広域スキャンで収集した応答を用いた全ポート待受型Webハニーポット

加藤 誠也[1]　森下 瞬[1]　田辺 瑠偉[2]　吉岡 克成[3]　松本 勉[3]

概要：インターネット上で起きている攻撃を観測することを目的とした，脆弱なネットワークサービスを模擬するハニーポットが広く運用されている．しかし，Web アプリケーションや Web サービスの多様化に伴い，標的となっている機器に特有のポートやサービスを動作させないと攻撃を観測できない事例が出てきている．本研究では，広域スキャンにより収集した応答を用いて Web ハニーポットの観測能力を向上させる方法を提案する．具体的には，ハニーポットを全 TCP ポートで待受状態とし，これらのポートへのアクセスを観測する．その後，アクセスのあった宛先ポートや HTTP 通信のアクセス先のパス情報を元に広域スキャンを行い，攻撃対象となっている機器の応答の収集を試みる．そして，ハニーポットに対して同様の攻撃が観測された際に，収集した応答を用いて攻撃対象を模擬する Web ハニーポットを提案する．評価実験では，提案手法を用いて実現したハニーポットと既存のハニーポットをそれぞれ単一の IP アドレスを用いてインターネット上に 28 日間公開したところ，提案手法では HTTP リクエスト数が約 28,000 件増加し，8088/tcp では，既存のハニーポットでは観測できなかった攻撃を観測することができた．また，既存のハニーポットと比べて，およそ 7 倍に当たる 228 件の検体を取得することができた．

キーワード：Web ハニーポット，ネットワーク攻撃観測

Web Honeypot with Full Ports Open Using Responses Collected by Network Scan

Kato Seiya[1]　Morishita Shun[1]　Tanabe Rui[2]　Yoshioka Katsunari[3]　Matsumoto Tsutomu[3]

Abstract: Honeypots that simulate vulnerable network services are widely used to observe attacks on the Internet. However, with the diversification of Web applications and Web services running on various connected devices, it is becoming increasingly difficult to observe attacks on specific ports and services on specific target devices. In this study, we propose a method to improve the observability of Web honeypot using responses collected by network scan. Namely, we first prepare honeypots that listen on all TCP ports to observe various incoming requests on these ports. Then, we scan the Internet with sanitized version of these requests to collect corresponding responses from potential target devices. Finally, we deploy the collected responses to the honeypot to emulate the target devices. In the evaluation experiment, we assigned a single IP address to each honeypot and observed the attacks for 28 days. The number of observed HTTP requests increased by about 28,000 and particularly, a new attack on 8088/tcp was observed by the proposed honeypot. Moreover, 228 malware binaries were captured by the proposed honeypot, which is approximately seven times as many as that of the existing honeypot.

Keywords: Web Honeypot, Observation of network-attacks

[1] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University
[2] 横浜国立大学先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University
[3] 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,

1. はじめに

IT 技術の発展により，インターネットに接続される機器の数が爆発的に増加した．しかし，その一方で，様々な機器やソフトウェアを対象としたサイバー攻撃が増加している．このため，脆弱なネットワークサービスやソフトウェアを模擬するハニーポットの研究開発が進んでいる．また，ハニーポットをインターネット上に設置することで，攻撃者が狙っているサービスや脆弱性の分析，マルウェア検体の収集が広く行われている．

近年，インターネット上に Web コンソールを持った機器やソフトウェア，Web コンテンツを公開している IoT 機器が増加しており，HTTP を用いた攻撃が今後も発生すると考えられる．一般に，ハニーポットの種類はプロトコルごとに大別されており，これらの攻撃の観測には HTTP 通信を模擬する Web ハニーポットが有効である．実際に，オープンソースの Web ハニーポットが数多く存在する．例えば，Dionaea[1] や Glastopf[2] のようにエミュレータを用いて Web サーバを模擬するハニーポットがある．また，phpmyadmin_honeypot[3] や Wordpot[4] のような特定の Web アプリケーションの脆弱性を模擬するハニーポットが存在する．これらのハニーポットでは，観測者が攻撃対象となるサービスや攻撃を観測できるポート番号を事前に把握しておく必要がある．しかし，昨今では，機器やソフトウェアが利用するポートが複雑化しており，どのポートにどのような攻撃が発生するのか把握することが困難になっている．NICTER のダークネット観測結果 [5] によると，通常では使用されないポートへの通信が多数観測されており，攻撃対象となる機器やソフトウェアが使用するポートが多様化していることがわかる．また，エミュレータにより攻撃対象の動作を模擬するハニーポットは一定の応答しか返さないため，実際の機器やサービスとは異なる応答を返す可能性がある．

