情報処理学会論文誌 Vol.61 No.3 555–566 (Mar. 2020)

動的解析を利用したフィッシングサイトのアクセス妨害機能の実態解明

小寺 博和[1,a)] 芝原 俊樹[1] 千葉 大紀[1] 青木 一史[1] 波戸 邦夫[1] 秋山 満昭[1]

受付日 2019年6月14日, 採録日 2019年11月29日

概要：フィッシングによる攻撃は過去最大規模に拡大している．フィッシング対策の 1 つに，ユーザによるフィッシングサイトへのアクセスをブラックリストを用いてフィルタリングする方式がある．セキュリティベンダに代表される解析者が疑わしいウェブサイトにアクセスしてフィッシングサイトであると判定することでブラックリストに追加される．しかしながら，フィッシングサイトにはサーバサイドで実装されるアクセス妨害機能が存在し，アクセス時の条件によりアクセスできない場合がある．フィッシングサイトを確実にブラックリストに追加するために，アクセス妨害機能の実態を把握することは重要であるが，インターネット上に実在するフィッシングサイトでこの機能がどの程度動作しているかは明らかではない．本研究では，フィッシングサイトのアクセス妨害機能の解析手法の提案と，アクセス妨害機能に関する大規模な調査を実施した．フィッシングキットと呼ばれるフィッシングサイト構築ツールに対して HTTP リクエストを送信することでアクセス妨害機能を動作させるための条件を解析し，フィッシングサイトのアクセス妨害機能の有無を調査した．調査の結果，インターネット上に実在する 4,901 件のフィッシングサイトのうち，HTTP リクエストで設定可能な User-Agent と Referer によるアクセス妨害機能を有するものが 10.4%あることが判明した．

キーワード：フィッシング，フィッシングキット，アクセス妨害

Understanding Cloaking Techniques of Phishing Websites by Dynamic Analysis

Hirokazu Kodera[1,a)] Toshiki Shibahara[1] Daiki Chiba[1] Kazufumi Aoki[1] Kunio Hato[1] Mitsuaki Akiyama[1]

Received: June 14, 2019, Accepted: November 29, 2019

Abstract: Phishing attacks have been ever-increasing on the Internet today. One of the promising countermeasures for phishing attacks is filtering by using blacklists. Such blacklists are composed of URLs of phishing websites and have been maintained by accessing/detecting the URLs. However, some phishing websites have implemented a special access control technique in server-side called cloaking to prevent them from accessing/detecting by the provider of such blacklists. In order to improve blacklist-based countermeasures, we need to better understand the cloaking technique and its actual situation on the Internet. To this end, we propose a new method to analyze the cloaking techniques of phishing websites and conduct a large-scale measurement study of them. Specifically, we analyze a specific condition to activate the cloaking techniques by focusing on phishing kits which are commonly used to deploy phishing websites today. We reveal that 10.4% of 4,901 real/active phishing websites implement the cloaking technique relying on User-Agent and Referer which can be configurable by blacklist providers.

Keywords: phishing, phishing kit, cloaking technique

1. はじめに

フィッシングは正規サイトを装ったWebサイトに誘導されたユーザが入力した認証情報を窃取する攻撃として知られている．フィッシングによる被害は拡大傾向にあり [1]，トレンドマイクロ社によると2018年上半期における日本からフィッシングサイトへ誘導された件数は過去最多の290万件となり，2017年下半期比の2.7倍となっている [2], [3]．フィッシングにより窃取された認証情報の多くはアンダーグラウンドマーケットで売買されているとの報告がある [4]．

フィッシングサイトを構築するための手法の1つとしてフィッシングキットと呼ばれるツールキットが利用されている．フィッシングキットとはフィッシングサイトを構築するためのファイルがパッケージ化されたアーカイブファイルで，フィッシングキットをWebサーバ上で展開することで容易にフィッシングサイトを構築できる [5]．フィッシングキットには，正規のWebサイトになりすますためのWebページ構成ファイルやフィッシングサイトにユーザが入力した認証情報を攻撃者に転送するためのコードに加えて，セキュリティベンダ等による解析を目的としたアクセスを妨害するためのコードが含まれる場合がある．

フィッシング対策の1つとしてブラックリストを用いてフィッシングサイトへのアクセスをフィルタリングする方式がある [6]．フィッシングサイトをブラックリストに登録するためには，疑わしいウェブサイトにアクセスしてフィッシングサイトであると判定する必要がある．しかしながら，フィッシングサイトには解析を妨害するためのサーバサイドで実装されるアクセス妨害機能が存在するため，解析者がフィッシングサイトのコンテンツを取得できず，解析に失敗することが想定される．フィッシングサイトに正しくアクセスするためにはアクセス妨害機能の実態を明らかにすることが必要である．Drive-by Download攻撃を行う悪性サイトにもアクセスした被害者の環境をクローキングすることが知られている [7]．一方，フィッシングサイトは正規のWebサイトを装って認証情報を窃取することが目的であり，異なるクローキング手法を持つことが考えられるため，フィッシングサイト特有のアクセス妨害機能の実態を明らかにすることが必要である．

アクセス妨害機能はWebサーバの動作を制御するために用いられるApacheの`.htaccess`や，サーバサイドで動作するPHPで実装されるため，フィッシングサイトのコンテンツ解析による手法 [8], [9], [10] ではアクセス妨害機能のためのコードを解析できない．フィッシングキットを入手し，その中からアクセス妨害のためのコードを入手することでアクセス妨害機能の解析が可能となる．Oestら [11] は2,300個以上のフィッシングキットに含まれる`.htaccess`ファイルを解析し，アクセス妨害の対象とされやすいIPアドレスの国別の傾向や，組織別の傾向を調査した．しかしながら，既存研究 [11] ではPHPで実装可能なアクセス妨害機能については明らかにされていない．また，攻撃者が正規のWebサイトの改ざんによりフィッシングサイトを構築した場合にWebサーバの設定で`.htaccess`による制御が有効化されていない可能性がある．したがって，`.htaccess`だけでなく，PHPによるアクセス妨害機能についてもその実態を調査することが重要である．また，フィッシングキットが入手できないフィッシングサイトも存在するため，既存研究のような入手したフィッシングキットを解析するだけではインターネット上に存在するフィッシングサイトが持つアクセス妨害機能の実態を明らかにできていない．

