# 高対話型 Web ハニーポットにおける攻撃情報収集方式の改善

八木 毅† 谷本 直人† 針生 剛男† 伊藤 光恭†

†NTT 情報流通プラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11
{yagi.takeshi,tanimoto.naoto,hariu.takeo,itoh.mitsutaka}@lab.ntt.co.jp

**あらまし** 本稿では，高対話型Webハニーポットへの攻撃の宛先URLを変換することで攻撃を制御し，収集する攻撃情報量を改善する方式を提案する．近年，Webアプリケーションのぜい弱なプログラムを悪用してWebサーバをマルウェアに感染させる攻撃が多発している．この攻撃の一解析手段として高対話型Webハニーポットでの情報収集が挙げられる．しかし，この攻撃の多くはツールで自動化されており，実在しないパス名が宛先URLに記述される可能性が高く，その結果，攻撃が失敗して収集情報が限定される．インターネット実験では，97%の攻撃が失敗するが，提案方式により，その約50%から攻撃成功時と同じ情報を収集できることを確認した．

# Improvement of Attack Information Collection Schemes on High-Interaction Web Honeypots

Takeshi Yagi† Naoto Tanimoto† Takeo Hariu† Mitsutaka Itoh†

†NTT Information Sharing Platform Laboratories
3-9-11 Midori-cho Musashino-Shi Tokyo, 180-8585 Japan
{yagi.takeshi,tanimoto.naoto,hariu.takeo,itoh.mitsutaka}@lab.ntt.co.jp

**Abstract** This paper proposes schemes for high-interaction web honeypots to collect a lot of attack information. Conventional high-interactive web honeypots can collect only limited information from attacks, whose path of destination URL dosen't match path structure of the web honeypot, because these attacks are failure. In our proposal, destination URL of these attacks will be converted by guessing the right path from path structure of the web honeypot. Our investigation reveals that 97 percent of attacks are failure. On the other hand, we confirmed that about 50 percent of these attacks will succeed by our proposal.

## 1 はじめに

インターネットの社会インフラ化に伴い，フィッシングやスパム配信などのサイバー攻撃が急増している．この攻撃の多くは，マルウェアと呼ばれる，攻撃者が作成した悪意あるツールを用いて実施される．攻撃者は，ユーザの端末やサーバにマルウェアを配布し，その後，マルウェアをリモート操作することで，端末やサーバを不正に制御する．近年，マルウェアの多くは HTTP 経由で配布されている[1]．この主な要因の一つとして，一般の Web サイトを踏み台にして攻撃元を隠ぺいする技術の普及が挙げられる．Web サイトは，Web アプリケーションのぜい弱性を悪用する攻撃を受けてマルウェアに感染し，踏み台サイトとして新たな攻撃に利用される．

このような攻撃を防御するために既存の攻撃を収集解析する手法として，ハニーポット[2]が検討されている．特に Web サイトへの攻撃を観測するハニーポットは Web ハニーポット[3]と呼ばれ，低対話型と高対話型に分類される．特に多種多様な Web サイトへの攻撃を収集して挙動を解析する場合は，ぜい弱な Web アプリケーションを搭載して実際に攻撃を受ける高対話型 Web ハニーポット[4]が適用される．

しかし，高対話型 Web ハニーポットのディレクトリ構造は，搭載する Web アプリケーションのバージョンやオペレータ判断に応じて変化する．一方，攻撃の多くはツールにより自動化されており，宛先として指定される URL 内のパス名は，全ターゲット Web サイトに対して共用される．このため，実際にぜい弱なプログラムが配置されたパスと，攻撃の宛先 URL のパス名が一致しない可能性が高い．その結果，多数の攻撃が失敗し，収集可能な攻撃情報量が制限される可能性がある．

この問題を解決するために，本稿では，攻撃の宛先 URL のパス名を，Web ハニーポットのディレクトリ構造に応じて変換する方式を提案する．提案方式では，実在しない宛先 URL のパス名を，逆ロンゲストマッチにより抽出した Web ハニーポットに実在するパスに変換する．さらに本稿では，プロトタイプを用いたインターネット実験結果から，提案方式の効果を考察する．

## 2　攻撃モデル

Web サイトへの攻撃には，特定の Web サイトに特化した攻撃を実施するスピア型と，ツールを用いて超多数 Web サイトへの攻撃を実施する無差別型が存在する．前者は Web サイトが管理する重要情報を不正入手する際に適用され，後者は踏み台サイトのように悪用可能な Web サイトを確保する際に適用される．Web サイトを悪用する攻撃としては，ユーザが閲覧するコンテンツを改ざんする SQL インジェクション[5] や，悪意あるスクリプトを混在させるクロスサイトスクリプティング[6]などが存在するが，Web サイトを

