Vol.2019-MBL-93 No.20
Vol.2019-ITS-79 No.20
2019/11/21

組織間協働可能な標的型メール攻撃対応訓練システムの設計

東野 正幸[1,a)]

概要：教育機関・自治体・企業等において標的型メール攻撃が脅威となっている．標的型攻撃メールの内容は，巧妙化・多様化しており，一般のメール利用者がメールの内容だけで本物と偽物を見分けることは難しくなってきている．標的となった組織が，標的型攻撃メールを大量に送付されてしまうと，十数パーセントの構成員がアカウント情報を詐取されてしまう．これにより踏み台となったアカウントから，さらに他の組織へ標的型攻撃メールを連鎖的に送付していく攻撃手法により被害が拡大していくことから，単一組織だけでの対応が難しい問題となっている．このような攻撃に対処するには，複数の組織間で迅速に情報共有と対策を講じる協働体制が必要となる．そこで本研究では，ある組織が受け取った標的型攻撃メールを無害化・匿名化するとともに複数の組織が持つサーバ間で共有し，共有した情報を元に実際の標的型メール攻撃の内容に基づいた現実感を有する標的型メール攻撃対応訓練を日常的に実施可能とする分散型の情報セキュリティ教育システムを提案する．

キーワード：標的型メール攻撃，情報セキュリティ，クラウドサービス，分散システム

1. はじめに

教育機関・自治体・企業等において標的型メール攻撃が脅威となっている．標的型メール攻撃の手段の一つとしてフィッシングメール（phishing email）が良く用いられる．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 [1], [2], [3], [4], [5].

フィッシングメールを使用する攻撃者は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．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 [6], [7].

Confense, Inc. による 2017 年の調査によると，フィッシングメールの約 88%に悪意のある URL (uniform resource locator) が含まれており，フィッシングメールからフィッシングサイト等への誘導が行われているとされる [3]．このため，対応訓練による教育効果を高めるには，実際の攻撃者と同様の環境から訓練用のフィッシングメールを送付するとともに訓練用のフィッシングサイトへ誘導を行うことが有効であると考えられる．

また，フィッシングメールの内容は，巧妙化・多様化しており，一般のメール利用者がメールの内容だけで本物と偽物を見分けることは難しくなってきている．標的となった組織が，フィッシングメールを大量に送付されてしまうと，十数パーセントの構成員がアカウント情報を詐取されてしまう．これにより踏み台となったアカウントから，さらに他の組織へ標的型攻撃メールを連鎖的に送付していく攻撃手法により被害が拡大していくことから，単一組織だけでの対応が難しい問題となっている．このような攻撃に対処するには，複数の組織間で迅速に情報共有と対策を講じる協働体制が必要であると考えられる．

そこで本稿では，ある組織が受け取った標的型攻撃メールを無害化・匿名化するとともに複数の組織で共有し，共有した情報を元に作成した実際の標的型メール攻撃の内容に基づいた現実感を有する訓練を日常的に実施可能とすることで，継続的に情報セキュリティ教育を行うためのシステムの設計を提案する．提案システムにより複数組織に対する連鎖的な攻撃手法による標的型メール攻撃対策を効率的に実現可能となる．

[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
[a)] higashino@tottori-u.ac.jp

2. 関連研究

2016 年に報告された DOGANA Project の調査によると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 [8]．この調査では様々なツールを使用目的ごとに 4 つのカテゴリへ分類し種々の機能の有無について評価を行っている．しかし，その評価には複数の組織に対する連鎖的な攻撃手法に関する事項やフィッシングメールの無害化・匿名化に関する事項は含まれていない．

このため，この調査で挙げられているツールのうち，オープンソースソフトウェアに該当し，かつフィッシングメールを使った標的型メール攻撃の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ものとして，Gophish [9], Phishing Frenzy [10]，SPF (SpeedPhising Framework) [11]，Social Engineering Toolkit (SET) [12] の更なる調査を実施した．その結果，これらのツールは，ウェブサービスとして動作し，電子メールとウェブサイトをテンプレートから生成して訓練対象者に送付する機能等は有するものの，テンプレートとして使用した電子メールやウェブサイトを無害化・匿名化したり，複数の組織間で連携した訓練を行う仕組みは有していないことが分かった．

フィッシング対策に関する多数の研究が行われている [6], [7], [13] が，既存の実際の攻撃メールを関連組織と共有し，これらのメールを訓練用メールとして再利用するための設計は提案されていない．また，多くの研究者がフィッシング攻撃に対応するための教育を行う情報システムを提案している [14], [15], [16]．しかし，これらの提案システムは有効性を示しているものの，関連する組織間で連携することを考慮して設計されていない．

