを行うことがある．具体的には，脆弱な Web アプリケーションが存在するか否かを判断するための HTTP リクエストを対象の Web サーバへ送信する．攻撃者によるスキャン行為を早期に検知できれば，攻撃者が狙っている Web アプリケーションを特定して注意喚起を行うことで被害を抑えることができる．本稿では，HTTP アクセスログからスキャンの標的にされた URI を検出する手法を提案する．提案手法では，攻撃者がスキャンを行う際に同じ送信元から複数の送信先へ同じ HTTP リクエストを送信する傾向があるという点に基づいて送信元 IP アドレスのスコア付けを行う．送信元 IP アドレスのスコアを用いて URI のスコア付けを行い、当該スコアに基づいてスキャンに用いられた URI を検出する．実 Web サーバ群の HTTP アクセスログに対して提案手法を適用することで，既存手法に比べ少ない計算コストで高精度にスキャンを検出できることを示す．

キーワード：Web，脆弱性スキャン

1. はじめに

公開されている Web サーバを狙う攻撃者は，実際に攻撃を行う前に標的の Web サーバが攻撃可能か否かを調べることがある．標的の Web サーバに脆弱性のある Web アプリケーションが存在したり，アプリケーションの設定の不備などがある場合，その Web サーバは攻撃の標的となり得る．具体的には，脆弱な Web アプリケーションが持つ URI を標的の Web サーバへリクエストして，リクエストに対する Web サーバからのレスポンスによってその Web アプリケーションの有無が判断できる．例えば，Web 上でデータベースの管理を行えるアプリケーションである phpMyAdmin の脆弱性を利用して攻撃を行おうとした際，攻撃者はまず phpMyAdmin が標的の Web サーバに存在するかどうかを調査するために，図 1 に示すような URI に対して HTTP リクエストを送信する．

図 1 の例では，攻撃者は phpMyAdmin の設定を行うための setup.php の有無を確認している．phpMyAdmin に限らず一部の Web アプリケーションでは，バージョンによってディレクトリ構造やファイル名が異なるため，図 1 のように同じ Web アプリケーションを対象としていても異なる URI を用いて脆弱性のあるバージョンを網羅的に確認する場合がある．

```
/phpMyAdmin-2/scripts/setup.php
/phpmyadmin/scripts/setup.php
/pma/scripts/setup.php
/phpMyAdmin-3.4.3.1/scripts/setup.php
```

図 1　Web スキャンに用いられる URI の例

本稿では，攻撃者による標的の Web サーバへのこのような調査行為を Web スキャンと呼ぶ．攻撃者による Web スキャンを早期に検知できれば，攻撃者が攻撃に利用しようとしている脆弱な Web アプリケーション等を特定することができる．攻撃者が狙っている脆弱な Web アプリケーションの情報が得られれば，注意喚起などによって被害を抑えることができる．

そこで本稿では，Web スキャンに利用された URI の検出手法を提案する．Web スキャンに利用された URI が分かれば，URI 中に含まれているアプリケーション名やディレクトリ構造から，それがどの Web アプリケーションやネットワーク機器の URI なのか推測することができる．実在する Web サーバ群への HTTP アクセスログを用いた実験によって，提案手法が高精度にスキャンに利用された URI を検出できることが分かった．

本稿の構成は以下の通りである．2 節では，Web スキャ

[1] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
[a)] kuroki.kotomi@lab.ntt.co.jp

ン検出における既存手法とその課題について述べる，3 節では，提案手法の詳細について述べる．4 節では，実在する Web サーバ群に対する HTTP アクセスログを用いた実験と評価について述べる．5 節では，関連研究について述べる．最後に 6 節でまとめとする．