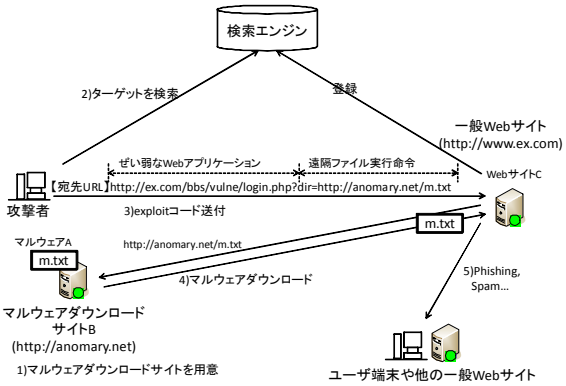

図 1　攻撃モデル

マルウェアに感染させて踏み台化する場合は，RFI(Remote File Inclusion)が有効な手段となる[7]．RFI は，図1に示すように，以下のステップをとる．

1）攻撃者は，マルウェア A を配置したマルウェアダウンロードサイト B を用意する．サイト B は，攻撃者に不正に利用された一般の Web サイトである場合が多い．
2）攻撃者は，検索エンジン等を利用して，ぜい弱な Web アプリケーションを搭載した Web サイト C を特定する．
3）攻撃者は，Web サイト C に対して，サイト B からマルウェア A をダウンロードするよう，Web アプリケーションのぜい弱性を利用した exploit コードを送信する．
4）Web サイト C は，サイト B からマルウェア A をダウンロードして実行する．

Web サイトへの攻撃で使用されるマルウェアは，OS 情報などターゲット Web サイトのホスト情報を HTTP レスポンスメッセージやメールに記述して攻撃者に送付するタイプと，ターゲットを不正操作するタイプが存在する．攻撃者は，前者の攻撃が成功した Web サイトをリスト化し，リストに掲載された Web サイト群に対して後者の攻撃を実施する．

このように複雑化する攻撃から Web サイトを防衛するためには，多種多様な攻撃を収集解析し，防御に利用できる情報を明らかにすることが非常に重要となる．

## 3　関連研究

Web サイトへの既存の攻撃を収集解析する Web ハニーポットは，低対話型と高対話型に分類される．低対話型 Web ハニーポット[8]は，特定の Web アプリケーションの動作をエミュレートして攻撃を監視するものである．一方，高対話型 Web ハニーポットは，ぜい弱性が存在する実際の Web アプリケーションで攻撃を監視するものである．前者は，エミュレートした範囲に機能が制限されるため，マルウェアに感染することなく比較的安全に攻撃情報を収集できる反面，収集可能な情報が制限される．一方，後者は，実際にマルウェアに感染して挙動を観測するため，感染に伴う被害発生の可能性があるが，感染時の情報を詳細に観測できる．

低対話型 Web ハニーポットでは，マルウェアに感染しないため，マルウェアによる攻撃者へのホスト情報送付は実施されない．このため，攻撃が成功した Web サイトとして攻撃者に認識されない可能性が高い．さらに，マルウェアが新たなマルウェアのダウンローダとして起動する攻撃に対しては，新たなマルウェアをダウンロードできない．このため，Web サイトへの多種多様な攻撃を観測する場合は，高対話型 Web ハニーポットでの観測が適切である．

## 4　従来技術の問題点

高対話型 Web ハニーポットは，ぜい弱な Web アプリケーションをインストールした囮 Web サイトを基盤として構築される．この際，囮 Web サイトのディレクトリ構造は，Web アプリケーションのバージョンやインストール手順に応じて変化する．一方，図 2 に示すように，攻撃に用いられるツールでは，ツールの作成者および攻撃者により，宛先 URL に記述するパス名がリストなどで予め規定されている．この結果，Web ハニーポットのディレクトリ構造と攻撃の宛先 URL に記述されたパス名が一致せず，攻撃が失敗する可能性がある．攻撃が失敗した場合，マルウェアは収集できず，攻撃後に発生する挙動は観測