本研究ではフィッシングサイトのアクセス妨害機能の動作を解析するための手法を提案し，インターネット上に実在するフィッシングサイトを対象にアクセス妨害機能に関する大規模な調査を実施した．フィッシングサイトのURLブラックリスト [12], [13] に掲載されたフィッシングサイトから入手したフィッシングキットに対してHTTPリクエストを送信し，その応答を観測することでアクセス妨害機能を動作させるHTTPリクエストヘッダの条件を解析した．解析した条件を用いてインターネット上のフィッシングサイトにアクセスし，アクセス妨害機能有無を調査した．アクセス妨害機能の動作条件を解析し，インターネット上のフィッシングサイトへのアクセス回数を最小化することで，調査対象のWebサーバの負荷軽減と調査の効率性を向上させた．これらにより，PHPで実装されたアクセス妨害機能や，フィッシングキットが入手できないフィッシングサイトについてもアクセス妨害機能の有無を調査可能にした．本論文の主な貢献を以下に示す．

- フィッシングキットが持つアクセス妨害機能の動的解析によりアクセス妨害機能を動作させる最小の条件を解析することで，効率的にフィッシングサイトのアクセス妨害機能有無を調査する手法を提案した．
- 収集した4,917件のフィッシングキットのアクセス妨害機能を提案した手法で解析した．その結果，`.htaccess`によるアクセス妨害機能を有するものが11.8%，PHPによるアクセス妨害機能を有するものが2.3%存在し，検索エンジンや正規のWebサイトにリダイレクトすることでアクセス妨害するものも存在するという知見を得た．
- フィッシングキットのアクセス妨害機能解析から得られた13種のUser-AgentとRefererのパターンを用いて実在するフィッシングサイトのアクセス妨害機能有

[1] NTTセキュアプラットフォーム研究所
NTT Secure Platform Laboratories, Musashino, Tokyo 180–8585, Japan
[a)] hirokazu.kodera.dh@hco.ntt.co.jp

無を調査した結果，User-Agent と Referer によるアクセス妨害機能を持つものは 10.4%存在したことを明らかにした．

本論文の構成は以下のとおりである．2 章で関連研究を示す．3 章でフィッシングサイトで用いられるアクセス妨害機能に関して述べる．4 章で動的解析を用いたフィッシングキットのアクセス妨害機能の解析手法を提案する．5 章で提案手法を用いた調査手法の詳細を示し，6 章は調査対象としたフィッシングキットの解析結果を報告する．7 章は本研究における制約，今後の課題，研究倫理について述べ，8 章で本論文のまとめを行う．

2. 関連研究

フィッシングに関する研究として，ブラックリストを用いたフィッシング検知に関する内容と，フィッシングキットの解析に関する内容が報告されている．

フィッシングキット解析

Cova ら [5] は無料で配布されているフィッシングキットやフィッシングサイトに残されたままになっているフィッシングキットを 500 個以上収集し，フィッシング攻撃のターゲットとなる組織，ユーザが入力した認証情報の転送手法，難読化手法を明らかにした．また，約 3 分の 1 のフィッシングキットに認証情報をフィッシングキット作成者に送信するためのバックドアが存在することを明らかにした．Thomas ら [14] はフィッシングサイトに入力された認証情報をメールで攻撃者に送信するためのメールテンプレートをフィッシングキットから抽出した．メールテンプレートを用いてフィッシングサイトから Gmail に送信された認証情報が含まれるメールを解析し，フィッシングサイトに入力される情報の種類や被害者が利用するメールプロバイダやパスワードの傾向を明らかにした．Zawoad ら [15] はフィッシングキットに含まれる認証情報送信先メールアドレスをもとにフィッシングサイトとフィッシングサイト設置者をクラスタリングした．クラスタリングの結果から 1,475 個のフィッシングサイトが 317 のユーザによって構築されたものであることを明らかにした．

ブラックリストを用いたフィッシングサイト検知

Sheng ら [6] は 8 個のフィッシング対策ツールバーが持つブラックリストを対象にブラックリストの有効性の調査を実施した．ブラックリストの更新速度やカバレッジが異なることを明らかにし，ヒューリスティック検知を用いた Microsoft と Symantec ではより早期にフィッシングサイトを検知できたことを明らかにした．Oest ら [16] はフィッシングキットで用いられるアクセス妨害のための HTTP リクエストフィルタを 6 つのカテゴリに分類し，それらを使用した 2,380 個のフィッシングサイトをインターネット上に配置した．配置したフィッシングサイトがブラックリストに登録されるまでのタイムラインを測定することでブラックリストの有効性を評価した．Tsalis ら [17] はモバイルデバイスで用いられるブラックリストによる Google Safe Browsing のフィッシング対策機能を評価し，PC で用いられるフィッシング対策機能とは機能に差があることを明らかにした．

フィッシングサイトの実態調査

Chhabra ら [18] はソーシャルメディアの 1 つである Twitter において，PhishTank に掲載されたフィッシングサイトの URL 情報をもとに Twitter における URL 短縮サービスを用いたフィッシングの実態調査を実施した．

フィッシングサイトのブラックリストの評価や，フィッシングキットの解析や解析結果から得られた情報を用いたアクセスしたユーザの調査に関する研究であるが，実在するフィッシングサイトのアクセス妨害機能に関する調査を行っていない．

Listing 1 .htaccess による Referer およびアクセス元 IP アドレスに対するアクセス妨害実装例

```
1 RewriteEngine on
2 RewriteCond %{HTTP_REFERER} example\.com
3 RewriteRule ^.* - [F]
4
5 order allow,deny
6 deny from 192.0.2.0/24
7 deny from example.com
8 allow from all
```

3. フィッシングサイトにおけるアクセス妨害

本章ではフィッシングサイトで動作するアクセス妨害機能について述べる．

フィッシングサイトのアクセス妨害機能は，Web サーバの動作を制御するために用いられる Apache の `.htaccess` や，サーバサイドで動作する PHP により実現される．`.htaccess` は Web サーバの設定を実施し，`.htaccess` ファイルを配置することで動作させることができる．また，PHP によるアクセス妨害機能は，アクセス妨害機能を持つ PHP ファイルを配置し，フィッシングサイトの入口となるファイルから呼び出して動作させることができる．

アクセス妨害機能は，アクセス元 IP アドレスや，HTTP リクエストヘッダに含まれる User-Agent や Referer を主な対象とされることが知られている [11]．このようなアクセス妨害機能は Listing 1，Listing 2 に示すように `.htaccess` および PHP によって簡易な実装で実現できる．また，これらのファイル自体や記載されている IP アドレス，User-Agent と Referer のリストを転用することで異なるフィッシングサイトに同様のアクセス妨害機能を持たせることができる．

Listing 2 PHP による User-Agent に対するアクセス妨害実装例

```
1 <?php
2   $userAgents = arrray("Bot","bot_test");
3   foreach($userAgents as $agent){
4     if ( strpos($_SERVER['HTTP_USER_AGENT'],$agent)
          !==false){
5       header("HTTP/1.0 404 Not Found");
6       exit();
7     }
8   }
9 ?>
```