2. Web スキャン検出における課題

Web スキャンを検出する既存の手法として，以下のものが挙げられる．それぞれの手法の概要と課題について述べる．

シグネチャマッチを用いた手法 Web Application Firewall (WAF) や Intrusion Detection System (IDS) の多くでは，シグネチャによって攻撃や Web スキャンの検出が行われている．シグネチャの作成には，人の目による判断やハニーポット [1–3] が用いられる．既に攻撃や Web スキャンであると判定されてシグネチャが登録されている場合には，そのシグネチャにマッチする Web スキャンを検出することができる．しかしながら，事前に Web スキャンであると判定してシグネチャを作成する必要があるため，新たに発生した Web スキャンには対応できず，網羅性にも問題がある．

Web ページの遷移に基づいた手法 Web サーバに対して攻撃を行う攻撃者は，Web ページの遷移の順序や間隔が正常なアクセスを行うユーザとは大きく異なる場合がある．正常なユーザの挙動を学習して，それと異なるアクセスを検出することで，攻撃等を検出する手法 [4–6] が存在する．攻撃者は，Web サイトのページの構造に沿って遷移することは少ないため，ページ遷移の順序が正常なアクセスと異なる傾向がある場合や，ページ遷移の間隔が極端に短い場合などに攻撃として検出が可能である．Web スキャンの場合でも同様の傾向を示すと考えられるため，これらの手法で Web スキャンを検出することも可能である．しかしながら，これらの手法では攻撃者に正常なアクセスを偽装される恐れがある．例えば，Web スキャンの中にダミーとなる正常なアクセスに似たリクエストを織り交ぜたり，リクエストの間隔を長くすることで，検出の回避が可能となってしまう．

URI の共起性に基づいた手法 Web スキャンにおいては，効率的に攻撃可能な Web サーバを探すために，攻撃者は一つの IP アドレスから複数の異なる Web サーバに対して同じ URI をリクエストする場合がある．この特徴を利用して，HTTP リクエストに含まれる URI の共起性に基づいて Web スキャンを検出する手法 [7] が提案されている．同じ Web アプリケーションと思われる URI をクラスタリングした上で URI の共起性に基づいて Web スキャンの検出を行っており，検出した URI について高い適合率を達成している．しかしながら，共起性のみに基づいて検出しているため，正常な HTTP リクエストであっても複数の Web サーバに同じ URI をリクエストするような場合に誤検出してしまう可能性がある．例えば，“/” や “/index.php” などはどの Web サーバでもリクエストされる場合が多く，正常なユーザが複数の Web サーバに対してリクエストする場合も多いと考えられる．また，クラスタリングによって Web スキャンに使われる URI と正常なリクエストに使われる URI が同じクラスタに分類されることによって，正常なリクエストに紛れて Web スキャンに使われた URI が検出できない可能性も考えられる．

3. 提案手法

本節では，2 節で述べた課題を踏まえて，HTTP アクセスログを元に Web スキャンに利用された URI を検出する手法を提案する．提案手法では，URI の共起性とステータスコードといった攻撃者のアクセスパターンを利用して，Web スキャンに利用された URI の検出を行う．提案手法は，1)URI の正規化，2) 送信者の評価，3) 正規化済み URI の評価，の 3 つの処理から構成されている．

3.1 URI の正規化

Web スキャンにおいては，同じ Web アプリケーションを対象としていても複数の異なる URI が使われていることがある．Web アプリケーションのバージョンの違いなどによってディレクトリ構造が変わることがあるため，それぞれに対応できる URI を利用して網羅的にスキャンが行われる場合がある．URI の正規化を行なって同じ Web アプリケーションを対象とした URI をまとめることで，Web スキャンに使われている URI をスコアをより際立たせることができる．具体的には，以下の 3 種類の正規化を行う．

(1) **バージョン等を表す数字を削除**
アプリケーションのバージョン等を表す数字の違いを統一するために，連続する数字，数字の前に付く “-”，数字の間に含まれる “.” をまとめて削除する．例えば，“xxx-2.1”，“xxx3.10”，“xxx2” はいずれも “xxx” に正規化される．