一方，本稿では，実際の攻撃に使用された電子メールを無害化・匿名化することで訓練メールを作成するとともに複数の関連組織で共有を行い，実際の訓練に使用すること想定しており，既存の訓練システムの設計に更なる機能拡張を加えるものである．

3. 機能要件

3.1 訓練メールを共有する機能

フィッシングメールを使った標的型メール攻撃対応訓練を行う場合，実際の攻撃に似たメールで訓練を行うことが効果的であると考えられる．また，関連する複数の組織に対して連鎖的に攻撃が行われる場合，関連先の組織に対して予め訓練メールを送付することができれば，より効果的な防御が行えると考えられる．しかし，実際の攻撃メールをテンプレートとして訓練用メールを作成し，他の組織と共有するためには，セキュリティとプライバシを考慮する必要がある．セキュリティに関しては，実際の攻撃メールの無害化が必要となる．また，プライバシに関しては、実際の攻撃メールが、どの個人や組織に宛てられた内容であるかを認識できなくする必要がある．

3.2 実際の攻撃と同じ環境の提供

フィッシングメールを使った標的型メール攻撃対応訓練の効果を高めるには，訓練用のメールの送信環境や訓練用のフィッシングサイトを実際の攻撃者と同様の環境で構築する必要がある．訓練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や IP アドレスが正しくないことやウェブサイトのサーバ証明書が正しくないことを認識する必要がある．しかし，仮に組織内の LAN 内などで訓練を行ってしまうとこれらを正しく認識する訓練にならない．このため，フィッシングサイトはインターネット上のパブリックサーバとして配置し，フィッシングメールは組織外の電子メールサーバーから送信する必要がある．

3.3 訓練用サーバからの情報漏洩の防止

標的型メール攻撃対応訓練に使用できるオープンソースソフトウェアの多くは，訓練を自動化するために，フィッシングメールを送信するための電子メールクライアントとフィッシングサイトの 2 つの役割を持っており，訓練対象者のメールアドレスや名前などの情報をサーバに保存する仕組みとなっている．しかし，前述のように訓練用のフィッシングサイトを組織外のパブリックサーバに配置するために，これらのサーバに訓練対象者のメールアドレスや名前を保存することは情報漏洩のリスクが大きくなる問題がある．また，多くの組織においては限られた予算で訓練を行う必要があるため，情報漏洩に関する情報セキュリティ対策のコストも大きくなる．このため，訓練に使用するパブリックサーバからの情報漏洩を防ぐために，訓練用のフィッシングサイトと訓練用の電子メールクライアントは，それぞれインターネットに配置されたパブリックサーバとプライベートネットワークにあるコンピュータに分割する必要がある．

3.4 組織間の独立性と分散化

標的型メール攻撃対応訓練に使用できるオープンソースソフトウェアの多くは複数の管理者向けアカウントを発行する機能を持っているため，1 つのサーバを複数の管理者で運用することは可能である．しかし，多くの組織ではそれぞれが独自の訓練実施計画を持っているため，訓練に使用するサーバはそれぞれ独立して管理・運用できる必要がある．そこで，提案システムでは，それぞれの訓練用サーバは独立して動作可能とし，関連組織ごとに分散したサーバ間において事前に連携の設定を行っておくことで，訓練

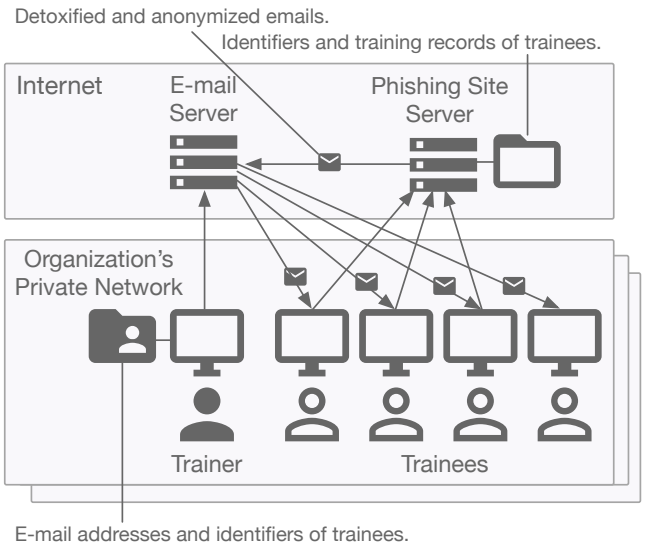

図 1　提案システムの概要
Fig. 1　The overview of our proposed anti-phishing training system.