4. アクセス妨害機能の動的解析手法の提案

本章では本研究における提案手法であるフィッシングキットが持つアクセス妨害機能の動的解析手法の詳細を述べる．

.htaccess によるアクセス妨害機能は.htaccess ファイルの記述フォーマットが決まっているため静的解析が可能である．しかしながら，PHP の場合は記述方法が多様である点や，難読化されている場合もあるため，静的解析が困難である．本研究では，静的解析が困難な PHP によるアクセス妨害機能も解析するために，インターネットに接続できない環境に配置した Web サーバに展開したフィッシングキットに対する動的解析によるフィッシングキットが持つアクセス妨害機能の解析手法を提案する．以降，インターネットに接続できない環境を閉環境と表記する．

フィッシングキットが有するアクセス妨害機能の動作条件を明らかにするために，閉環境の Web サーバに展開したフィッシングキットに対して動的解析を実施した．アクセス元 IP アドレスや，HTTP リクエストヘッダに含まれる User-Agent や Referer がアクセス妨害機能の主な対象である．本研究ではそのうちインターネット上のフィッシングサイトを調査の対象とした際に HTTP リクエストで設定可能であるという理由から User-Agent と Referer を解析対象とした．動的解析では，Web サーバに展開したフィッシングキットに対して User-Agent と Referer が異なる複数の HTTP リクエストを送信することで Web サーバからの応答を観測し，応答に含まれる HTTP ステータスコードを用いてアクセス妨害機能の有無を解析した．本解析の概要を**図 1** に示す．

まず，アクセス妨害機能の動作条件となりうる User-Agent と Referer を事前にフィッシングキットに含まれる .htaccess ファイルから抽出した．次に，閉環境の Web サーバに展開したフィッシングキットに対して，抽出した User-Agent と Referer を HTTP リクエストヘッダに付加して HTTP リクエストを送信し，その応答を記録した．

実在するフィッシングサイトを調査する際に，調査対象の Web サーバの負荷軽減のためにアクセス回数を最小化して調査の効率性を向上させる必要がある．そこで，アクセス妨害機能を動作させるための User-Agent と Referer のパターンを最小化する手法を用いる．アクセス妨害機能を動作させるために必要な User-Agent と Referer の最小のパターンを決定するため，フィッシングキットに対して異なる User-Agent と Referer を HTTP リクエストヘッダに付与して順に HTTP リクエストを送信し，その応答結果を解析する．本手法による User-Agent と Referer の最小パターンの決定例を**表 1** を用いて述べる．たとえば，表 1 の 1 行目はフィッシングキット A に対して “User-Agent: A”（以後簡単のため，UA_A と表記）を付与して HTTP リクエストを送信した場合に，“403 Forbidden” のステータスコードが応答されたことを示し，“User-Agent: B”，“Referer: Y”（以後簡単のため，Ref_Y と表記），“Referer: Z” を付与した HTTP リクエストを送信した場合には “200 OK” のステータスコードが応答されたことを示す．

図 1 閉環境の Web サーバに展開したフィッシングキットの動的解析の概要

Fig. 1 Overview of dynamic analysis for phishing kits deployed on web server located in the closed environment.

表 1 Web サーバに展開したフィッシングキットに HTTP リクエストを送信して得られた HTTP ステータスコード

Table 1 HTTP status code which is obtained from phishing kit when sending HTTP requests.

	UA_A	UA_B	Ref_Y	Ref_Z
フィッシングキット A	403	200	200	200
フィッシングキット B	403	200	200	403
フィッシングキット C	200	200	403	200

User-Agent と Referer の最小パターンの決定手法は以下の 3 つの手順で構成される．

- アクセス妨害されたフィッシングキットが最も多いパラメータを 1 つ目のパターンとして決定．
- 上記でアクセス妨害されたフィッシングキットは除外し，アクセス妨害されたフィッシングキットが次に多いパラメータを 2 つ目のパターンとして決定．
- すべてのフィッシングキットがアクセス妨害されるまで上記を繰り返し，試行回数を最小化する User-Agent と Referer のパターンを決定．

この手順を行うことで，最もアクセス妨害される User-Agent と Referer から順に選択することができ，アクセス妨害されたすべてのフィッシングキットを特定するパターンを得ることができる．表 1 で試すと，以下のようになる．なお，Referer を変更する場合の User-Agent には Internet

Explorer 11 が使用する文字列を用い，User-Agent を変更する場合の Referer は設定せずに実施した．

(1) UA_A が最もアクセス妨害機能を動作させたため，UA_A を 1 つ目のパターンとする．
(2) UA_A でアクセス妨害されたフィッシングキット A, B を除外し，次に多くアクセス妨害機能を動作させた Ref_Y を 2 つ目のパターンとする．
(3) 上記の結果，UA_A，Ref_Y を試行回数を最小化する User-Agent と Referer のパターンとする．

5. 調査手法

本章では本研究における調査手法の詳細を述べる．

5.1 全体の概要

本調査で用いた提案手法の全体概要を図 2 に示す．本研究の調査は大きく 3 つの段階に分かれる．まず，動的解析の対象となるフィッシングキットを収集するために，PhishTank [12] および OpenPhish [13] に掲載されたフィッシングサイトからフィッシングキットを収集した．

次に，4 章に示した提案手法を用いて，閉環境に配置した Web サーバにフィッシングキットを展開し，アクセス妨害機能を動作させる条件を動的解析により解析した．後続の調査において調査対象の Web サーバの負荷軽減や調査の効率化のために，アクセス妨害機能を動作させる条件を解析し，フィッシングサイトへのアクセス回数を最小化する．

最後に，インターネット上のフィッシングサイトのアクセス妨害機能を調査するために，アクセス妨害機能を動作させる条件を用いてインターネット上のフィッシングサイトにアクセスし，その結果を解析した．

5.2 フィッシングキットの収集

フィッシングキットはフィッシングサイトを構築するためのファイルがパッケージ化されたアーカイブファイルであり，攻撃者はこのアーカイブファイルを Web サーバ上で展開することでフィッシングサイトを構築できる．一部のフィッシングサイトでは，攻撃者の削除し忘れ等により，フィッシングキットのアーカイブファイルが残されたままになっていることがある．本調査では，PhishTank と OpenPhish に掲載されたフィッシングサイトからフィッシングキットである可能性が高いアーカイブファイルを収集した．

Web サーバ上でフィッシングキットが展開された場合，フィッシングキットのフォルダ名がそのままフィッシングサイトの URL のパス部に含まれることがある．また，フィッシングサイトの Web サーバの設定によりディレクトリリスティングが有効になっている場合，Web サーバ上にフィッシングキットが配置されていることを確認してフィッシングキットをダウンロードすることができる [5]．

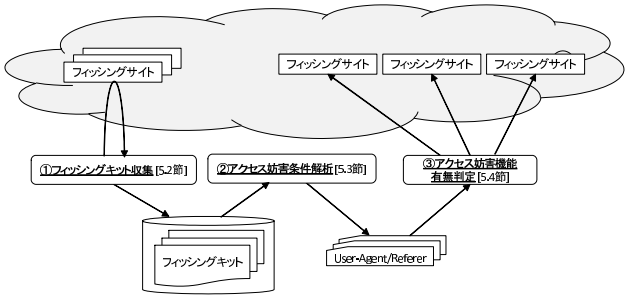


図 2 調査の全体概要
Fig. 2 Overview of the research.