(2) **全て小文字に統一**
アルファベットの大文字と小文字の表記による違いを統一するために，全て小文字に変換する．

(3) **パス部の抽出**
“?” 以降のクエリ部が違っても，その前のパス部が同じであれば同じアプリケーションである可能性が高いため，“?” 以降の文字列は削除してパス部のみを抽出する．

URI の正規化の例を表 1 に示す．表 1 内の URI はいずれも異なる URI ではあるが，全て phpMyAdmin が有する

表 1　URI 正規化の例

URI	正規化済み URI
/phpMyAdmin-2/scripts/setup.php /phpMyAdmin-3.4.3.1/scripts/setup.php /phpmyadmin/scripts/setup.php?param=xx /phpmyadmin2/scripts/setup.php	/phpmyadmin/scripts/setup.php

setup.php を対象とした URI である．前述の通り正規化を行うことで，全て “/phpmyadmin/scripts/setup.php” となり，同じ URI として扱うことができるようになる．

3.2　送信者の評価

送信元 IP アドレス毎に，Web スキャンを行う攻撃者らしさを評価する．攻撃者らしさを判定する基準として，以下の 2 点の特性を利用する．

(1) **同じ URI を複数のホストに対してリクエストしている**
攻撃者は，効率的に攻撃可能な Web サーバを探すために，同じ IP アドレスから複数の Web サーバに対して同じ URI をリクエストする傾向があると考えられる．

(2) **HTTP リクエストのレスポンスが 200 番台でない**
Web サーバ上に存在するページに対して正常なリクエストを行った場合は HTTP リクエストのステータスコードは 200 番台となる．一方で，Web スキャンでは標的の Web サーバ上に存在しないページがリクエストされる場合が多いため，HTTP リクエストのステータスコードが 404 などの 200 番台以外となることが多い．

以上の 2 点を踏まえて，送信者のスコア P_{src} を式 (1) を用いて算出する．

$$P_{\mathrm{src}} = \frac{|S_{\mathrm{src}} \cap (S_{\mathrm{error}} \cap S_{\mathrm{multi}})|}{|S_{\mathrm{src}}| + W} \tag{1}$$

S_{src} は当該送信者から送信された全てのリクエストの集合，S_{error} はステータスコードが 200 番台以外のリクエストの集合，S_{multi} は送信元 IP アドレスとリクエスト URI が同じで宛先ホストが異なるリクエストの集合，W はユーザが事前に設定する定数であり，分母に加算する重みを表している．当該送信者が送信したリクエストのうち，同じ URI を異なるホストへリクエストしており，かつ正常なレスポンスが返っていないリクエストの数が多いほどスコアが高くなる．W によって，リクエスト数が少なく Web スキャンとは断定できない送信者のスコアが極端に高くなることを防ぐ．

3.3　正規化済み URI の評価

3.2 節で算出した送信者のスコアを元に，各正規化済み URI のスコアを求める．正規化済み URI のスコアは，正規化前の URI をリクエストした送信者のスコアの最大値とする．最大値を用いることで，1 人でも当該 URI を用いて Web スキャンを行なっている送信者がいた場合にスコアが高くなり，Web スキャンとして検出が可能となる．

ここで得られた各正規化済み URI のスコアに対して閾値 T と比較を行い，T よりもスコアが高い URI を，Web スキャンに用いられた URI として検出する．T は 0 から 1 の任意の値で，事前にユーザが設定する．

4.　実験および評価

実在する Web サーバ群に対する HTTP アクセスのログを取得して，Web スキャンに用いられた URI の検出を行った．解析対象の HTTP アクセスログの収集期間は 2017 年 9 月 20 日から 2018 年 6 月 26 日までのおよそ 9 ヶ月間，HTTP リクエスト数は 20,687,905 リクエスト，送信元 IP アドレス数は 227,668 IP アドレス，送信先 IP アドレスは 124 IP アドレスである．