用メールを連携先に配信して共有する方式とする．

4. システムの設計

4.1 概要

図 1 に提案システムの概要図を示す．訓練実施者（trainer）の計算機は，所属組織のプライベートネットワーク内に設置し，訓練対象者（trainee）のメールアドレスや名前などの機密データを持つ．一方，所属組織の訓練用のフィッシングサイトのサーバ（phishing site server）はインターネット上のパブリックネットワークに設置するが，機密情報は保有せず訓練対象者の訓練記録となる一般的なウェブサイトのアクセスログを記録する．アクセスログには仮名化 [17] により生成された訓練対象者を識別する識別子を付与する．識別子と訓練対象者と対応付けるデータを訓練実施者の計算機で保有することで，訓練用フィッシングサイトのサーバと訓練実施者の計算機の両方が攻撃を受けない限り，どの識別子がどの訓練対象者なのかを特定することは困難となる．なお，訓練用メールを送信する電子メールサーバ（email server）は，訓練メールの送信する際に生成し送信後は削除するため情報漏洩に繋がるリスクは小さい．

訓練用のフィッシングサイトのサーバには，それぞれの組織の訓練実施者が訓練用メールを登録することができる．訓練用メールは，実際に受信した攻撃メールから，攻撃先を特定する情報を取り除いて匿名化を行い，マルウェアや悪性 URL を訓練用のフィッシングサイトのさーばの URL に置換することが無害化を行う．こうして登録された訓練用メールは，他の関連組織へ共有することができ，関連する組織のいずれかが攻撃を受けたあとに連鎖的な攻撃を受ける前に他の組織で事前に訓練を行うことが可能となる．

4.2 動作

提案システムは以下の手続きで動作する．

(1) 訓練実施者は，訓練対象者のメールアドレスと訓練対象者を識別するランダムな識別子のリストを作成する．
(2) 訓練実施者は，実際の攻撃メールを匿名化・無害化することで訓練用メールを作成し，訓練用サーバに登録を行うとともに，関連する組織の訓練用サーバと共有を行う．
(3) 訓練実施者は，訓練対象者を識別する識別子を含んだ訓練用サーバの URL が記載された訓練用メールをそれぞれの訓練対象者に送付する．
(4) 訓練対象者は，訓練用メールを受信し，それぞれの所属組織で定められた情報セキュリティインシデント対応を実施する．

もし，訓練用サーバが攻撃を受け，訓練対象者のアクセスログが漏洩しても，アクセスログに記載されている識別子と対応する訓練対象者のメールアドレスや名前は，訓練実施者が使用するプライベートネットワーク内の計算機に保存されているため，識別子を削除することにより攻撃者は訓練対象者の個人情報を得ることは困難となる．

5. 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アクセスログを記録する機能を持つ．また，訓練用メールのテンプレートを保存すると共に他のサーバと共有する機能を持つ．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訓練メールを送付する機能を持つ．

5.1 サーバ・アプリケーション

サーバ・アプリケーションは Ruby on Rails [18] で実装する．訓練サーバは訓練の管理（**図 2**）及び訓練対象者のアクセスログの記録（**図 3**）を行う．アクセスログには以下に示す，組織 ID（*Organization ID*），訓練 ID（*Campaign ID*），人物 ID（*Person ID*），訓練対象者の振る舞い（*Action*），及びタイムスタンプ（*Timestamp*）が含まれる．これらの訓練サーバは組織ごとに独立して管理・運用を行う．

- *Organization ID* — 組織を識別する ID であり，サーバによりランダムに自動生成される．
- *Campaign ID* — 訓練を識別する ID であり，サーバによりランダムに自動生成される．
- *Person ID* — 訓練対象者を識別する ID であり，クライアントによりランダムに生成される．