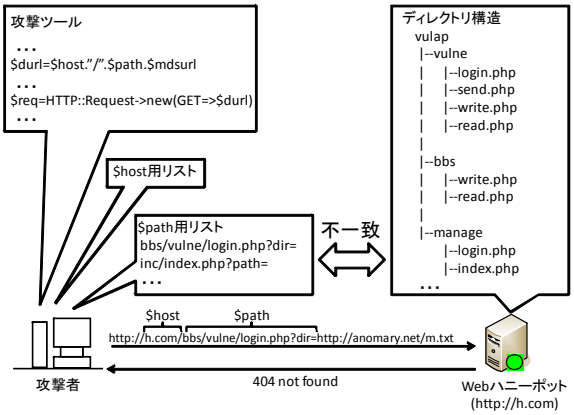


図 2　従来技術の問題点

できない．さらに，攻撃が失敗した場合，攻撃者への HTTP レスポンスメッセージは 404 not found などのエラーメッセージとなる．このため，攻撃者が Web ハニーポットにぜい弱性がないと判断し，その後の攻撃対象から Web ハニーポットが除外される可能性がある．

## 5　提案の攻撃情報収集方式

### 5.1　提案の概要

提案方式では，インターネットと Web ハニーポット間に配置される HTTP 転送機能が，宛先 URL のパス名が Web ハニーポットのディレクトリ構造内に実在するかを調査する．さらに，実在しない宛先 URL のパス名を特定した際に，攻撃者が攻撃目標としている可能性が高いパスを推測して宛先 URL のパス名を変換する．宛先 URL のパス名を推測するための手段としては，キャッシュテーブルとアルゴリズムを備える．

### 5.2　ディレクトリ構造との整合性調査

HTTP 転送機能は，Web ハニーポットのディレクトリ構造を記憶するとともに，受信した HTTP データの宛先 URL に記載されたパス名を抽出し，両者の整合性を調査する．

宛先 URL のパス名と一致するパスが Web ハニーポット上で確認できた場合は，宛先 URL を

変換せずに Web ハニーポットへ HTTP データを転送する. 一方, 一致するパスを確認できなかった場合は, 後述のキャッシュテーブルによる URL 変換を試行する.

### 5.3 キャッシュテーブルによる URL 変換

HTTP 転送機能は, 攻撃者が指定する可能性が高いパス名と, そのパス名に対する変換候補となる Web ハニーポット上のパス名の対を, キャッシュテーブルとして保有する. キャッシュテーブルの初期エントリは, 過去の観測や攻撃ツールの解析結果からオペレータが作成する.

整合性調査において, 変換が必要であると判断された宛先 URL について, パス名を検索キーとしてキャッシュテーブルを検索する. この結果, エントリを特定できた場合は, 宛先 URL のパス名を, 当該エントリに記載されている変換候補パス名に変換して Web ハニーポットへ HTTP データを転送する. 一方, エントリを特定できなかった場合は, 後述のアルゴリズムによる URL 変換を試行する.

### 5.4 アルゴリズムによる URL 変換

HTTP 転送機能は, 宛先 URL を変換するためのアルゴリズムを記憶するとともに, キャッシュテーブルでは対応できない URL の変換を試行する. 今回は, 以下に示す, 逆ロンゲストマッチに基づく変換アルゴリズムを採用する.

本アルゴリズムでは, キャッシュテーブルで変換できない URL のパス名と類似したパスを, Web ハニーポットに実在するパス群から抽出する. まず, パス名の末尾に記載されるファイル名を検索キーとして, ディレクトリ構造を検索し, ファイル名が一致する Web ハニーポット上のパスを特定する. ここで, パスが特定できなかった場合は, 変換を断念して Web ハニーポットへ HTTP データを転送する. また, パスを特定し, 且つ, それが単一である場合, 宛先 URL のパス名を特定したパスに変換する. 一方, 特定したパスが複数である場合, 一階層上のディレクトリ名が一致するパスを選択する. このように, ディレクトリ名が下位から最長一致するパスを特定し, 宛先 URL のパス名を, 特定したパスに変換する. なお, この際特定した変換前と変換後のパスの対からキャッシュテーブルの新規エントリを生成することで, 以降の変換処理を高速化できる.