図 2 に，取得した HTTP アクセスログに対して提案手法を適用した結果のスコアに対する正規化済み URI のヒストグラムを示す．スコアは 0 から 1 の範囲で，高いほど Web スキャンに用いられた可能性が高いと考えられる．得られた正規化済み URI 数は全部で 44,226 URI であった．図 2 から，スコアが高いものと低いもので二分されていることが分かる．スコアが (0.5, 0.6] の正規化済み URI 数が最も少なくなっており，スコアが概ね 0.6 以上の正規化済み URI と 0.6 未満の正規化済み URI でグループ化されており，スコアが 0.6 以上の URI が特に Web スキャンに用いられた可能性が高いと考えられる．また，スコアが [0, 0.1] の正規化済み URI 数が 32,263 となっており，スキャンの可能性が低い正常なアクセスに用いられる URI が全体の約 73%を占めていることが分かる．

4.1　検出精度の評価

提案手法の検出精度を評価するために，攻撃やフィンガープリントに使われると知られている URI を正しく検出できた割合を示す検出率を算出した．検出率の算出には，公開されている攻撃情報サイトやデータベース等に攻撃やフィンガープリントに使われるとして登録されている URI の中から，解析対象の HTTP アクセスログに含まれている URI を抽出して用いた．具体的には，攻撃コードがデータベース化されている Exploit-DB[*1]，Web アプ

[*1] https://www.exploit-db.com/

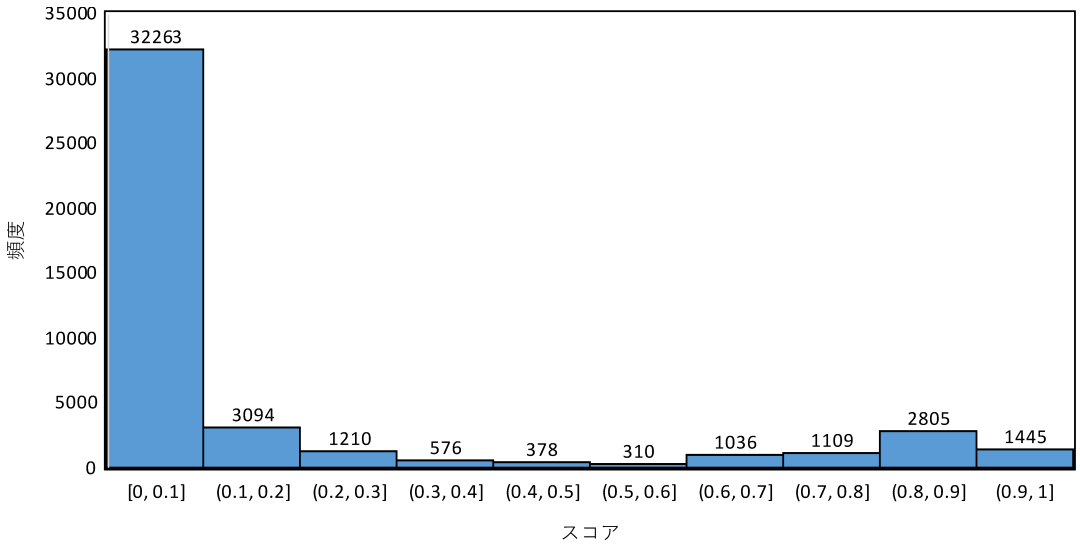


図 2　スコアに対する正規化済み URI の分布

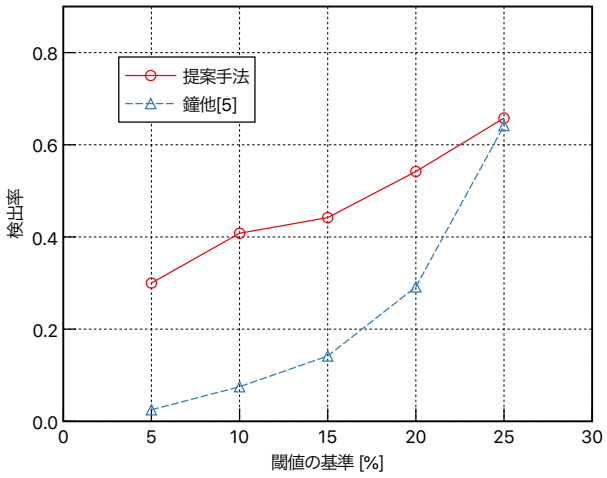


図 3　閾値毎の検出率

リケーションのフィンガープリンティングツールである BlindElephant[*2]に含まれている URI を用いた．また，比較対象として，鐘らの URI の共起性に基づいた手法 [7] を用いた．

