Vol.2017-CSEC-78 No.38
Vol.2017-SPT-24 No.38
2017/7/15

仮名化による個人情報の保護に配慮したパブリッククラウド型フィッシングメール対応訓練システムの開発

東野 正幸[1,a)]　川戸 聡也[1,b)]　大森 幹之[1,c)]　川村 尚生[2,d)]

概要：近年，特定の組織を対象としたフィッシングメールが脅威となっている．フィッシングメールによる攻撃手段の 1 つとして，電子メールにより攻撃対象者を偽物のウェブサイトへ誘導し偽物のログイン画面にユーザ名やパスワードを入力させることで情報システムへ不正アクセスするための情報を収集する方法がある．この対策として，電子メールフィルタや侵入防止システムといった情報セキュリティシステムの導入に加えて，組織の構成員に対する情報リテラシ教育も重要であり，フィッシングメールに対する訓練が多くの組織で実施されている．攻撃対象者がフィッシングメールにより誘導されるウェブサイトを偽物であると判断するためには，ウェブサイトのドメイン名が正しいことの確認に加え，サーバ証明書も正しいことを確認する必要がある．より現実の攻撃に似た状況で訓練を実施して教育効果を高めるには，組織内よりも組織外のドメイン及びサーバ証明書を用いて訓練用のフィッシングサイトを運用することが望ましいと考えられる．しかし，多くのオープンソースソフトウェアのフィッシングメール対応訓練システムでは，サーバに訓練対象者のメールアドレスを保存する実装になっており，組織外のサーバに個人情報を保存することは運用リスクを高めてしまう．そこで本研究では，組織外のパブリッククラウドに配置可能なフィッシングメール対応訓練システムでありながら，訓練対象者の氏名，役職，メールアドレスといった個人情報は仮名化により管理主体を分離することで，訓練システムに係る情報セキュリティインシデント発生時に個人情報の漏洩対策を講じられるシステムを開発した．本稿ではそのシステムの開発について述べる．

キーワード：フィッシング攻撃，標的型攻撃，セキュリティ，セキュリティ教育

Masayuki Higashino[1,a)]　Toshiya Kawato[1,b)]　Motoyuki Ohmori[1,c)]　Takao Kawamura[2,d)]

1. はじめに

近年，フィッシングメールが脅威となっている [1], [2]．フィッシングメールとは，価値のある情報を攻撃対象者から奪い取ろうとする行動のうち，信頼されている人間やシステムに成りすまして送付される電子メールのことである．フィッシングメールによる攻撃手段の 1 つとして，電子メールにより攻撃対象者を偽物のウェブサイトへ誘導し偽物のログイン画面にユーザ名やパスワードを入力させることで情報システムへ不正アクセスするための情報を収集する方法がある．この攻撃の対策として，電子メールフィルタや侵入防止システムといった情報セキュリティシステムの導入に加えて，組織の構成員に対する情報リテラシ教育も重要であり，フィッシングメールに対する訓練が多くの組織で実施されている．

攻撃対象者がフィッシングメールにより誘導されるウェブサイトを偽物であると判断するためには，ウェブサイトのドメイン名が正しいことの確認に加え，サーバ証明書も正しいことを確認する必要がある．より現実の攻撃に似た状況で訓練を実施して教育効果を高めるには，組織内よりも組織外のドメイン及びサーバ証明書を用いて訓練用のフィッシングサイトを運用することが望ましいと考えら

[1] 鳥取大学 総合メディア基盤センター
Center for Information Infrastructure & Multimedia, Tottori University, 4-101, Koyama-Minami, Tottori, Tottori 680-8550, Japan
[2] 鳥取大学大学院 工学研究科 情報エレクトロニクス専攻
Department of Information and Electronics, Graduate School of Engineering, Tottori University, 4-101, Koyama-Minami, Tottori, Tottori 680-8550, Japan
a) higashino@tottori-u.ac.jp
b) t.kawato@tottori-u.ac.jp
c) ohmori@tottori-u.ac.jp
d) kawamura@ike.tottori-u.ac.jp