図 3 フィッシングサイト URL パス部と展開後ファイルパスの比較によるフィッシングキット判定手法
Fig. 3 Detection method for phishing kits by comparing a path of phishing site URL with file paths of expanded file.

これらの手法でダウンロードしたアーカイブファイルにはフィッシングキットではないファイルも含まれる．そこで，入手したアーカイブファイルがフィッシングキットかどうかの判定を行い，フィッシングキットではないアーカイブファイルを調査対象から除外した（図 3）．具体的には，入手したアーカイブファイルの展開後のファイルパスが，フィッシングサイト URL のパス部と後方一致するかを順に確認し，ファイルパスと URL のパス部が後方一致した場合はフィッシングキットであると判定した．

5.3 閉環境でのフィッシングキットの動的解析

4 章に示した提案手法を用いて，以下の 2 つの解析を実施した．

(1) アクセス妨害を動作させるパターンの解析

収集したフィッシングキットを対象に 4 章の提案手法を用いてアクセス妨害機能を動作させるための User-Agent と Referer のパターンを解析した．5.4 節の調査では，本解析で得られた User-Agent と Referer のパターンを用いて実在するフィッシングサイトのアクセス妨害機能の調査を実施した．

(2) .htaccess と PHP によるアクセス妨害の解析

既存研究 [11] で明らかになっていない PHP によるアクセス妨害機能を調査するために，各フィッシングキットが

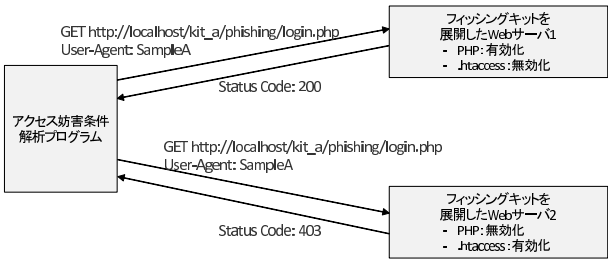


図 4 .htaccess と PHP によるアクセス妨害の解析環境
Fig. 4 Analysis environment of cloaking by .htaccess and PHP.

.htaccess ファイルと PHP のどちらによるアクセス妨害機能を持つかを解析した．図 4 のような.htaccess ファイルによるアクセス制御機能を有効にした環境と，PHP を有効にした環境にフィッシングキットをそれぞれ展開し，4 章の提案手法を用いて.htaccess ファイルと PHP によるアクセス妨害の割合を調査した．

5.4 実在するフィッシングサイトの応答調査

5.3 節で得られたアクセス妨害機能を動作させるための User-Agent と Referer のパターンを用いてインターネット上のフィッシングサイトにアクセスすることで，アクセス妨害機能を有するフィッシングキットがどの程度存在するかを調査した．

一般的にフィッシングサイトは動作期間が短く，ブラックリストに掲載されてから短い期間で閉鎖されることが多い [19]．本調査ではアクセスした時点でフィッシングサイトが閉鎖されていないかを確認するために，5.3 節で決定したパターンでアクセスする前に Web ブラウザ（Internet Explorer，Google Chrome，Firefox）が持つ User-Agent を付加した 3 種類の HTTP リクエストを事前に送信した．フィッシングサイトが動作中であることが確認された場合，5.3 節の結果で得られた User-Agent と Referer のパターンでのアクセスを継続する．

6. 調査結果

本章では，まずフィッシングキットの収集結果と.htaccess ファイルの解析結果を述べる．次に，フィッシングキットの動的解析結果と，アクセス妨害機能有無を判定するための User-Agent と Referer の解析結果について述べる．最後に，アクセス妨害機能を動作させるための User-Agent と Referer を用いてインターネット上のフィッシングサイトのアクセス妨害機能有無を調査した結果を述べる．

6.1 フィッシングキットの収集結果

PhishTank および OpenPhish に 2018/7/1–2018/10/31 の期間に掲載されたフィッシングサイトから収集したフィッシングキットの件数を表 2 に示す．239,320 件のフィッシングサイトにアクセスした結果，フィッシングサイトからダウンロードできたアーカイブファイルは 24,607 件であった．そのうち MD5 ハッシュ値ユニークなファイルは 6,887 件あり，さらに展開できたファイルは 6,811 件であった．展開に失敗した 76 件のファイルは，アーカイブファイルでなかったことやフォーマットエラーが原因であった．5.2 節のフィッシングサイトを構築したフィッシングキットかどうかを判定する手法により調査対象を抽出した結果，4,917 件が本研究において収集できたフィッシングキットであった．.htaccess ファイルのわずかな違いで同一のフィッシングキットを異なるものと判断していないことを確認するために，フィッシングキットから.htaccess を除外した状態で再圧縮してハッシュ値を計算し，ユニーク数を集計したところ同じく 4,917 件となった．図 5 に示すように，フィッシングキットには作成者と利用者が存在する．作成者はフィッシングサイトの機能だけではなく.htaccess ファイルも含めてフィッシングキットとしてパッケージ化し，利用者はパッケージ化されたフィッシングキットを用いてフィッシングサイトを構築しているものと考えられる．作成者が配布や販売をした時点でフィッシングキットは 1 つのパッケージとなっているため，.htaccess ファイルの有無にかかわらずハッシュ値のユニーク数は同じ件数になったものと考えられる．

表 2 フィッシングサイトから収集したフィッシングキットの概要
Table 2 Overview of collected phishing kits.

	件数
フィッシングサイト URL	239,320
アーカイブファイルダウンロード	24,607
MD5 ハッシュユニーク（展開失敗除く）	6,811
フィッシングキット	4,917

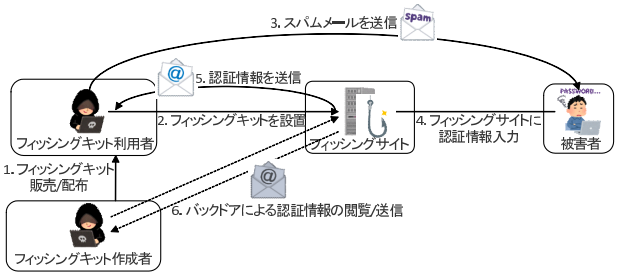


図 5 フィッシングのエコシステム
Fig. 5 Ecosystem of phishing attack.