図 3 にそれぞれの手法の検出率を示す．両手法とも，スコアが事前に設定した閾値を超える URI を Web スキャンとして検出するが，評価において閾値を公平に設定するため，それぞれの手法で算出したスコアの上位から 5%刻みで 25%までそれぞれ抽出した URI を，スキャンとして検出した URI とした．図 3 から，いずれの場合でも提案手法の方が高い検出率を達成していることが分かる．これは，鐘らの手法では Web スキャンに用いられた URI がその他の正常なリクエストによって低いスコアとなってしまって検出できていない場合があるのに対して，提案手法では一度でも Web スキャンに用いられていればスコアが高くなるように設計しているためである．

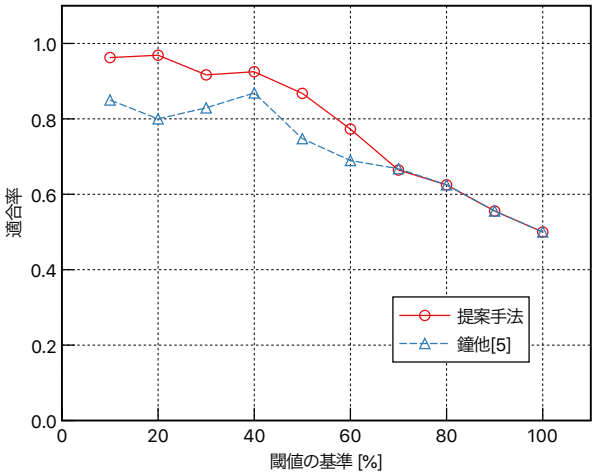
[*2] http://blindelephant.sourceforge.net/

4.2　既知のデータセットを用いた検出精度の評価

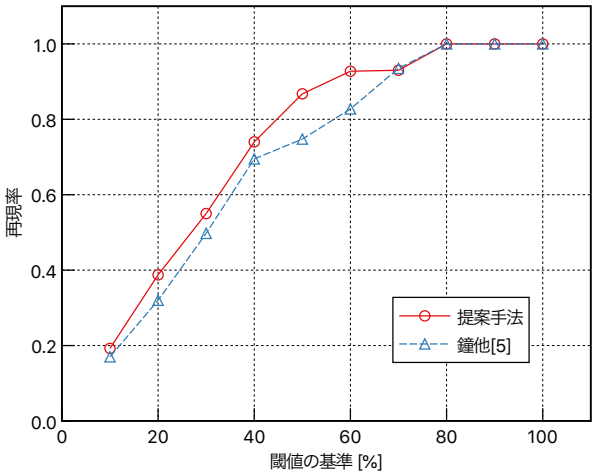
4.1 節では，取得した全ての HTTP アクセスログに対して検出を行なった結果を用いて評価を行なった．しかしながら，検出した URI が実際に Web スキャンに用いられたものであるかの判定を人力で行うのは困難であるため，正確な適合率や再現率を算出して評価することができない．そのため本節では，既に Web スキャンに用いられていると判明している URI と，Web スキャンに用いられていないと思われる URI を同数抽出して，それらの URI が含まれている HTTP リクエストのみを抽出したデータセットを用意して適合率と再現率を算出した．正解データセットは，4.1 節での評価に用いた URI を元に 400 種類の URI を抽出して作成した．不正解データセットは，取得した HTTP アクセスログに含まれる URI の一覧から，正解データセットに含まれない 400 種類の URI をランダムに抽出して作成した．4.1 節での評価と同様に，鐘らの URI の共起性に基づいた手法 [7] との比較を行う．