れる．

しかし，多くのオープンソースソフトウェアのフィッシングメール対応訓練システムでは，サーバに訓練対象者の氏名，役職，メールアドレス等を保存する実装になっており，組織外のサーバに個人情報を保存することは運用リスクを高めてしまう．

そこで本研究では，組織外のパブリッククラウドに配置可能なフィッシングメール対応訓練システムでありながら，訓練対象者の氏名，役職，メールアドレスといった個人情報は仮名化（pseudonymization）により管理主体を分離することで，訓練システムに係る情報セキュリティインシデント発生時に個人情報の漏洩対策を講じられるシステムを提案する．

2. 関連研究

2016 年に報告された DOGANA Project の調査 [3] によるとオンライン上ではフィッシング攻撃に関するツールが 48 件確認されている．この調査ではツールを使用目的ごとに分類し，それぞれの使用目的ごとに基準を設けてツールの評価を行っている．しかし，その評価にはツール自体のセキュリティに関する項目は含まれていない．

また，確認されているツールのうち，オープンソースソフトウェア（OSS）に該当し，かつフィッシングメール対応訓練に必要な攻撃の実行（TEAT: tools for the execution of the attack）機能と情報の集約とレポート（TIAR: tools for the information aggregation and reporting）機能の両方が利用可能なツールについて，訓練対象者の個人情報がどのように保存されているかはこの調査では明らかにされていない．このため，これらに該当する OSS のツールである Gophish[*1]，Phishing Frenzy[*2]，SPF (SpeedPhishing Framework)[*3]の機能を調査した結果，これらはサーバ上に訓練対象者の氏名，役職，メールアドレスといった識別特定情報を記録する実装となっていることが分かった．

フィッシングメール対応訓練を実施するためのツールとして OSS を採用することは，予算が限られた組織においては実施費用の削減できる可能性があるが，訓練用のサーバに組織の構成員の氏名，役職，メールアドレスといった識別特定情報を記録することは，訓練用サーバから個人情報の漏洩リスクを高めてしまうため容易には運用できない．しかしながら，訓練用のフィッシングサイトは組織外のドメインで稼働させなければ，フィッシングサイトを見分けるための技術的な説明に説得力を持たせることが難しくなる．

一方，本研究で提案するシステムは，組織外のパブリッククラウドに配置可能なフィッシングメール対応訓練システムでありながら，訓練対象者の氏名，役職，メールアドレスといった識別特定情報は仮名化により管理主体を分離することで，訓練システムに係る情報セキュリティインシデント発生時に個人情報の漏洩対策を講じられるシステムである点で従来のシステムとは異なり，サーバの運用や情報セキュリティ対策に関する高度な技術は多くを要求しない．

3. システムの設計

訓練の実施状況を把握するためには，訓練対象者が訓練用のフィッシングサイトに情報を入力したかどうかを調査する必要がある．ヒアリングやアンケートなどによる調査の場合，フィッシングサイトに引っかかる人はなぜ引っかかったのか把握できていない場合があり正確性の担保が難しい．また，将来的に組織の構成員全員に訓練を実施することが想定される場合には，正確性の担保の難しさだけでなく，調査に要する時間が膨大となる．このため，フィッシングメール対応訓練を安価に実施するためには訓練対象者の行動を自動的に記録する必要がある．

しかしながら，既存のオープンソースソフトウェアは，訓練対象者のメールアドレスやパスワードをサーバ上のデータベースに保存するタイプがほとんどであり，これを組織外のサーバで運用した場合，個人情報の漏洩対策が必要となり，情報セキュリティにおけるリスクが高まる．