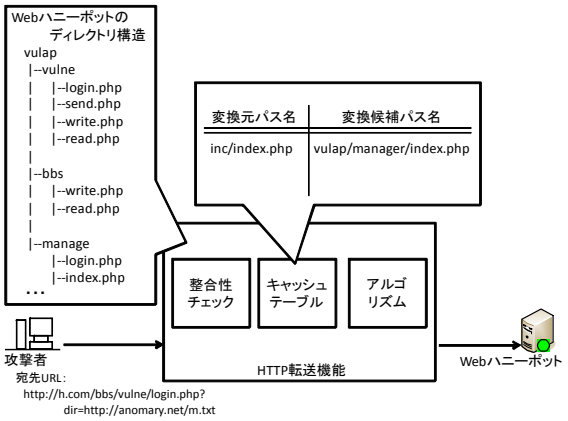


図 3 提案方式の概要

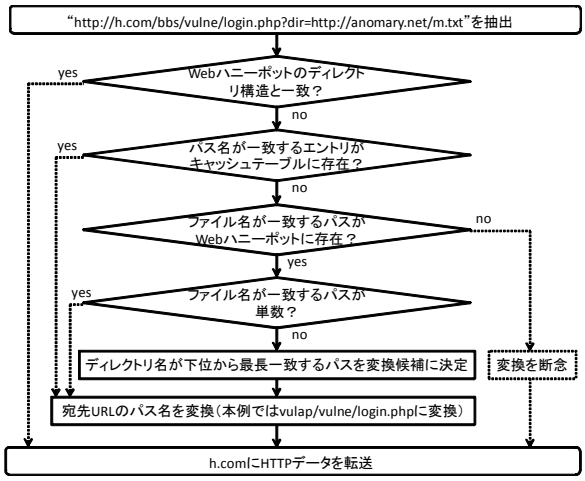


図 4 シーケンス例

図 3 の構成において, HTTP 転送機能が HTTP データを受信したと仮定する. この際, HTTP 転送機能は, 図 4 のシーケンスに従い, URL 変換を試行する. 宛先 URL のパス名は, Web ハニーポットのディレクトリ構成と一致しない. さらに, キャッシュテーブルでも変換候補を特定できない. このため, アルゴリズムでの変換を試行する. まず, 宛先 URL のパス名に記載されたファイル名と同一のファイル名を保有する

パスを複数特定できる. その後, 一階層上のディレクトリ名が一致する候補を検索することで, 変換候補となるパス名を特定する.

このように, 提案方式では, 宛先 URL のパス名が実在しない攻撃に関しても, ファイル名が実在しない攻撃以外は, 宛先のパス名を推定して URL を変換する. これにより, Web ハニーポットが搭載する Web アプリケーション宛である可能性が高い攻撃のみを故意に成功させる.

# 6 インターネット実験結果

## 6.1 概要

高対話型 Web ハニーポットを適用する目的は, マルウェアに感染して攻撃者に HTTP レスポンスを返信することで, 攻撃が成功する Web サイトとして攻撃者に認識されることである. さらに, マルウェアが新たなマルウェアのダウンローダとして起動する攻撃を含め, 可能な限り多くのマルウェアを収集することも重要でなる.

ここでは, 高対話型Webハニーポットのプロトタイプをインターネットに接続し, 提案方式の効果を評価する. なお, 今回の実験では, 提案方式をプロトタイプに搭載せず, プロトタイプで採取したデータを机上で分析することで提案方式の効果を算出した.

## 6.2 プロトタイプ

インターネット実験に際し実装した高対話型 Web ハニーポットのプロトタイプは, 図 5 に示すように, ぜい弱な Web アプリケーションを搭載した囮 Web サイトを保有する. さらに, 囮 Web サイトへの通信ログを取得しつつ, 囮 Web サイトが感染したマルウェアを定期的に駆除するコントローラ[9]を配置する. 提案方式を実装する場合は, コントローラへの実装を想定している. なお囮 Web サイトの OS には Linux を採用し, Web アプリケーションは PHP で開発されたぜい弱な Web アプリケーションを採用した.

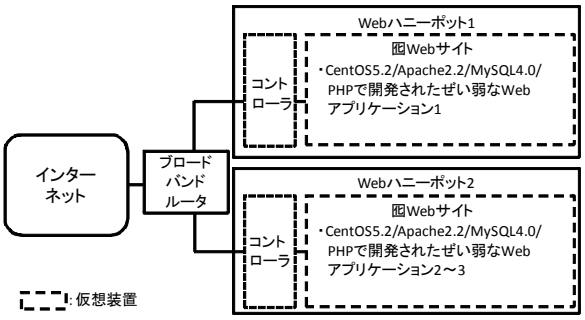


図 5 インターネット実験環境

表 1 インターネット実験結果

| | 総数 | 成功した攻撃 | 提案方式の適用により成功する攻撃 |
|---|---|---|---|
| 攻撃件数 | 1035 | 33(3.2%) | 516(49.9%) |
| マルウェア種別数 | 63 | 14(22.2%) | 45(71.4%) |

## 6.3 プロトタイプを用いた実験結果

プロトタイプをインターネットに接続し, 2009 年 1 月 30 日から 2009 年 7 月 22 日に収集した攻撃を分析し, 提案方式の効果を評価する. なお, 本実験では, 3 種類の Web アプリケーション宛の RFI 攻撃を対象とした. 攻撃全体, 成功した攻撃, 本来失敗するが提案方式により成功させることが可能な攻撃の件数と, 各攻撃において採取可能なマルウェア種別数を表 1 に示す. この際, マルウェア種別数は SHA1 値の比較により算出した. また, マルウェア種別数の総数に関しては, 失敗した攻撃を手動解析してマルウェアをダウンロードし, SHA1 値を計算することで算出した.