図 4 に作成したデータセットを用いた評価結果を示す．図 4(a)，図 4(b)，図 4(c) はそれぞれの手法で算出したスコアの上位から 10%刻みで抽出した URI の適合率と再現率，またそれを元に算出した F 値である．図 4(a)，図 4(b)，図 4(c) から，以下の 2 つのことが分かる．

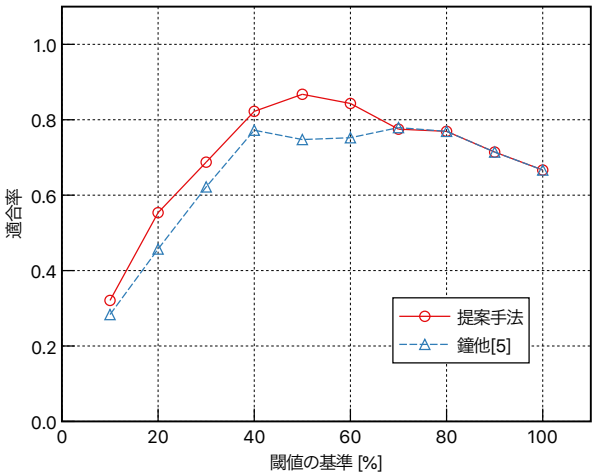
1 つ目は，スコアの上位 10%から 60%までの範囲において提案手法の性能が高くなっていることである．例えば，スコアの上位 20%をスキャンの URI として抽出した場合に提案手法の適合率が約 0.969，鐘らの手法の適合率が約 0.800 となっており，鐘らの手法と比較して提案手法が約 1.21 倍の適合率を達成している．再現率においても，提案手法が 0.388，鐘らの手法が 0.320 となっており，同じく提案手法が約 1.21 倍の再現率を達成している．また，それぞれの手法において F 値が最大となる場合を比較すると，スコアの上位 50%までを抽出した場合に提案手法の F 値



(a) 閾値毎の適合率



(b) 閾値毎の再現率



(c) 閾値毎の F 値

図 4 データセットによる評価結果

が約 0.868，スコアの上位 70%までを抽出した場合に鐘らの手法の F 値が約 0.779 であった．このことから，それぞれの手法で最適な閾値を設定した場合においても，提案手法の方が約 1.11 倍の性能を達成できていることが分かる．

2 つ目は，スコアの上位 70%から 100%までの範囲において 2 つの手法がほぼ同じ性能となっていることである．これは，両手法において Web スキャンに用いられた URI のほとんどがスコアの上位 70%までに含まれていることから差が出なかったものと考えられる．

4.3 新たなスキャンの検出

脆弱性があるとは知られていないが攻撃の対象となっている Web アプリケーションを発見できることを確認するために，HTTP アクセスログの収集期間以降に脆弱性が発見された Web アプリケーションに関して，脆弱性が発見されるより前の HTTP アクセスログから検出が可能か実証した．提案手法によって検出した URI から，外部サイトやデータベースから得られた攻撃やスキャンに用いられると報告されている URI を除いて残った URI に対して調査を行った．提案手法において閾値 T は 0.5 とした．

発見できた例として，オープンソースの CMS である Drupal[*3]が挙げられる． Drupal においてリモートから任意のコードが実行可能となる脆弱性 (CVE-2018-7600) が 2018 年 3 月 28 日に公開されている．提案手法によって “`/drupal/CHANGELOG.txt`” という URI が Web スキャンとして検出できた．実験に用いた HTTP アクセスログ内では，“`/drupal/CHANGELOG.txt`” は 2017 年 10 月 11 日に一度 Web スキャンに利用されており，その後は公開後の 2018 年 5 月に再びアクセスされている．また，脆弱性の公開より 5 ヶ月以上前のログから検出できることが確認できた．また，2018 年 4 月 12 日に PoC が公開されて以降の HTTP アクセスログでは PoC に含まれる “`/user/register`” に対する Web スキャンが活発となっていることも確認できた．