本研究では，広域スキャンを使用して実ホスト (以下，ホストとする) から応答を取得することで，そのポートやパスで動作する機器やソフトウェアを模擬する Web ハニーポットを提案する．具体的には，ハニーポットを全 TCP ポートで待受状態とし，これらのポートへのアクセスを観測する．その後，アクセスのあった宛先ポートや HTTP 通信のアクセス先のパス情報を元に広域スキャンを行い，攻撃対象と思われる機器の応答を収集する．そして，ハニーポットに対して同様の攻撃が観測された際に，収集した応答を用いて攻撃対象を模擬する Web ハニーポットを提案する．また，HTTP リクエスト中の攻撃コードから URL とダウンロードコマンドを抽出して，攻撃元からファイル (以降では，検体と呼ぶこととする) のダウンロードを行う．

Yokohama National University / Institute of Advanced Sciences, Yokohama National University

評価実験では，提案手法を実現したシステムを 2019 年 7 月 22 日から 2019 年 8 月 18 日までインターネット上に設置し，観測した攻撃と検体の分析を行った．また，同期間に既存の Web ハニーポットをインターネット上に設置し，観測した攻撃と検体の比較を行った．その結果，提案システムでは既存の Web ハニーポットよりも多くの HTTP リクエストを観測する事に成功した．特に，8088/tcp では既存の Web ハニーポットよりも HTTP リクエストを 28,000 件以上多く観測することができた．また，同ポートで応答を変更した際には段階的な攻撃を観測することができ，検体のダウンロードに成功した．その結果，提案システムでは既存の Web ハニーポットの約 7 倍に当たる 228 件の検体を取得することができた．

以降では，2 章で関連研究について述べ，3 章で広域スキャンで収集した応答を用いた全ポート待受型 Web ハニーポットを提案する．そして，4 章で Web ハニーポットの観測結果を示すとともに考察を行い，5 章でまとめと今後の課題を述べる．

2. 関連研究

ハニーポットはその実装方法から高対話型と低対話型に分類することができる．高対話型とは，実際に脆弱性をもった機器やソフトウェアを使用することで攻撃の観測を行う方式である．一方，低対話型とは，エミュレータやスクリプトを用いて攻撃対象の動作を模擬することで攻撃の観測を行う方式である．そのため，高対話型では，認識していない他の脆弱性によって悪用された場合のリスクが高いが，低対話型と比べて観測能力は高い．

高対話型の Web ハニーポットでは文献 [6] や Honnypotter[7] のような，ある特定の Web アプリケーションに対する攻撃を観測するものが多い．また，bwpot[8] のような，複数の Web アプリケーションを Docker [9] を用いて構築し，攻撃の観測を行う高対話型 Web ハニーポットも存在するが，同時に構築できる Web アプリケーションの数や観測できる攻撃は限られている．

低対話型の Web ハニーポットでは phpmyadmin_honeypot[3] や Wordpot[4] などの特定ソフトウェアを模擬するものや，Dionaea[1] や Glastopf[2] などの Web サーバの動作を模擬するハニーポットが存在する．高対話型と比べて低コストで安全に運用することが可能だが，ハニーポットが模擬する動作は一部のみであるため，実際の攻撃対象と動作が異なり，攻撃者にハニーポットとして検知されてしまう可能性がある．文献 [10] では，オープンソースのハニーポットのシグネチャを作成し，これらがハニーポット として検知可能であることを示している．また，Web アプリケーションを狙った攻撃を幅広く観測する低対話型の Web ハニーポットとして WOWhoneypot[11] が存在するが，全ての HTTP リクエストに対してステー

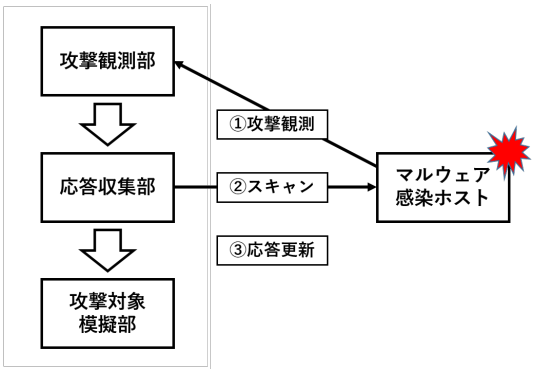

図 1　提案手法の基本アイディア
Fig. 1　Basic idea of the proposed approach.

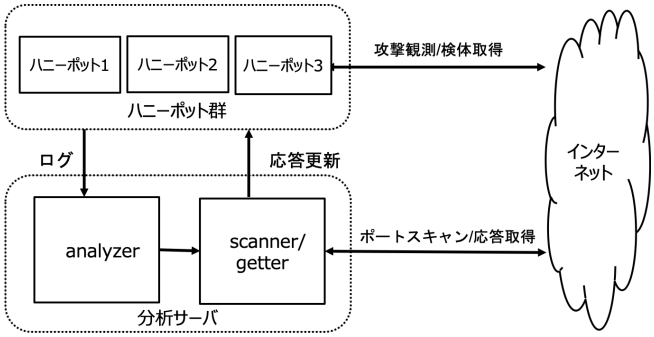

図 2　提案システムの構成
Fig. 2　Configuration of the proposed system.