そこで提案システムでは，仮名化（pseudonymization）により，識別特定情報と非識別非特定情報に情報を分離し，後者のみをサーバに記録する．ここで，識別特定情報とは「個人が（識別されかつ）特定される状態の情報（それが誰か一人の情報であることがわかり，さらに，その一人が誰であるかがわかる情報）」である．非識別非特定情報とは「一人ひとりが識別されない（かつ個人が特定されない）状態の情報（それが誰の情報であるかがわからず，さらに，それが誰か一人の情報であることが分からない情報）」である．これらの定義は文献 [4] による．仮名化の導入により，もしサーバから情報が漏洩した場合であっても，分離して保管されている識別特定情報を削除することで，サーバから漏洩した情報から個人を特定することが難しくなる．

4. システムの実装

提案システムはサーバ・アプリケーションとクライアント・アプリケーションの 2 つで構成される．サーバ・アプリケーションは訓練対象者の振る舞いを記録する機能を持つ．クライアント・アプリケーションは訓練用サイトへ誘導するメールを送付する機能を持つ．

4.1 サーバ・アプリケーションの実装

訓練用サイトと訓練対象者の振る舞いを記録するサー

[*1] Gophish: https://getgophish.com/
[*2] Phishing Frenzy: https://www.phishingfrenzy.com/
[*3] SPF (SpeedPhishing Framework): https://github.com/tatanus/SPF

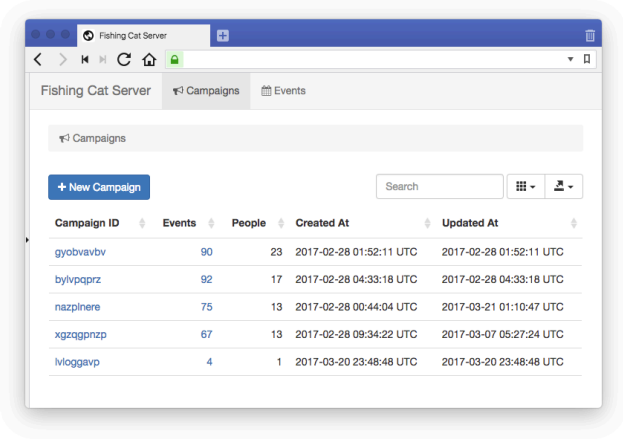

図 1　訓練一覧画面

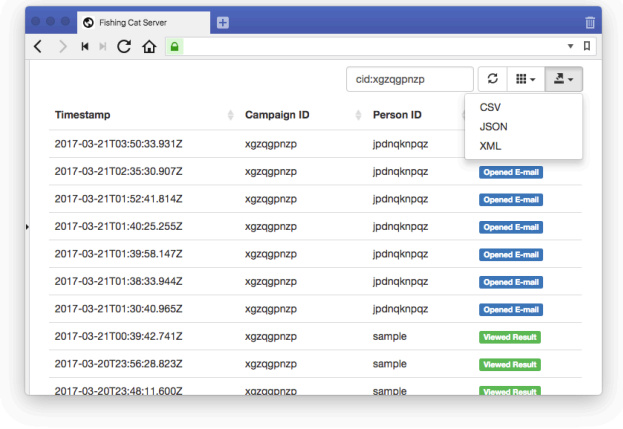

図 2　イベント管理画面

バ・アプリケーションは Ruby on Rails[*4]で実装した．

本サーバには，タイムスタンプ，訓練 ID（Campaign ID），訓練対象者 ID（Person ID），及び訓練対象者のアクションが記録される（図 1，図 2）．

アクションには，メールの開封（Opened E-mail），リンクのクリック（Clicked Link），情報の送信（Submitted Data），訓練結果画面の閲覧（Viewed Result）の 4 種類を定義する．それぞれのアクションの取得方法は下記のとおりである．

Opened E-mail　HTML メールに含める img タグの src 要素にサーバの URL を記述する．HTML メールを開封すると画像が表示され，その際にアクセスした URL により訓練対象者によるメール開封を判定する．URL には訓練対象者 ID を含めているため，誰がメールを開封したかを識別できる[*5]．