5. 関連研究

本節では，Web スキャンに関する既存の研究について述べる．本研究で行なっているような，Web スキャンに着目して用いられた URI の検出を行う研究は，2 節にて触れた文献 [7] で行われていて，検出した Web スキャンの分類や傾向などが分析されている．共起性を際立たせるために URI のクラスタリングを行なっているという特徴がある．また，文献 [8] でも同様に URI の共起性に基づいて検知を行なっているが，こちらは悪意のあるリクエストを検出することを目的としている．文献 [9] では，SVN を用いて攻撃者による悪意のある Web セッションを攻撃セッションと脆弱性スキャン（Web スキャン）セッションに分類する研究が行われている．実際に攻撃を行っている段階の Web アクセスと比較した場合の Web スキャンの特徴が分析されている．文献 [4] では，URI をパス部とクエリ部

[*3] `https://www.drupal.org`

に分けてモデル化と学習を行い，異常なアクセスを検出する手法が提案されている．文献 [5] では，ページ遷移の順序に着目して異常なセッションを検出する手法が提案されている．文献 [6] では，Web サービスに対する DoS 攻撃を検出するためにページのアクセスパターンを利用する手法が提案されている．ページのブラウジングの順序やブラウジング時間，ページの情報サイズ等に着目して検出を行う．文献 [10] では，悪意のある Web クローラを識別するために，Web アクセスログの分析に 2 種類のニューラルネットワークによる教師なし学習を用いている．

本研究では，文献 [7] と同様に URI の共起性を利用して Web スキャンに用いられた URI の検出を行なっている．しかしながら，URI の正規化やステータスコードの活用，手順の追加等を行なうことで検出精度の向上を実現した．

6. おわりに

本稿では，アクセスパターンに基づいて Web スキャンに利用された URI を検出する手法を提案した．実在する Web サーバ群に対する HTTP アクセスのログを用いた実験によって，提案手法が既存手法に比べ少ない計算コストで高精度にスキャンを検出できることが分かった．

参考文献

[1] Portokalidis, G., Slowinska, A. and Bos, H.: Argos: An Emulator for Fingerprinting Zero-day Attacks for Advertised Honeypots with Automatic Signature Generation, *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 (EuroSys '06)*, pp. 15–27 (2006).

[2] Anagnostakis, K. G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E. P. and Keromytis, A. D.: Detecting Targeted Attacks Using Shadow Honeypots, *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14 (SSYM '05)*, pp. 9–9 (2005).

[3] Kreibich, C. and Crowcroft, J.: Honeycomb: Creating Intrusion Detection Signatures Using Honeypots, *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 1, pp. 51–56 (2004).

[4] 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).

[5] Cho, S. and Cha, S.: SAD: web session anomaly detection based on parameter estimation, *Computers & Security*, Vol. 23, No. 4, pp. 312–319 (2004).

[6] Yatagai, T., Isohara, T. and Sasase, I.: Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior, *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim '07)*, pp. 232–235 (2007).

[7] 鐘　揚，折原慎吾，谷川真樹，嶋田　創，村瀬　勉，高倉弘喜，大嶋嘉人：URI の共起性検知に基づく Web スキャンの実態調査，電子情報通信学会技術研究報告，Vol. 115, No. 488, pp. 25–30 (2016).

[8] 齊藤聡美，吉岡克成，松本　勉：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，情報処理学会論文誌，Vol. 59, No. 2, pp. 574–590 (2018).

[9] Goseva-Popstojanova, K., Anastasovski, G. and Pantev, R.: Classification of Malicious Web Sessions, *2012 21st International Conference on Computer Communications and Networks (ICCCN '12)*, pp. 1–9 (2012).

[10] Stevanovic, D., 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).