6.2 .htaccess ファイルの解析結果

収集したフィッシングキットに含まれる.htaccess ファイルの解析を行った結果を表 3 に示す．複数の.htaccess ファイルを持つフィッシングキットがあり，6,435 件の.htaccess ファイルが確認され，MD5 ハッシュ値ユニー

表 3 .htaccess ファイルに記述されたアクセス妨害条件
Table 3 The number of cloaking conditions described in .htaccess files.

	件数（Hash ユニーク）
.htaccess ファイル数	6,435（207）
IP アドレス（ホスト名）	5,963（143）
User-Agent	3,212（87）
Referer	3,344（93）

クで見ると 207 件であった．最大で 113 件の.htaccess ファイルを持つフィッシングキットが存在したが，複数のディレクトリに同じ.htaccess ファイルを配置されており，ディレクトリごとに細かいアクセス制御をしているものではなかった．多くのフィッシングキットは.htaccess ファイルを 1 つだけ持ち，1 つのフィッシングキットが持つ.htaccess ファイル数の平均値は 1.88 件であった．

解析を困難にするために実装されたと考えられる，.htaccess ファイルの作成や変更を実施する処理が 2 件確認された．

（1）アクセス元 IP アドレスの.htaccess への追加機能

多くのフィッシングキットのアクセス妨害機能はフィッシングキット作成時に記述された条件によって動作するが，PHP のファイル書き込み関数を用いてアクセス元 IP アドレスを動的に追加するフィッシングキットが確認された．当該機能は Listing 3 のようなプログラムコードで記述されていた．Listing 3 は，コード内の 3–5 行目でアクセス元 IP アドレスからのアクセスを次回以降 PayPal 社の公式サイト（https://www.paypal.com）にリダイレクトさせるという記述を.htaccess ファイルに追記する．このプログラムコードは，解析者による詳細な解析を回避するための攻撃者の意図があると考えられる．

当該機能を有するフィッシングキットで構築されたフィッシングサイトは同一 IP アドレスでの 2 回目以降のアクセスは妨害される．アクセス妨害を回避するために，解析者はフィッシングサイト調査時にはアクセス元 IP アドレスを定期的に変更する必要がある．

当該機能を有するフィッシングキットには，アクセスしたユーザの HTTP ヘッダの “Accept-Language” に応じて表示言語を変更するような作り込まれた機能があることが確認された．フィッシングサイトと考えられる疑わしいウェブサイトの調査時に同様のアクセス妨害を受けた場合，そのフィッシングサイトは高機能なフィッシングキットが使用された可能性があるため，調査の優先度付けに活用できると考えられる．

なお，5.3 節の調査ではパーミッション設定により DocumentRoot 配下のフォルダで Apache によるファイルの書き込みを許可せず当該機能は動作しない状態で調査を実施したため，6.3 節の調査結果に影響はない．6.4 節の結果においても，2 回目以降すべてのアクセスが妨害された結果は存在しなかったため，当該機能の影響はなかった．

Listing 3 アクセス元 IP アドレスの追加機能

```
1 <?php
2 $file = fopen (".htaccess","a");
3 fwrite ($file, 'RewriteCond %{REMOTE_ADDR} ^'.
      $_SERVER['REMOTE_ADDR'].'$
4 RewriteRule .* https://www.paypal.com [R,L]
5 ');
6 fclose ($file);
7 ?>
```

Listing 4 .htaccess ファイルの動的生成

```
1 <?php
2 @copy ("_HIROn.txt",".htaccess");
3 ?>
```

表 4 .htaccess ファイルから抽出したアクセス妨害条件パターン数
Table 4 The number of cloaking condition patterns extracted from .htaccess files.

カテゴリ	User-Agent	Referer
件数	2,298	140

（2）.htaccess ファイル作成機能

フィッシングキットを展開した時点では.htaccess ファイルが存在せず，アクセス時に実行された PHP によって.htaccess ファイルが作成される機能が確認された．当該機能は Listing 4 のようなプログラムコードで記述されていた．_HIROn.txt というファイル名で.htaccess ファイルの内容が記述されたファイルがフィッシングキット展開時点からあり，このプログラムコードが実行されると.htaccess ファイルがコピーされる．よって，当該機能が実装されたフィッシングキットに対しては，アーカイブファイルからファイル名に基づいて.htaccess ファイルを抽出することができない．このような場合であっても，解析者は提案の閉環境の Web サーバに展開したフィッシングキットを動的解析する手法を利用することでこのような.htaccess ファイルを解析できる．

6.3 閉環境でのフィッシングキットの動的解析結果

表 4 にフィッシングキットが持つすべての.htaccess ファイルから抽出した User-Agent と Referer の件数を示す．これらの条件を用いて，閉環境の Web サーバに展開したフィッシングキットに対して User-Agent と Referer を付与した HTTP リクエストを送信し，応答を観測した．

（1）アクセス妨害機能有無判定パターンの解析結果

表 5 に本調査で収集したフィッシングキットのアクセス妨害機能有無を判定するために必要となる User-Agent /Referer パターンの解析結果を示す．本調査で収集した 4,917 個のフィッシングキットに対して，13 種の User-Agent/Referer パターンを持つ HTTP リクエストを送信

表 5 アクセス妨害機能有無を判定するための HTTP リクエストヘッダの条件一覧
Table 5 List of conditions of HTTP request header to determine presence of cloaking function.

No.	HTTP ヘッダ	パラメータ	件数	備考
1	User-Agent	Surfbot	551	ツール
2	Referer	spamcop.net	49	メールスパム対策サービス
3	User-Agent	imo-google-robot-intelink	19	Google
4	User-Agent	AdsBot-Google	5	Google アドワーズ
5	Referer	http://http://safebrowsing-cache.google.com/	2	Google Safe Browsing
6	User-Agent	ASPSeek	2	検索エンジンソフトウェア
7	User-Agent	HSFT - LVU Scanner	2	ツール
8	Referer	altavista.com	2	検索エンジン
9	Referer	google.com.ar	1	検索エンジン
10	User-Agent	CoolBot	1	ツール
11	User-Agent	DISCo Pump 3.2	1	ツール
12	User-Agent	NetZip Downloader	1	ツール
13	User-Agent	tor-exit	1	（不明）

表 6 アクセス妨害機能を持つフィッシングキットの調査結果
Table 6 Research result of phishing kits including cloaking functions.