攻撃件数に着目した場合, 攻撃総数に対して, 成功した攻撃は 3.2%であった. 一方, 提案方式により故意に成功させることができる攻撃は 49.9%であった. また, 成功した攻撃で収集したマルウェアは 22.2%である一方, 提案方式の適用により成功する攻撃から採取可能なマルウェアは 71.4%であった.

## 6.4 考察

提案方式では, 本来失敗する攻撃の約 50%

を故意に成功させる. 一方, 提案方式では, 約71%のマルウェアを収集できる. これは, 攻撃件数の割合以上にマルウェアを効率的に収集できることを示している.

また, 提案方式で宛先 URL を変換できない約 50%の攻撃には, プロトタイプが搭載していない Web アプリケーション宛の攻撃が混在していた. さらに, 入手した複数の攻撃ツールで予め設定されていたパス名が宛先 URL に記述されていた. このような宛先 URL を保有する全ての攻撃を故意に成功させた場合, 振舞いが不自然になる可能性が高い. このため, Web ハニーポットであることが攻撃者に検知される可能性がある. このことを考慮すると, 提案方式のように, ディレクトリ構造に類似したパス名を保有する宛先 URL のみを変換対象とする方式が適切であると考えられる.

なお, 今回成功した攻撃に関しては, 囮 Web サイトがイニシエートした宛先をコントローラで監視することで, マルウェアダウンロードサイトを機械的に特定できた. 一方, 成功しない攻撃に関しては, 通信ログおよび exploit コードを手動解析してマルウェアダウンロードサイトを特定した. このように, 提案方式により, 約 50%の攻撃に関して, マルウェアダウンロードサイトを機械的に抽出することが可能となる. これにより, 解析コストの削減が期待できる.

## 7 おわりに

本稿では, 低精度な無差別攻撃の宛先 URL を制御して高対話型 Web ハニーポットで収集可能な攻撃情報量を改善する手法を提案した.

提案方式では, 受信した攻撃の宛先 URL に記載されたパスが Web ハニーポット上に実在しない際に, 攻撃者が攻撃目標としている可能性が高いパスを推測して宛先 URL を変換する. 本稿では, 推測方法として, 宛先URLのパス名とディレクトリ構造の逆ロングストマッチをとる方法を採用し, その効果を示した.

インターネット実験では, 提案方式を適用することにより, 実在しないパス名を保有する宛先 URL の約 50%を, 実在する宛先 URL へ変換できる見通しを得た. なお, 提案方式での変換が困難な約 50%の攻撃には, Web ハニーポットが搭載していない Web アプリケーション宛の攻撃が混在していた. これらを故意に成功させた場合, 攻撃者に不自然な振舞いだと発見される可能性がある. このため, 提案方式の変換成功率は適切な範囲だと考えられる.

提案方式の適用により, 高対話型 Web ハニーポットで収集可能な攻撃情報を改善することが可能となる. その結果, 多種多様な攻撃を収集して攻撃の実態を詳細に解析できる. これにより, Web サイトを防御するために適用できる情報の質と量を改善でき, 安心・安全な Web サイトを運用する環境を構築できる.

## 参考文献


[1]http://www.symantec.com/business/resources/articles/article.jsp?aid=20080513_sym_report_attacks_increasingly

[2]http://www.honeynet.org/

[3]http://sites.google.com/site/webhoneypotsite/

[4]http://www.honeynet.org/project/HIHAT

[5]Anley Chris. “Advanced SQL Injection In SQL Server Applications.” An NGSSoftware Insight Security Research (NISR) Publication, 2002.

[6]http://httpd.apache.org/info/css-security/

[7] H. F. G. Robledo, “Types of hosts on a Remote File Inclusion (RFI) botnet,” Electronics, Robotics and Automotive Mechanics Conference (CREMA) 2008, Sep, 2008.

[8]http://www.honeynet.org/gsoc/project8

[9]八木毅, 谷本直人, 浜田雅樹, 伊藤光恭, “プロバイダによる Web サイトへのマルウェア配布防御方式”, 信学技報, IN2009-34, 2009.