タスコード 200 を返すため，攻撃者に攻撃対象の機器やソフトウェアではないと判断される可能性がある．

低対話型のハニーポットの課題を解決するため，ホストの応答を模擬するハニーポットが提案されている．文献 [12] では IoT 機器を対象とした攻撃を観測するため，インターネット上に存在する IoT 機器応答を学習し模擬するハニーポットを構築している．しかし，応答の模擬を行う対象が一部のポートの IoT 機器に限られており，観測できる攻撃は限定されている．また，全ポートで攻撃を観測するハニーポットが提案されている．文献 [13] では，全ての TCP ポートで 3 ウェイハンドシェイクを行う handshaker を構築し，サイバー攻撃の観測を行なっている．しかし，その後の応答は設定されておらず，サーバの応答内容によって条件分岐を行う攻撃は観測できないと考えられる．

本研究では，ホストの応答を模擬するハニーポットを構築し，全てのポートで攻撃の観測を行うことでより多様な攻撃の観測を行う．また，IoT 機器によらない，Web アプリケーションを対象とした応答の模擬を行う．

3. 広域スキャンで収集した応答を用いた全ポート待受型 Web ハニーポット

本章では，広域スキャンにより収集した応答を用いて，そのポートやパスで動作する機器やサービスを模擬する Web ハニーポットを提案する．以降では，3.1 節で提案手法の基本アイディアを説明し，3.2 節で提案手法の実現形態の一つである，広域スキャンで収集した応答を用いた全ポート待受型 Web ハニーポットシステムについて説明する．

3.1 基本アイディア

マルウェアの感染経路の一つに，ネットワークサービスの脆弱性を突いてマルウェア感染を引き起こすリモートエクスプロイト攻撃がある．攻撃者は，感染を拡大するために広範囲の IP アドレスをスキャンする場合が多い．また，スキャンの宛先となる IP アドレスは無作為に選ばれる場合が多い．このため，ハニーポットなどを用いて攻撃を観測することができる．しかし，機器やソフトウェアが利用するポートの複雑化に伴い，どのポートにどのような攻撃が発生するのか把握することが困難になっており，ハニーポットの観測能力の向上が求められている．そこで，広域スキャンの結果を用いてハニーポットの応答を更新することで，インターネット上に存在する様々な機器の動作を模擬する方法を提案する．**図 1** に提案システムの基本アイディアを示す．提案手法は，(1) 全 TCP ポートで攻撃の観測を行う**攻撃観測部**，(2) アクセスのあったネットワークに対してスキャンを行うことで応答を収集する**応答収集部**，(3) ホストの応答を用いて攻撃対象を模擬する**攻撃対象模擬部**から構成され，Web ハニーポットの観測能力を向上させることを目的とする．以降では，提案手法の構成要素について説明する．

3.1.1 攻撃観測部

インターネット上の攻撃の実態を把握するためには，実際にその攻撃を観測する必要がある．観測した攻撃を分析することでどのような機器やソフトウェアが攻撃の対象となっているのか，どのような脆弱性が悪用されているのかを把握することができる．攻撃者は，インターネット上に存在する攻撃対象を発見する際や感染の拡大を行うために，ネットワークスキャンやコネクション確立時の応答を見てホストの判別を行なっている場合が多い．近年，機器やソフトウェアが利用するポートが多様になってきているため，攻撃者は攻撃対象が用いるポートに対して通信を行っていると考えられるが，どのポートにどのような攻撃が発生するのかを事前に把握することは困難である．よって，攻撃観測部では全ポート待受型ハニーポットを用いてインターネット上で発生している多様な攻撃の観測を行う．

3.1.2 応答収集部

ハニーポットの技術が広く使用されるようになった現在，攻撃者は対象のホストがハニーポットか否か，調査を行う可能性がある．つまり，ハニーポットの挙動が攻撃対象と異なる場合には攻撃を行わない可能性がある．このため，攻撃対象を正しく模擬する必要がある．インターネット上の通信は，一方からの要求に対する他方からの応答で成り

立っており，攻撃観測部で観測した攻撃情報を元にネットワークスキャンを行い，ホストの応答を収集する方法や，Web ブラウザを用いて応答を収集する方法などがある．

3.1.3 攻撃対象模擬部

応答収集部で取得した応答を，攻撃観測部で観測した攻撃の応答として用いる．このように，攻撃対象の応答をハニーポットの応答として用いることで，攻撃対象の動作を模擬することが可能である．