また，訓練実施者はそれぞれの組織に応じた訓練用のフィッシングサイトを作成することができる（**図 4**）．

訓練対象者の振る舞いは，**表 1** に示すように訓練用サーバへのアクセス時に記録される HTTP（hypertext transfer

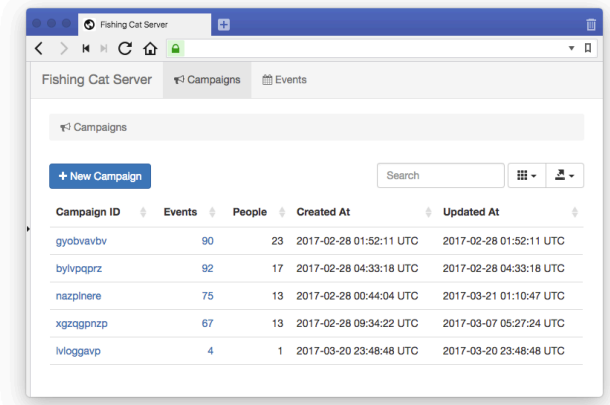

図 2　訓練用サーバにおける訓練の管理
Fig. 2　A list of campaigns on a phishing server.

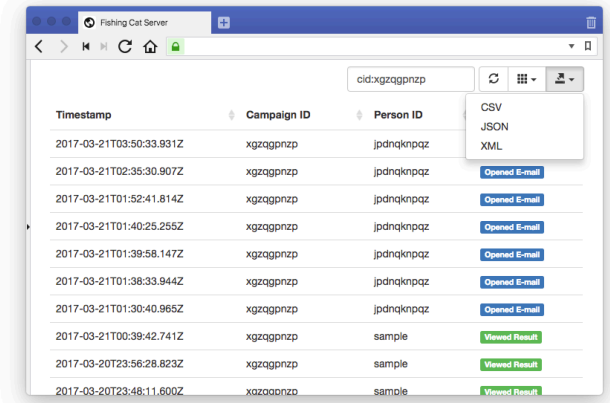

図 3　訓練用サーバにおける訓練対象者の振る舞いの記録
Fig. 3　A list of access records of trainees on a phishing server.

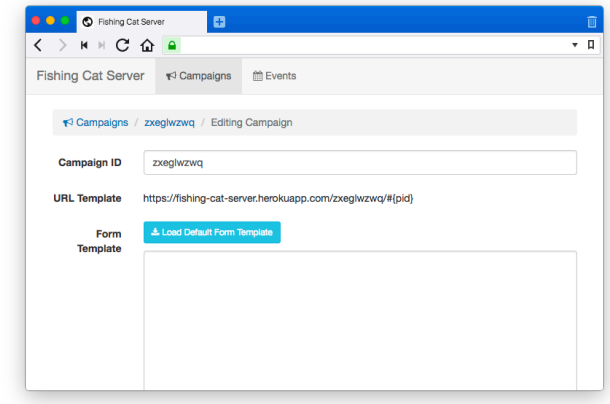

図 4　訓練用サーバにおけるフィッシングサイトの編集
Fig. 4　An edit page of a phishing page for training.

protocol）のメソッドと URL により分類される．

URL には *Organization ID* として`:oid`が，*Campaign ID* として`:cid`が，そして *Person ID* として`:pid`が含まれており，これらの ID により訓練対象者がどの組織のどの訓練にどのようなアクセスを行ったのかを判定する．

表 1　HTTP メソッド及び URL のパスと振る舞いの対応
Table 1　A map of an action, an HTTP Method, and a URL-Path.

Action	Method	URL-Path
Opened Email	`GET`	`/images/:oid/:cid/:pid`
Clicked Link	`GET`	`/forms/:oid/:cid/:pid`
Submitted Data	`POST`	`/forms/:oid/:cid/:pid`
Viewed Result	`GET`	`/results/:oid/:cid/:pid`

- *Opened Email* — 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．
- *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．アクセス先のウェブページにはウェブフォームを設置する．
- *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．本ウェブサイトが訓練用サイトであることの説明を記載する．

5.2　クライアント・アプリケーション

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 [19] で実装する．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する．ランダムな文字列の生成には [20] を使用する．

また，フィッシングメール対策訓練は継続的に実施することでフィッシングメールに対する対応方法の変化を定量的に評価し対策を講じることが有効と考えられる．このため，訓練をある程度は自動的に実施できるインタフェースにすることが望ましい．そこで，本ツールはコマンドラインツールとして実装し，crontab などのコマンドの定時実行スケジュール管理ツール等と組み合わせて使用することで継続的かつ自動的な訓練の実施にも対応できるように

する．

5.3 フィッシングメールの無害化と匿名化

実際の攻撃に使用されるフィッシングメールには悪性URLが含まれている可能性がある．このため，訓練実施者はこの悪性URLを訓練用フィッシングサイトのURLに置き換えることで無害化を行いつつ訓練に転用可能なメールとする．また，マルウェアなどのファイルが添付されている場合はこれを除去する必要がある．