Clicked Link　訓練メールに記載した URL に HTTP による GET メソッドでアクセスした場合に記録される．

Submitted Data　訓練メールに記載した URL に HTTP による POST メソッドでアクセスした場合に記録される．

Viewed Result　HTTP による POST メソッドでアクセスした後にリダイレクトさせる URL にアクセスした場合に記録される．

上記のイベントは下記の URL により表現する．URL に含まれる `cid` は訓練 ID（Campaign ID），`pid` は訓練対象者 ID（Person ID）を表す．

- Opened E-mail `GET /images/:cid/:pid HTTP/1.1`
- Clicked Link `GET /forms/:cid/:pid HTTP/1.1`
- Submitted Data `GET /forms/:cid/:pid HTTP/1.1`
- Viewed Result `GET /results/:cid/:pid HTTP/1.1`

以上の情報には，訓練対象者の氏名やメールアドレスといった識別特定情報は含まれない．その代わりに，識別特定情報に依存せず生成された訓練対象者 ID（Person ID）を用いて仮名化データ（pseudonymized data）として保存する．これにより，標的型メール攻撃訓練を組織外のサーバで実施しながら，訓練用サイトにおける情報漏洩のリスクを低減させることが可能となる．訓練対象者 ID の生成は，後述するクライアント・アプリケーションで訓練メールを送付した際に生成するため，訓練対象者 ID とメールアドレスの紐付けはサーバでは無くローカルでのみ実施する．

この様に，訓練用サーバの運用組織と，訓練用メールの送信組織を分離可能にすることで，仮に訓練用サーバに情報セキュリティインシデントが生じた場合には，情報が分離されているため訓練対象者の識別及び特定を困難にできる．このため，安価にかつ効果的な標的型攻撃訓練を実施しながらも，訓練用サーバにかかる情報セキュリティインシデントのリスクを低減することが可能となる．

4.2　クライアント・アプリケーションの実装

訓練メールを送信するためのクライアント・アプリケーションは Ruby[*6]で実装した．訓練メールの本文は ERB（Embedded Ruby）によるテンプレートエンジンによりプレーンテキスト形式と HTML 形式のメールを生成できる．メールの送信には ActionMailer を使用した．ActionMailer は Ruby on Rails にも標準で組み込まれており Ruby のメール送信ライブラリとして使用実績が多数存在する．訓練メールの送信時にはそれぞれのメールアドレスに対して

[*4] Ruby on Rails — A web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern.: `http://rubyonrails.org/`

[*5] 最近の電子メールクライアントはリモートコンテンツをデフォルトではブロックすることが多いため，Opened E-mail アクションが記録されることは比較的少ない．しかし，信頼できない送信者からの HTML メールに含まれるリモートコンテンツはデフォルトでは表示しない設定にすることが情報セキュリティ的には好ましいと考えられるため，訓練対象者からこのアクションが検出された場合には，設定の変更を検討するための情報として有用である

[*6] Ruby Programming Language: `https://www.ruby-lang.org/`

全く関係しないランダムな文字列を訓練対象者 ID として生成及び付与し，その訓練対象者 ID をメールの本文中の URL に含めることで，訓練対象者のアクションを追跡するようにした．ランダムな文字列の生成には hashids[*7]を使用した．

5. システムの試運用

本システムの試運用のため本学の職員 35 名を対象に訓練を実施した．

5.1 訓練概要

本学では統一認証基盤により様々なサービスのシングルサインオンを実現している．また，学外からのアクセスに対しては多要素認証等の導入により安全性を確保している．統一認証のユーザ ID とパスワードが漏洩した場合には，機密性の高い情報のさらなる流出に繋がる可能性があるため，事前の訓練により教育・啓蒙を実施することで被害回避能力を組織的に高めることは重要である．そこで，標的型メール攻撃により統一認証のユーザ ID とパスワードが漏洩する状況を想定した訓練を実施した．本訓練は，訓練対象者に標的型メール攻撃を模倣した HTML メールを送付することで，訓練対象者を統一認証のログイン画面に模倣した訓練用のフィッシングサイトへ誘導し，統一認証のユーザ ID 及びパスワードを訓練用のフィッシングサイトのウェブフォームで送信させることで，機密情報が漏洩する状況を想定して実施した．実施期間は 2017 年 3 月 21 日（火）から 2017 年 3 月 31 日（金）までの 11 日間とした．訓練対象者は本学の職員 35 名とした．