3.2 広域スキャンで収集した応答を用いた全ポート待受型 Web ハニーポットシステム

本節では，提案手法の実現形態の一つである，広域スキャンで収集した応答を用いた全ポート待受型 Web ハニーポットシステムについて説明する（以降では，提案システムと呼ぶこととする）．**図 2** に提案システムの構成を示す．4 章の実験では，実験を効率的に行う目的から攻撃観測部と攻撃対象模擬部を同一のマシン上で実装した．このため，提案システムはインターネット上で起きている攻撃を観測するハニーポット群と，観測できた攻撃を分析してアクセス元ネットワークに対してスキャンを行い，ハニーポットの応答を更新する分析サーバから構成される．以降では，各構成要素について説明する．

3.2.1 ハニーポット群

ハニーポット群は，インターネット上に設置された複数の Web ハニーポットから構成され，ある IP アドレスで観測された全通信を iptables を用いてハニーポットに転送することで全ポートでの攻撃観測を行う．各ハニーポットは python3 を用いて HTTP 通信を行う簡易的な Web サーバであり，デフォルトでステータスコード 200 と文字列「It Works!」を応答する仕組みとなっている．また，HTTP リクエストに wget/curl コマンドと URL が記述されていた場合，攻撃元から検体をダウンロードする機能を実装している．加えて，一日ごとに分析サーバに観測ログを転送するとともに，分析サーバの結果を Web ハニーポットの応答に反映する機能を実装している．

各 Web ハニーポットの応答は，Session_table を用いて管理される．**表 1** に Session_table の例を示す．Session_table は，ハニーポットで攻撃を観測したときの待受ポート，ホストより取得したレスポンスヘッダ（HTTP メソッド，パス，ステータスコード）とレスポンスボディからなるハニーポットの応答とその応答を返した後にアクセス元ホストの動作により決定される評価値（value）から構成される．ここで，ハニーポットが応答を返した後，アクセス元ホストから再度アクセスがあった場合，アクセス元ホストに何らかのアクションを発生させる応答であると判断して，value に高い値を設定することとする．具体的には，Session_table に保存されているポートとパス (以降，アクセス URL と呼ぶことにする．) へのアクセスが発生した場合，保存されている応答をランダムで返す．そして，応答を返した後に同一アクセス元ホストからアクセスが発生するか否かを確認し，発生した場合は 2 を，発生しない場合は 1 を設定する．なお，value が 0 である応答はまだ一度も使用していない未評価の応答であり，優先的に応答として返されるように設定している．応答の設定方法についてはさらなる検討が必要であるが，4 章の実験では，次の設定で応答を行った．

(1) value=0，あるいは，評価値が高いものをランダムな確率で返す
(2) 評価値が低いものは今後応答には使用しない

表 1 Session_table
Table 1 Session_table.

Port	Method	Path	Header (md5_hash)	Status_code	Body (md5_hash)	Value
8888	GET	/hoge	7381vdh7...	200	281bdywd1...	0
〃	POST	/fuga	6132ens7...	404	173hwdd1...	1
9000	GET	/	2353cfw1...	403	938kavh1...	0

3.2.2 分析サーバ

分析サーバは，ハニーポット群のログ分析を行う **analyzer** と，インターネット上へのスキャンを行う **scanner**，ホストの応答を取得する **getter** から構成される．

- **analyzer**
  analyzer では一日に一度，ハニーポット群からログと検体を収集する．そして，一日分のログからアクセス URL が同じ HTTP リクエストを抽出し，送信元 IP アドレス数を算出する．このうち，ある程度攻撃の規模が大きいと考えられる HTTP リクエスト，つまり，送信元 IP アドレス数が一定数以上であった HTTP リクエストを scanner に送信する．また，収集した検体は VirusTotal [21] を用いて解析し，結果を保存する．加えて，ハニーポット群の Session_table を同期し，評価値の更新を行う．
- **scanner**
  scanner では，analyzer から受け取った HTTP リクエストの送信元 IP アドレスとその周辺ネットワーク（/16），ハニーポット群が設置されている IP アドレスの周辺ネットワーク（/16）に対して，masscan [22] を用いて広域スキャンを行う．なお，この時のスキャン先ポートは，ハニーポットが攻撃を観測した際のアクセス URL に含まれていたポートである．ハニーポットに通信を行ってきたホストはマルウェア感染している可能性があり，その周辺ネットワークにもマルウェア感染ホストが存在する可能性がある．また，ハニーポットが設置されているネットワークの周辺に攻撃対象となり得るホストが存在する可能性があるため，こ

表 2　実験結果
Table 2　Experimental result.

	region1 honeypot(normal)	region1 honeypot1	region2 honeypot2	region3 honeypot3
reqest	71,797	100,872	67,519	63,011
ipaddr	1,569	1,579	2,753	2,966
sample	30	183	177	101