また，フィッシングメールに攻撃先を特定する固有名詞や機密性を持つ内容が含まれている場合は，他の組織での訓練では使うことができない．このような匿名化は，文脈に依存する場合も多く，機械的に匿名化することは難しい．このため，訓練実施者が最終的には匿名化されたことを確認した上で，システムで共有することとする．

6. おわりに

本稿では組織間協働可能な標的型メール攻撃対応訓練システムの設計の提案を行った．提案システムにより，ある組織が受け取った標的型攻撃メールを無害化・匿名化するとともに複数の組織が持つサーバ間で共有することで，共有した情報を元に実際の標的型メール攻撃の内容に基づいた現実感を有する標的型メール攻撃対応訓練を日常的かつ安価に実施可能となる．今後は，提案システムを実装し複数の組織と連携して有効性を評価する．

参考文献

[1] Anti-Phishing Working Group (APWG), Inc.: Phishing Activity Trends Report 2nd Quarter 2018, Technical report, Anti-Phishing Working Group (APWG), Inc. (2018).
[2] KnowBe4, Inc.: 2018 Phishing By Industry Benchmarking Report, Technical report, KnowBe4, Inc. (2018).
[3] Cofense, Inc.: Enterprise Phishing Resiliency and Defense Report, Technical report, Cofense, Inc. (2017).
[4] Cofense, Inc.: Enterprise Phishing Susceptibility and Resiliency Report, Technical report, Cofense, Inc. (2016).
[5] TrendLabs APT Research Team: Trend Micro Incorporated Research Paper 2012 Spear-Phishing Email: Most Favored APT Attack Bait, Technical report, Trend Micro, Incorporated. (2012).
[6] Aleroud, A. and Zhou, L.: Phishing environments, techniques, and countermeasures: A survey, *Computers & Security*, Vol. 68, pp. 160–196 (online), DOI: 10.1016/j.cose.2017.04.006 (2017).
[7] Al-Daeef, M. M., Basir, N. and Saudi, M. M.: Security Awareness Training: A Review, *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2017*, International Association of Engineers (IAENG), Newswood Limited, pp. 446–451 (2017).
[8] 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).
[9] Wright, J.: Gophish - Open-Source Phishing Framework, , available from ⟨https://github.com/gophish⟩ (accessed October 18, 2019).
[10] Geek, P.: Phishing Frenzy: Ruby on Rails Phishing Framework, , available from ⟨https://github.com/pentestgeek/phishing-frenzy⟩ (accessed October 18, 2019).
[11] Compton, A.: SPF (SpeedPhish Framework), , available from ⟨https://github.com/tatanus/spf⟩ (accessed October 18, 2019).
[12] Kennedy, D.: The Social-Engineer Toolkit (SET), , available from ⟨https://github.com/trustedsec/social-engineer-toolkit⟩ (accessed October 18, 2019).
[13] Gupta, B. B., Tewari, A., Jain, A. K. and Agrawal, D. P.: Fighting against phishing attacks: state of the art and future challenges, *Neural Computing and Applications*, Vol. 28, No. 12, pp. 3629–3654 (online), DOI: 10.1007/s00521-016-2275-y (2017).
[14] Wash, R. and Cooper, M. M.: Who Provides Phishing Training?: Facts, Stories, and People Like Me, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 492:1–492:12 (online), DOI: 10.1145/3173574.3174066 (2018).
[15] Wen, Z. A., Lin, Z., Chen, R. and Andersen, E.: What.Hack: Engaging Anti-Phishing Training Through a Role-playing Phishing Simulation Game, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 108:1–108:12 (online), DOI: 10.1145/3290605.3300338 (2019).
[16] Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M. A. and Pham, T.: School of Phish: A Real-world Evaluation of Anti-phishing Training, *Proceedings of the 5th Symposium on Usable Privacy and Security*, pp. 3:1–3:12 (online), DOI: 10.1145/1572532.1572536 (2009).
[17] Pfitzmann, A. and Hansen, M.: A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management (2010).
[18] Hansson, D. H.: 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., , available from ⟨https://rubyonrails.org/⟩ (accessed October 18, 2019).
[19] The Ruby Community: Ruby Programming Language, , available from ⟨https://www.ruby-lang.org/⟩ (accessed October 18, 2019).
[20] Akimov, I.: Hashids - generate short unique ids from integers, , available from ⟨https://hashids.org/⟩ (accessed October 18, 2019).