カテゴリ	件数
アクセス妨害有り	636（12.9%）
.htaccess のみ	523（10.6%）
PHP のみ	55（1.1%）
.htaccess & PHP	58（1.2%）
アクセス妨害無し	4,281（87.1%）

表 7 アクセス妨害によるリダイレクト転送先
Table 7 Redirect forwarding destination by cloaking function.

No.	FQDN	フィッシング対象	件数
1	google.com	DropBox, Apple	3
2	yahoo.com	Paypal	1
3	www.linkedin.com	LinkedIn	1
4	www.paypal.com	Paypal	4
5	www.gov.uk	英国歳入税関庁	3
6	www.asb.co.nz	ASB Bank	1

することでアクセス妨害機能の有無を判定できることが分かった．

（2）.htaccess と PHP のアクセス妨害の解析結果

アクセス妨害機能を持つフィッシングキットの割合を**表 6** に示す．アクセス妨害機能を持つフィッシングキットは全体の 12.9%で，.htaccess によるアクセス妨害機能を持つフィッシングキットは 11.8%あることを確認した．一方で，PHP によるアクセス妨害機能が動作するフィッシングキットも 2.3%あることを確認した．この調査結果は，同一のフィッシングキットで構築されたフィッシングサイトにおいてアクセス妨害の条件が異なる場合に，PHP によるアクセス妨害のみが動作していた可能性や，攻撃者が.htaccess の設定を有効化できないサーバ環境であった可能性を類推することに寄与できると考えられる．

（3）フィッシングキットの動的解析から得られた知見

アクセス妨害された場合，HTTP ステータスコード “403 Forbidden” や “404 Not Found” を応答する場合がほとんどだが，一部のフィッシングキットで.htaccess の Rewrite 機能や PHP の header 関数を用いて正規の Web サイトへリダイレクトさせるものがあった．確認されたリダイレクト先の FQDN を**表 7** に示す．No.1, 2 は検索エンジンサイトである Google や Yahoo!にリダイレクトされた．No.3–6 はフィッシングサイトがターゲットとした正規の Web サイトにリダイレクトされた．No.3–6 のようなフィッシングサイトのターゲットとなる正規の Web サイトへの誘導はフィッシングサイト特有のクローキングであると考えられる．また，Drive-by Download 攻撃を行う Web サイトの場合，アクセスした被害者のプラグイン等の脆弱性有無を調査するためのクローキングがあるが，フィッシングサイトは脆弱性を悪用する攻撃ではないことからこのような機能は確認されなかった．

表 8 プログラミング言語やスクリプトで用いられる User-Agent をアクセス妨害したフィッシングキットの調査結果
Table 8 Results of cloaking condition of User-Agent used in programming languages and scripts.

User-Agent	件数	備考
Wget	522	wget コマンド
Curl	19	curl コマンド
Python-urllib	16	python
WinHttp.WinHttpRequest	15	VBA
WWW-Mechanize	10	Mechanize

表 8 にプログラミング言語のライブラリや OS のコマンドによって設定される User-Agent に関するアクセス妨害機能を調査した結果を示す．各 User-Agent ごとにアクセス妨害された件数を調査したところ，Wget が最も多くアクセス妨害されることが確認された．User-Agent を考慮

表 9 インターネット上に実在するアクセス妨害機能を持つフィッシングサイトの調査結果
Table 9 Research result of phishing site with cloaking functions existing on the Internet.

	件数
巡回 URL	12,076
フィッシングサイト生存 URL	4,901
アクセス妨害機能あり URL	511

せずにフィッシングサイトの解析ツールを実装した場合にアクセス妨害される可能性があるため，アクセス妨害される可能性が低い Web ブラウザが持つ User-Agent に変更する必要がある．

6.4 実在するフィッシングサイトの調査結果

OpenPhish に 2018/11/01–2018/12/11 の期間に掲載されたフィッシングサイトに対して，表 5 のアクセス妨害機能の有無を判定するパターンを付与した HTTP リクエストを送信し，調査した結果（2018/12/12 実施）を**表 9** に示す．巡回した URL 数は 12,076 件で，巡回時点で生存していたと考えられるフィッシングサイトは 4,901 件あった．そのうち，511 件（生存サイトに対して 10.4%）がアクセス妨害機能を持つフィッシングサイトであった．

フィッシングキットの動的解析結果（12.9%）と差異が発生した理由の 1 つとして，フィッシングキットには `.htaccess` によるアクセス妨害機能が実装されているが，フィッシングキットを展開した Web サーバで `.htaccess` を有効化する設定がされていなかった可能性が考えられる．`.htaccess` を有効化するためには Apache の設定ファイルの AllowOverride ディレクティブを変更する必要があるが，正規の Web サイトを攻撃者が改ざんした場合は Apache の設定ファイルを変更できない．巡回した URL のうちフィッシングキットを入手した 187 件から確認したところ，2 件のフィッシングサイトがフィッシングキットにアクセス妨害機能があるにもかかわらず，アクセス妨害されなかった．

7. 議論

本章では本研究での制約，今後の課題，研究倫理に関して述べる．

7.1 IP アドレスによるアクセス妨害の考慮

本研究では HTTP リクエストヘッダの User-Agent と Referer を変更して実在するフィッシングサイトのアクセス妨害機能有無を調査した．表 3 にあるように送信元 IP アドレスによるアクセス妨害機能のみが動作するフィッシングサイトは多く存在する．表 3 の送信元 IP アドレスによるアクセス妨害機能と，User-Agent または Referer によるアクセス妨害機能の件数を比較すると，今回の調査結果の 1.5 倍程度アクセス妨害機能を持つフィッシングサイトが存在することが考えられる．送信元 IP アドレスによるアクセス妨害の影響を軽減するためには，複数の IP アドレスからアクセスしてフィッシングサイトを解析する必要性が考えられる．

7.2 フィッシングキットの入口ページの選定手法

フィッシングキットには，ブラックリスト対策のためにフィッシングキットのエントリページ（`index.php`）にアクセスするたびにランダム文字列のサブディレクトリを作成し，そのサブディレクトリに転送するものが存在する [19]．5.2 節の閉環境での動的解析の対象とするファイルを決定するための手法において，ブラックリスト掲載 URL とファイルパスの後方一致を試行しているため，ブラックリスト掲載 URL にランダム文字列が含まれる場合に正しく決定できない．本研究では英数字 32 桁からなる文字列等，調査中に確認できたランダム文字列を後方一致での比較対象から除外することで対応した．フィッシングキットのさらなる調査により用いられるランダム文字列パターンの網羅性を高める必要がある．