れらをスキャン対象とする．

- **getter**
  getter では，広域スキャンにおいてポートが開いていることを確認できたホストに対し，HTTP リクエストを送信してその応答を保存する．このリクエストは，analyzer で分析を行なった際に出力された HTTP リクエストのアクセス URL に対して行う．通信の結果，得られたレスポンスヘッダ，レスポンスボディ，ステータスコードからなる応答を保存し，同時に Session_table の更新を行う．レスポンスヘッダ，レスポンスボディはそれぞれハッシュ値 (md5) に変換した後に対応づけて保存する．なお，レスポンスヘッダをそのままハニーポット群の応答として用いると，正常に応答を行えない場合があるため，Content-Length などの一部のフィールド情報を削除して保存している．また，getter で用いる HTTP リクエストは，ハニーポット群で観測した攻撃を元に作成しているため，外部に攻撃を行う可能性がある．そのため，実際に送信する HTTP リクエストのヘッダ，データ部は攻撃とならないよう自作し，アクセス URL を手動確認することで，エクスプロイトが含まれないようにしている．

4. 評価実験

本章では，提案システムをインターネット上に設置することで，HTTP リクエストや攻撃の観測，検体の収集を行なった結果を説明する．また，既存の Web ハニーポットで観測された攻撃と検体の比較を行った結果を説明する．以降では，4.1 節で実験方法を説明し，4.2 節で実験結果を説明する．

4.1 実験方法

提案システムのハニーポット群を 2019 年 7 月 22 日から 2019 年 8 月 18 日までの 28 日間，クラウドサービスの一種である AWS [23] 上に設置した．観測には全 TCP ポート待受け，Session_table によって応答を適宜変更する 3 つのハニーポットを用いた．また，ハニーポットは全て異なるリージョンにて作成した．加えて，比較分析を行うため，既存の Web ハニーポットを AWS に設置した．なお，提案システムの分析サーバは ISP 回線上に設置し，スキャンやその応答の取得を行う旨と我々の連絡先を明記した Web ページを用意した上で，ホストへのスキャンを行なった．

4.2 実験結果

ハニーポット群で観測した HTTP リクエスト数，ユニークな IP アドレス数，検体数の結果を**表 2** に示す．以降では，提案システムの Web ハニーポットを honeypot1，honeypot2，honeypot3，既存の Web ハニーポットを honeypot(normal) と呼ぶこととする．同一リージョンに設置した honeypot1 と honeypot(normal) を比較した場合，どちらのハニーポットにおいても同様のポートに HTTP リクエストが届いているが，honeypot1 がより多くの HTTP リクエストを観測し，より多くの検体を取得できていた．どちらのハニーポットも，Web サーバがデフォルトで動作する 80/tcp 番ポート周辺や，関連した 8000/tcp 番ポート周辺で多くのリクエストを受けおり，特に，8088/tcp において，提案システムでは，既存の Web ハニーポットと比べて 28,000 件もの多くの HTTP リクエストを受けていた．加えて，IoT 機器を狙った Mirai やその亜種による通信と考えられる 60001/tcp 番ポートや 52869/tcp 番ポートへのリクエストが多く観測できていた．

4.2.1 エフェメラルポートへの攻撃

一般に，HTTP 通信の多くはウェルノウンポートである 80/tcp 番ポートが用いられるが，近年の機器やソフトウェアでは 1024 番ポートから 65535 番ポートのエフェメラルポート [14] を利用する場合がある．実際に，ハニーポット群のエフェメラルポートでのみ観測した攻撃の事例を**表 3** に示す．6780/tcp 番ポートへのリクエストでは，CDN 構築ツールである fikker[15] に対するログイン試行が行なわれ，POST のデータ部にユーザ名とパスワードが確認された．また，40424/tcp 番ポートへのリクエストでは，Hanbang[16] 製の IP カメラの脆弱性（CVE-2017-14335）を狙ったと思われる攻撃が観測された．さらに，40001/tcp 番ポートや 60001/tcp 番ポートでは，ディレクトリトラバーサルによる攻撃や OS コマンドインジェクションなど，マルウェアと考えられるファイルのダウンロードやその実行コマンドが記載された攻撃が観測された．

上記で述べた攻撃事例は，ハニーポット群でのみ観測された攻撃であるが，該当するアクセス URL の応答は変更していない．また，他のアクセス URL の応答を変更したこととの明確な因果関係は確認できていない．

4.2.2 広域スキャンで収集した応答

観測期間内にスキャンの対象となり，攻撃対象模擬部の Session_table に反映したポート，メソッド，パス，スキャン結果数，応答の取得数の一部を**表 4** に示す．スキャン対象となったポートは，複数のホストからのリクエストが多かったポートであり，Web サーバがデフォルトで動作する 80/tcp 番ポートに関連した 8000/tcp 番台のポートをはじめ，Oracle WebLogic が使用する 7000/tcp 番台のポー