5.2 サーバ・アプリケーションの運用

サーバ・アプリケーションの運用には Heroku[*8]の無料プランを使用した．無料プランでは 30 秒間アクセスがない場合にアプリケーションのインスタンスが停止する制限（再度アクセスすればインスタンスは自動的に復帰するが停止から起動のための時間がかかり停止時の初回アクセスのレスポンスが低下する．）や，1 ヶ月あたりの稼働時間の制限や，データベースに PostgreSQL を使用した場合にレコードの行数が 10,000 行までの制限がある。本システムでは 4 種類のアクションを記録することから，全員が全てのアクションを 1 回ずつ実行したとして無料プランでは最大で 2,500 人分まで記録できる．ただし，何度もアクションを実行する訓練対象者もいることや，全員がアクションを実行することはあまりないことから，1,000 人程度までであれば無料プランで実施可能であると考えられる．

5.3 クライアント・アプリケーションの運用

クライアント・アプリケーションによる訓練メールを送信するためのメールサーバには GMO インターネット株式会社のクラウドサービスである ConoHa[*9]を採用した．ConoHa は数クリックでメールサーバを配置可能であり，メールサーバを利用する場合の費用は日本（東京）リージョンのサーバでは 2017 年 4 月 17 日現在で 1 時間あたり 1.7 円となっている．このため，訓練メールを送信した後にメールサーバを削除することで，メールサーバ費用は数円程度で実施可能となる上，メールアドレスも自由に作成可能であり，訓練の柔軟性においても利点がある．

5.4 訓練結果

訓練対象者 35 名に本のうち 7 名が何らかのアクションを行った．うち 1 名はリンクのクリックのみにとどまり情報の送信は行わなかった．その他の 6 名は情報の送信を行い，17.1%にあたる訓練対象者が情報を送信した結果となった．今後は訓練対象者にヒアリングやアンケートなどを実施し，情報セキュリティインシデント発生時の対応方法が十分に認識されているか，調査を実施する予定である．

6. おわりに

本システムは，標的型メール攻撃やフィッシング攻撃の対応訓練を効果的かつ安全に，そして安価に実施可能な設計になっている．本システムのソースコードは `https://github.com/fishing-cat/` で公開している．ただし，現在は一般利用において必要なマニュアルの整備やエラー時の処理の実装が未完のまま残っている．今後は，本学以外の組織においても活用できるようにシステムとマニュアルを整備しその有用性を評価していく．

参考文献

[1] Anti-Phishing Working Group, Inc. (APWG): Phishing Activity Trends Report 4th Quarter 2016 (2017).
[2] フィッシング対策協議会ガイドライン策定ワーキンググループ：フィッシングレポート 2016 ―世界に広がるフィッシング対策の輪 ― (2016).
[3] Dambra, C., Gralewski, A., Frumento, E., Puricelli, R., Valentini, F., Mamelli, A., Russo, M., Weiss, N., Pacheco, B., Segou, O., Beaume, J. and Custodio, F.: Report on existing tools, their evaluation and the gap to be filled by DOGANA development, *Advanced Social Engineering and Vulnerability Assessment Framework*, DOGANA Project (2016).
[4] 技術検討ワーキンググループ：【資料 2-1】技術検討ワーキンググループ報告書，第 5 回パーソナルデータに関する検討会 議事次第 (2013).

[*7] Hashids - generate short unique ids from integers: `http://hashids.org/`
[*8] Heroku: Cloud Application Platform `https://www.heroku.com`
[*9] ConoHa: `https://www.conoha.jp/`