7.3 アクセス妨害条件の抽出手法

本研究ではアクセス妨害機能の有無を判定するためのパターン抽出のために試行する User-Agent と Referer を `.htaccess` ファイルより抽出している．そのため，PHP のみで記述されたアクセス妨害の条件となる User-Agent と Referer のパターンは対象外となる．PHP によるアクセス妨害機能は `strpos` や `substr_count` 等の文字列を対象とする関数により実現される場合が多いため，動的解析によってこれらの関数の引数をロギングすることで PHP によるアクセス妨害機能で用いられる可能性のある User-Agent と Referer を抽出することが可能になると考えられる．

7.4 フィッシングキットにおけるバックドアとそのアクセス制御

フィッシングのエコシステムには図 5 のように，フィッシングキットの作成者と，フィッシングキットによってフィッシングサイトを構築する利用者が存在する．フィッシングキットの入手方法の 1 つとして，ダークウェブで購入する方法がある [20]．また，フィッシングキット作成者はフィッシングキットにバックドアを仕込むことで，フィッシングサイトが被害者から収集した認証情報をさらに窃取することが知られている [21]．バックドアを仕込んだフィッシングキット作成者はバックドアの存在を解析されないようにする目的としてもアクセス妨害機能を追加すると考えられる．

本調査において収集したフィッシングキットには被害者によって入力された認証情報をフィッシングキット作成

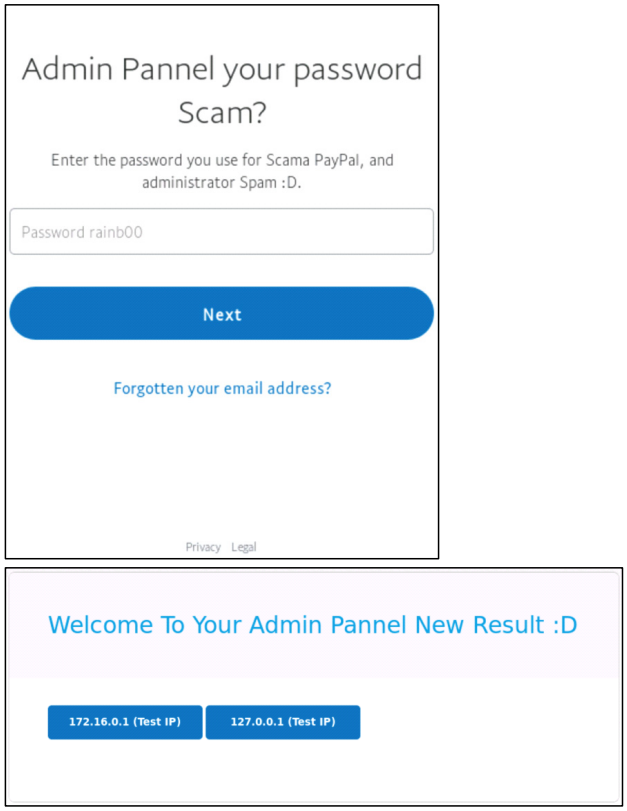


図 6 管理コンソール例
Fig. 6 Admin panel of phishing site.

者が閲覧するためと考えられる管理コンソールを持つものが確認された．**図 6** に確認された管理コンソールを示す．図 6 上は管理コンソールを表示するための認証画面で，図 6 下は被害者が入力した認証情報を閲覧するための一覧画面である．管理コンソールを持つフィッシングキットの中には特定の HTTP ヘッダを持つ場合に限りアクセスできるようなアクセス制御機能があり，そのアクセス制御機能も難読化され，解析を困難にするための処理が施されていた．

アクセス妨害機能を持つフィッシングキットはアクセス妨害機能を持たないものと比較して管理機能やバックドア機能を持つ可能性が高い．よってアクセス妨害機能を持つフィッシングキットの解析は，フィッシングキットの高度な機能やフィッシングのエコシステムの解明に役立つと考えられる．

7.5 研究倫理

フィッシングサイトのアクセス妨害機能の実態調査をするにあたって，実際のフィッシングサイトにアクセスすることは避けられない．よって本研究を実施するにあたって，研究倫理の観点から慎重に実験設計を行った．具体的には，正規のウェブサイトやホスティングインフラに対する悪影響を最小化する努力を行った．まず，調査対象となるウェブサイトは信頼性の高いブラックリストに掲載されているもののみを対象にしたため，フィッシングサイトである可能が高く，フィッシングサイトとは無関係のウェブサイトが含まれる可能性はきわめて低い．またアクセス妨害機能の調査において，無害なリクエスト送信とサーバ・インフラの負荷軽減の点に留意した．調査には限られた量の正常な HTTP リクエストを送信するだけであり，ウェブサイトの脆弱性を攻撃するリクエストは送信しない．ウェブサイトおよびホスティングインフラに対する負荷を軽減するために，1 つのウェブサイトに対するリクエストは 3 秒以上の間隔を開けて送信し，調査対象ウェブサイトはランダムに選定することで特定のホスティングインフラにアクセスが集中することを避けた．最後に，本調査で得られた知見を公表することで今後のフィッシングサイト対策の推進に貢献できるため公益性があると考える．

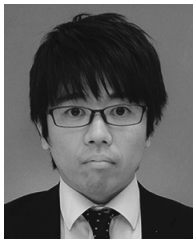
8. まとめ

本研究では，フィッシングサイトが持つアクセス妨害機能の解析手法を提案し，実際のフィッシングサイトを対象とした大規模な調査を実施した．閉環境の Web サーバに展開したフィッシングキットの動的解析により，アクセス妨害機能を持つフィッシングキットの割合と，アクセス妨害機能有無を特定するための User-Agent と Referer のパターンを解析した．解析の結果，収集したうちの 12.9%のフィッシングキットがアクセス妨害機能を有することが確認された．既存研究 [11] で明らかにされなかった PHP によるアクセス妨害機能を有するフィッシングキットは 2.3%あった．解析を困難にするアクセス妨害機能として，`.htaccess` ファイルにアクセス元 IP アドレスを動的に追加する手法と，`.htaccess` ファイル自体を動的に作成する手法があることを明らかにした．閉環境での動的解析から得られたアクセス妨害機能有無を特定するパターンを用いて，インターネット上のフィッシングサイトのアクセス妨害機能の有無を調査した．調査の結果，フィッシングサイトのうち 10.4%が User-Agent と Referer によるアクセス妨害機能が動作していることを明らかにした．アクセス妨害機能の対象となる User-Agent と Referer を避けることで，アクセス妨害機能を持つと確認された 10.4%のフィッシングサイトを解析できる．アクセス妨害機能によって解析者によるフィッシングサイト解析ができなくなる場合があるため，今後はアクセス妨害機能有無を特定するパターンを増やし，より網羅的にフィッシングサイトのアクセス妨害機能の実態を調査することが必要である．