表 3　エフェメラルポートへの攻撃
Table 3　Attacks against ephemeral port

Port	Method	Path	type/device/software
6780	POST	/fikker/webcache.fik?type=sign&cmd=in	fikker
40001	GET	WEBACCOUNT.CGI?OkBtn=++Ok++&RESULTPAGE=../../../../../../../../../../../../../../../Windows/system.ini&USERED IRECT=1&WEBACCOUNTID=&WEBACCOUNTPASSWORD===	ディレクトリトラバーサル
40424	GET	/ISAPI/Security/users	Hanbang IP Camera
60001	GET	/shell?cd%20/tmp;wget%20http:/%5C/XXX.XXX.116.64/Hilix.arm7%20-O%20zz;%20chmod%20777%20zz;./zz%20Jawsarm7	OS コマンドインジェクション

表 4　広域スキャンで収集した応答
Table 4　Responses collected by network scan.

Port	Method	Path	# Scan_result	# Response_result
81	GET	/	5,699	78
88	GET	/	210	65
7001	POST	/wls-wsat/CoordinatorPortType11	1,755	44
7002	GET	/wls-wsat/CoordinatorPortType11	1,525	5
7002	POST	/wls-wsat/CoordinatorPortType11	1,418	14
8080	GET	/manager/html	9,548	264
8080	GET	/TP/public/index.php	5,114	429
8088	POST	/ws/v1/cluster/apps/new-application	3,579	245
55555	POST	/tmUnblock.cgi	1,273	4

表 5　提案システムで取得した検体
Table 5　Collected samples

検知名 (Kaspersky)	検体名	件数
HEUR:Backdoor.Linux.Mirai.*	ankit.x86, UnHAnaAW.x86, ...	55
HEUR:Backdoor.Linux.Gafgyt.*	SinixV4.x86, Corona.x86_64, ...	38
HEUR:Trojan-Downloader.Shell.Agent.*	bins.sh, aws, ...	13
HEUR:Trojan-Downloader.Shell.Miner.gen.*	init.sh	2
Trojan.Win32.Siscos.wgv	download.exe	1
None	richard, frosty.x86, ...	4
—	bins.sh, soul.x86, x-8.6-.SNOOPY, ...	115

表 6　攻撃対象機器を模擬した高評価値の応答
Table 6　High evaluation value response simulating attack target device.

Port	Path	Body
8088	/ws/v1/cluster/apps/new-application	{ "application-id":"application_1563785736352_2234", "maximum-resource-capability":{ "memory":352924,"vCores":48 } }
55555	/tmUnblock.cgi	{ "errors":[{ "message":"No operations provided." }] }

ト，Cisco/Linksys ルータのファームウェアで利用している tmUnblock.cgi を狙った攻撃などが対象となった．

4.2.3　ハニーポットで取得した検体

観測期間内に提案システムで取得した検体は 228 件であり，このうち VirusTotal で 1 つ以上のアンチウイルスソフトで検知されたものは 112 件であった．また，検知レポートが得られなかったものが 115 件あった．VirusTotal の検知結果を**表 5** に示す．取得した検体の多くは IoT 機器を攻撃対象とする Mirai や Gafgyt に分類されており，IoT 機器を狙った攻撃が観測されていたことがわかった．

5.　考察

5.1　提案システムでのみ観測された攻撃

同一リージョンに設置したハニーポットのうち，提案システムを用いて応答を変更したハニーポットでは，既存の Web ハニーポットよりも多くのリクエストを観測することができた．特に，提案システムでは 8088/tcp 番ポートへの攻撃を多数観測する事ができた．当該ポートへのリクエストログを確認してみると”/ws/v1/cluster/apps/new-application”へのアクセスが多く発生しており，広域スキャンの応答をハニーポットに反映することで，”/ws/v1/cluster/apps”へのリクエストが発生していた．これらのリクエストは Apache Hadoop[17] の ResourceManager に対する攻撃であると考えられ，後者のリクエストでは，データ部に検体のダウンロードコマンドが確認された．したがって，攻撃者は応答の内容を確認して任意のコマンドが実行可能であるか判別していることが予想される．このような攻撃は評価実験の観測期間ではこの 1 件のみであったが，ハニーポットの応答によって攻撃を行うか判断するリクエストは他にも存在することが考えられるため，引き続き調査を行なっていく必要がある．

表 7　応答を変更することで取得できた検体
Table 7　Collected samples(only proposed system)

検体名	検知名 (Kaspersky)	検体取得時のアクセス URL
owari.x86	HEUR:Backdoor .Linux.Mirai.ba	8088/ws/v1/cluster/apps
hoho.x86	HEUR:Backdoor—par .Linux.Mirai.ba	8088/ws/v1/cluster/apps
8UsA.sh	HEUR:Trojan-Downloader.Shell.Agent.p	80/
hax.mips	HEUR:Backdoor .Linux.Mirai.ba	52869/picsdesc.xml

5.2　攻撃対象機器を模擬した応答

広域スキャンを用いてホストから取得した応答のうち，評価値（value）が高いと判断された応答の一部を**表 6** に示す．これは，アクセス URL が”8088/ws/v1/cluster/apps/new-application”に対する応答のうち，評価値が高くなった応答である．応答の内容は Apache Hadoop の Cluster New Application API[18] と同様のものと思われ，攻撃対象となっているソフトウェアの応答を模擬できていると考えられる．また，アクセス URL が”55555/tmUnblock.cgi”に対しての応答では，ハニーポットで変更した応答の全てで評価値が高くなっていた．これは同じアクセス元ホストが応答の種類のよらず，同じアクセス URL に対して GET，POST メソッドの HTTP リクエストを連続して送信してきているからだと考えられる．

5.3　応答を変更することで取得できた検体

提案システムで取得した検体のうち，提案システムで応答を変更した場合にのみ取得できた検体の一部を**表 7** に示す．取得できた検体の多くは Apache Hadoop に対する攻撃を観測した際に取得できたものであった．Apache Hadoop は分散処理ミドルウェアであり，通常は Linux サーバで利用される．しかし，VirusTotal の検知結果から，取得した検体の多くは IoT 機器を狙った Mirai や Gafgyt マルウェアであった．このことから，IoT 機器を狙っていたマルウェアの亜種が，Linux サーバも攻撃対象にし，利用されていると予想される．

5.4　ハニーポット検知の可能性

同一の応答で全ポート待受を行う攻撃観測部は，通常の Web サーバの挙動と大きく異なるため，攻撃者にハニーポットであると検知されてしまう可能性が高い．一方で，攻撃対象模擬部ではアクセス URL によって応答が変化するため，ハニーポットであると検知される可能性が低くなると考えられる．

今回の実験では，攻撃観測部と攻撃対象模擬部は共通であったが，異なるネットワークに構築することも可能であるため，両者でハニーポットとして検知され得る可能性がどの程度変化するのか，対象のホストがハニーポットか否かを判断するサービスである Honeyscore [24] を利用して調査を行なった．Honeyscore では既知のハニーポットの特徴量とホストの挙動を比較し，ハニーポットである度合いを 0.0 から 1.0 のスコアで算出している．

2019 年 8 月 11 日から 2019 年 8 月 18 日まで，Honeyscore を用いてハニーポット群を定期的に調査したところ，honeypot1，honeypot2，honeypot3 のスコアは [0.3, 0.8, 0.3] と一定値であった．同様に，2019 年 8 月 16 日に設置した既存のハニーポットを調査したところ，スコアは 0.3 であった．ここで，honeypot2 が比較的高いスコアとなり，ハニーポットとして検知された可能性がある．ハニーポット群の応答はアクセス URL によってランダムであるため，収集した応答の中に既知のハニーポットの応答が存在し，Honeyscore の調査時に該当の応答を返してしまったこと考えられる．honeypot1 と honeypot3 については，既存の Web ハニーポットと同様の値を示したが，これはハニーポットを設置した時間が短く，応答の変化の差が少ないことが考えられるため，今後も引き続き調査を行なっていく．

5.5　提案手法の限界

提案手法では HTTP 通信に注目して攻撃の観測を行なっているが，文字列ベースのデータの送受信を行う通信プロトコルであれば，HTTP 通信と同様に攻撃対象機器の模擬が可能である．一方，応答のみならず，機器やソフトウェア特有の動作や画面遷移，任意コマンドの実行結果を判別条件に含んでいる攻撃については，攻撃対象機器の動作を模擬をできない場合がある．

評価実験では，スキャン先ネットワークを限定しているが，得られる応答の数が少ない場合や攻撃対象となり得るホストが存在しない場合も考えられる．このため，スキャン先ネットワークへの負荷を考慮しつつ，さらに広範囲なスキャンを行うことでより多くの応答を観測することができる．また，デファクトスタンダードとして使用されているポートであれば，shodan[19] や censys[20] といった公的なスキャンサービスを用いてスキャン結果を入手し，利用することが可能である．

5.6　研究倫理

本研究では，観測した攻撃情報をもとにインターネット上に存在するホストに対してネットワークスキャンを行っている．そのため，送信される HTTP リクエストが外部への攻撃となる可能性がある．また，広範囲にネットワークスキャンを行うため，スキャン先のネットワークに負荷をかけてしまう可能性がある．そこで，本研究では以下の対策を行なっている．

(1) HTTP リクエストについて
　ハニーポットで観測した攻撃情報をもとにホストに対してスキャンを行っているが，送信される HTTP リ

クエストによっては外部への攻撃となってしまう可能性がある．そこで，ログイン試行やエクスプロイトが含まれる可能性があるクエリ文字列を取り除き，リクエストヘッダ，データ部を自作している．また，送信する HTTP リクエストは手動で確認し，不正なリクエストとならないことを確かめている．