参考文献

[1] Anti-Phishing Working Group: Phishing Activity Trends Report 4th Quarter 2016, available from ⟨https://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf⟩.
[2] トレンドマイクロ：「クラウド時代の認証情報」を狙いフィッシング詐欺が急増，2018 年上半期の脅威動向を分析，

入手先 ⟨https://blog.trendmicro.co.jp/archives/19461⟩.
[3] トレンドマイクロ：2018 年「個人」を狙う三大脅威：「フィッシング詐欺」, 入手先 ⟨https://blog.trendmicro.co.jp/archives/20138⟩.
[4] Thomas, K., McCoy, D., Grier, C., Kolcz, A. and Paxson, V.: Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse, *Proc. 22nd USENIX Security Symposium*, pp.195–210 (2013).
[5] Cova, M., Kruegel, C. and Vigna, G.: There Is No Free Phish: An Analysis of "Free" and Live Phishing Kits, *Proc. 2nd USENIX Workshop on Offensive Technologies* (*WOOT*) (2008).
[6] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J. and Zhang, C.: An empirical analysis of phishing blacklists (2009).
[7] Akiyama, M., Yagi, T., Yada, T., Mori, T. and Kadobayashi, Y.: Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots, *Computers & Security*, Vol.69, pp.155–173 (2017).
[8] Xiang, G., Hong, J., Rose, C.P. and Cranor, L.: Cantina+: A feature-rich machine learning framework for detecting phishing web sites, *ACM Trans. Information and System Security* (*TISSEC*), Vol.14, No.2, p.21 (2011).
[9] Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Mureddu, G., Ariu, D. and Roli, F.: Deltaphish: Detecting phishing webpages in compromised websites, *European Symposium on Research in Computer Security*, pp.370–388, Springer (2017).
[10] Mao, J., Tian, W., Li, P., Wei, T. and Liang, Z.: Phishing-alarm: Robust and efficient phishing detection via page component similarity, *IEEE Access*, Vol.5, pp.17020–17030 (2017).
[11] Oest, A., Safei, Y., Doupé, A., Ahn, G.-J., Wardman, B. and Warner, G.: Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis, *APWG Symposium on Electronic Crime Research* (*eCrime*), pp.1–12 (2018).
[12] PhishTank, available from ⟨https://www.phishtank.com/⟩.
[13] OpenPhish, available from ⟨https://openphish.com/⟩.
[14] Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A., et al.: Data breaches, phishing, or malware?: Understanding the risks of stolen credentials, *Proc. 2017 ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), pp.1421–1434 (2017).
[15] Zawoad, S., Dutta, A.K., Sprague, A., Hasan, R., Britt, J. and Warner, G.: Phish-net: Investigating phish clusters using drop email addresses, *eCrime Researchers Summit* (*eCRS*), pp.1–13 (2013).
[16] Oest, A., Safaei, Y., Doupé, A., Ahn, G., Wardman, B. and Tyers, K.: PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists, *2019 IEEE Symposium on Security and Privacy* (*SP*), Vol.40, pp.762–779, IEEE Computer Society (online), DOI: 10.1109/SP.2019.00049 (2019).
[17] Tsalis, N., Virvilis, N., Mylonas, A., Apostolopoulos, T. and Gritzalis, D.: Browser blacklists: The Utopia of phishing protection, *International Conference on E-Business and Telecommunications*, pp.278–293, Springer (2014).
[18] Chhabra, S., Aggarwal, A., Benevenuto, F. and Kumaraguru, P.: Phi. sh/$ ocial: the phishing landscape through short urls, *Proc. 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pp.92–101, ACM (2011).
[19] Han, X., Kheir, N. and Balzarotti, D.: Phisheye: Live monitoring of sandboxed phishing kits, *Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security* (*CCS*), pp.1402–1413 (2016).
[20] Check Point Software Technologies Ltd.: Tracking Down the [A]pache Phishing Kit, available from ⟨https://blog.checkpoint.com/wp-content/uploads/2018/04/Tracking_Down_Apache_Phishing_Brochure_Final.pdf⟩.
[21] McCalley, H., Wardman, B. and Warner, G.: Analysis of back-doored phishing kits, *IFIP International Conference on Digital Forensics*, pp.155–168, Springer (2011).



小寺 博和 （正会員）

1989 年生．2011 年早稲田大学基幹理工学部情報理工学科卒業．2013 年同大学大学院修士課程修了．同年エヌ・ティ・ティ・コミュニケーションズ（株）入社．2017 年日本電信電話（株）．サイバー攻撃対策技術の研究開発に従事．現在，NTT セキュアプラットフォーム研究所研究員．

芝原 俊樹

1989 年生．2012 年東京大学工学部機械情報工学科卒業．2014 年同大学大学院修士課程修了．同年日本電信電話（株）入社．以来，機械学習を応用したサイバー攻撃対策技術の研究開発に従事．現在，NTT セキュアプラットフォーム研究所研究員．

千葉 大紀

1988 年生．2011 年早稲田大学基幹理工学部情報理工学科卒業．2013 年同大学大学院修士課程修了．同年日本電信電話（株）入社．以来，サイバー攻撃対策技術の研究開発に従事．現在，NTT セキュアプラットフォーム研究所研究員．博士（工学）．電子情報通信学会，IEEE 各会員．

青木 一史 （正会員）

1981 年生．2004 年東北大学工学部情報工学科卒業．2006 年同大学大学院情報科学研究科修士課程修了．同年日本電信電話（株）入社．以来，サイバー攻撃対策技術の研究開発に従事．現在，NTT セキュアプラットフォーム研究所主任研究員．電子情報通信学会会員．

波戸 邦夫

1997 年東京工業大学工学部電気電子工学科卒業．1999 年同大学大学院総合理工学研究科物理情報工学専攻修士課程修了．同年日本電信電話（株）入社．NTT 情報流通プラットフォーム研究所にて IP-VPN，広域イーサネットの研究開発に従事．2016 年 NTT コミュニケーションズ（株）ネットワークサービス部テクノロジー部門担当部長．現在，日本電信電話（株）NTT セキュアプラットフォーム研究所サイバーセキュリティプロジェクトグループリーダ・主幹研究員．サイバー攻撃対策技術の研究開発に従事．電子情報通信学会会員．

秋山 満昭 （正会員）

2005 年立命館大学工学部卒業．2007 年奈良先端科学技術大学院大学情報科学研究科修士課程修了．同年日本電信電話（株）入社，NTT 情報流通プラットフォーム研究所にてマルウェア対策技術の研究開発に従事．2016 年 NTT セキュアプラットフォーム研究所特別研究員．2019 年同所上席特別研究員．主としてサイバー攻撃対策技術の研究開発に従事．博士（工学）．電子情報通信学会，IEEE 各会員．