(2) ネットワークスキャンの影響について
広範囲にネットワークスキャンを行うため，スキャン先ネットワークによっては帯域を圧迫してしまう可能性がある．そこで，スキャンレートを 10,000pps 程度に抑え，スキャンに使用するサーバの IP アドレスを固定し，スキャンを行なっている旨やその目的，また，連絡先を明記した Web サーバをたてることで，連絡があった際に，特定のネットワークをスキャン対象から外すことができるようにしている．

6. まとめと今後の課題

広域スキャンによりホストから収集した応答を用いて，Web ハニーポットの観測能力を向上させる手法を提案した．提案手法を実現したシステムを 2019 年 7 月 22 日から 2019 年 8 月 18 日までインターネット上に公開したところ，Apache Hadoop や Cisco/Linksys ルータのファームウェアを狙った攻撃を観測することに成功した．また，提案システムでは，既存のハニーポットよりも 28,000 件ほど多くの HTTP リクエストを観測した．8088/tcp では，既存のハニーポットでは観測できない，応答によって分岐を行う攻撃を観測し，より多くの検体を収集することができた．

今後は HTTP リクエストを行う際のヘッダや POST のデータ部に，ハニーポット群で観測できた攻撃情報を適切に使用できる機能を追加し，ホストの応答が正常に得られるように工夫を行う．また，現在の提案システムはランダムに応答を返すが，ホストから得られる応答の数が膨大になる場合があるため，応答に優先順位をつけるなど，応答のアルゴリズムについて改良を行う．収集した検体では，Mirai や Gafgyt といった IoT 機器を攻撃対象としていたマルウェアの亜種が，Linux サーバを狙った攻撃に利用されていたため，マルウェア動的解析を行うことで攻撃者の意図をより明確に把握し，攻撃への対策を講じる．

謝辞 本研究成果の一部は、国立研究開発法人 情報通信研究機構 (NICT) の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」により得られた．

参考文献

[1] Dionaea，Welcome to dionaea's documentation! https://github.com/DinoTools/dionaea.
[2] Graftopf, mushorg/glastopf: Web Application Honeypot - GitHub https://github.com/mushorg/glastopf.
[3] phpmyadmin_honeypot，gfoss/phpmyadmin_honeypot: A simple and effective ... - GitHub https://github.com/gfoss/phpmyadmin_honeypot.
[4] Wordpot，gbrindisi/wordpot: A Wordpress Honeypot - GitHub https://github.com/gbrindisi/wordpot.
[5] Nicter Web 2.0 https://www.nicter.jp.
[6] 田辺瑠偉，上野航，吉岡克成，松本勉,"ハニーポットによる Apache Struts の脆弱性に対する攻撃の観測"，情報処理学会論文誌，Vol.60，No.3，2019.
[7] MartinIngesen/HonnyPotter: WordPress Honeypot - GitHub https://github.com/MartinIngesen/HonnyPotter.
[8] bwpot，graneed/bwpot: 高対話型ハニーポット - GitHub https://github.com/graneed/bwpot.
[9] Docker https://www.docker.com.
[10] 森下瞬，上野航，田辺瑠偉，吉岡克成，松本勉，"シグネチャベースの検知に基づくオープンソースハニーポットの実態調査"，暗号と情報セキュリティ シンポジウム,2018.
[11] Wowhoneypot，morihisa/WOWHoneypot: 簡単に構築可能で、シンプルな機能で ... - GitHub https://github.com/morihisa/WOWHoneypot.
[12] Tongbo Luo, Zhaoyan Xu, Xing Jin, Yanhui Jia, Xin Ouyang, "IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices", Black Hat USA, 2017.
[13] 牧田大佑，島村隼平，久保 正樹，井上大介，"全ポート待受型の簡易ハニーポットによるサイバー攻撃観測"，暗号と情報セキュリティ シンポジウム,2019.
[14] RFC 6056 - Recommendations for Transport-Protocol ... - IETF Tools https://tools.ietf.org/html/rfc6056#section-3.2.
[15] Fikker - 自建 CDN — 搭建 CDN — 反向代理件 https://www.fikker.com.
[16] Hanbang Digital Technology Co., Ltd. http://www.hbgk.net/en.
[17] Apache Hadoop https://hadoop.apache.org.
[18] Hadoop Apache Hadoop 3.1.0 https://hadoop.apache.org/docs/r3.1.0/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Cluster_New_Application_API.
[19] Shodan https://www.shodan.io.
[20] Home — Censys https://censys.io.
[21] VirusTotal https://www.virustotal.com.
[22] Masscan https://github.com/robertdavidgraham/masscan.
[23] Amazon Web Service https://aws.amazon.com/jp/.
[24] Honeyscore https://honeyscore.shodan.io/.