Marchal, Samuel; Asokan, N.

On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World

Published in:
11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)

Published: 01/01/2018

Document Version
Peer reviewed version

Please cite the original version:
Marchal, S., & Asokan, N. (2018). On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World. In *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)* USENIX : THE ADVANCED COMPUTING SYSTEMS ASSOCIATION.
https://www.usenix.org/conference/cset18/presentation/marchal

On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World

Samuel Marchal
Aalto University
samuel.marchal@aalto.fi

N. Asokan
Aalto University
asokan@acm.org

Abstract

While a plethora of apparently foolproof detection techniques have been developed to cope with phishing, it remains a continuing problem with an increasing number of attacks and victims. This is due to a gap between the reported experimental detection accuracy of solutions from the academic literature and their actual effectiveness in real-world scenarios. For instance, design choices made while only considering how to maximize the accuracy of phishing detection sometimes has the unintended effect of constraining deployability or limiting usability. We hope to raise awareness about practices causing this gap and present a set of guidelines for the design and evaluation of phishing webpage detection techniques. These guidelines can improve the effectiveness of phishing detection techniques in real-world scenarios and foster technology transfer. They also facilitate unbiased comparison of evaluation results of different detection techniques.

1 Introduction

Phishing attacks deceive their victims into revealing sensitive information predominantly using phishing webpages (phish) [2, 10] that mimic the content of legitimate websites. Over the past decade, numerous techniques have been developed to detect phishing webpages [9, 15, 20]. Searching for "phishing" in Google Scholar paper titles shows that it is half as popular as "malware", matching over 3,800 results. Although some proposed techniques [15, 21] report stellar performance in evaluation setup ($> 99.9\%$ accuracy), no definitive solution has seen widespread adoption. The number of reported attacks is increasing [19] and phishing is successfully used as support for new malicious activities like distributing ransomware [2], showing that current protection is insufficient. Hence, we can observe a gap between high accuracy reported by detection techniques from the literature and the actual state of increasingly successful phishing attacks. We claim that this phenomenon is due to two major reasons. 1) Typically the only criterion during the design phase is to achieve detection accuracy levels that are higher than previous academic work. Equally important deployability and usability considerations are often ignored. 2) The evaluation methodology is often not representative of real-world scenarios, thus failing to assess the actual effectiveness of the proposed method.

Although there have been several comparative surveys on phishing detection [9, 12], they only compare the methods used (e.g. machine learning algorithms, features, etc.) and their detection accuracy. They identify the scope of the methods, list theoretical strengths and weaknesses, and analyze the hypothetical resilience to circumvention. However, none of these surveys evaluate ease of deployment or impact on usability. Moreover, performance comparison is biased by the absence of a standardized evaluation protocol, which makes it difficult to compare different schemes reliably.

In contrast, in this paper we tackle the problem of ensuring the effectiveness of a phishing detection technique in real-world scenarios. We consider effectiveness as a combination of *detection performance*, *temporal resilience*, *deployability* and *usability*. We point out practices to avoid and provide recommendations on the design and implementation of phishing detection techniques (Section 2). We provide guidelines for ground truth dataset composition (Section 3) that ensure representativity of evaluation results. We introduce a standardized evaluation protocol (Section 4) that 1) assesses the suitability of phishing detection techniques for real-world deployment and 2) eases comparison with state-of-the-art techniques.

These recommendations relate to the design and evaluation of machine learning-based phishing webpage detection techniques, which represent the main focus of the literature [6, 15, 12, 20, 21]. We formulate these recommendations based on our experience in designing,

implementing and evaluating the accuracy and usability of phishing detection techniques [15, 16, 17, 18]. This experience is supplemented by technology transfer discussions we had with major security vendors including McAfee, Huawei and F-Secure.

2 Design and Implementation

Apart from detection performance, which has been extensively evaluated and compared [9, 12], we discuss the strengths and weaknesses of several design and implementation alternatives regarding *deployability* and *usability*.

2.1 Detection Technique Implementation

Two major components can "uniquely" define a webpage, its *content* and its pointer: a *URL*. Most phishing detection techniques analyze one of these components to render their decisions. This analysis is done either in real-time, every time a webpage is visited, or offline to compose *blacklists*.

Webpage content techniques rely on the analysis of the information contained in the webpage after being loaded and rendered in a browser. Features are computed from this content, and they can be augmented with external information, e.g., search engine data [21, 23]. Machine learning techniques are applied to features extracted from this data to decide whether it is a phish or not.

Pros: The decision is based on the exhaustive analysis of the webpage content that is actually depicted in the browser. This class of techniques currently has the best accuracy [18, 21, 20]. It is resilient to many circumvention techniques including adaptive attacks that would serve different content at different times to different users while having the same pointer (URL).

Cons: Loading the content of a malicious webpage can harm a device if the page contains, e.g., malicious javascript code, or if the link points to a drive-by download. Feature extraction from webpage content requires many interactions with the browser. Thus, integration of these techniques to the large diversity of available browsers can be cumbersome. The computation of a large number of features, especially external, can be time consuming and computationally expensive.

URL-based techniques analyze the composition of a URL to identify in real-time whether it points to a phish or not, e.g., [13]. The analysis can be augmented with external information such as Alexa website ranking [18], DNS information [20, 21] and semantic information [17].

Pros: The decision depends only on the analysis of the URL, preventing malicious content from being loaded in the browser. It is usually fast because only a few features need to be computed. It requires limited interaction with the browser, only to extract the URL, which eases integration.

Cons: URLs only provide limited information to analyze, impacting phishing detection accuracy negatively [13, 17]. The analysis of the URL only does not guarantee that the content it points to is safe: the URL can remain the same and the content can be changed at will by loading it dynamically or using different link redirection chain.

Blacklists list URLs pointing to probable phishing websites. Every time a link is clicked or typed, it is checked against the list and the connection is prevented if the link is found on the list. Blacklist composition relies on the analysis of webpage content pointed by a URL. In contrast to *webpage content* methods that compute a decision every time a webpage is visited, blacklists use centralized web crawlers that fetch the page content only once, compute the decision and add the URL to the blacklist accordingly. Most current phishing detection techniques are implemented in this manner, e.g., Google's Safe Browsing [10].

Pros: As with URL methods, no content is loaded in the browser, preventing infection and easing integration. Phishing detection performance is high as for *webpage content* methods [10] but decision is faster.

Cons: A delay of several days is observed between the availability of a content on the Web and its pointing URL being analyzed and added to a blacklist [10]. During this time, users remain unprotected. Blacklists are composed based on the analysis of several features, most of them referring to the pointed webpage. As for URL methods the pointed content can change over time, letting outdated entries in the blacklist. Also, phishers can easily detect automated crawlers used during blacklists composition and dynamically serve a legitimate content to those.

2.2 System Design

Two system design approaches can be chosen to implement a phishing detection system: *centralized* or *client-side*. This choice is usually driven by the requirements of the detection technique regardless of the impact it has on usability.

Centralized implementations involve two parts. A client software component is installed on the user device. It sends requests to a centralized service where all or part of the processing needed to render the decision is carried out and the decision is returned to the requesting client.

Pros: It enables the use of distributed computing and storage. Detection techniques can use a large amount of (external) data and have heavy computational requirements without impacting or being limited by client de-

vice performance. Updates to the detection model and addition of new features are easy to manage since they are implemented in one single location.

Cons: Users must share part of their browsing information, i.e., the reference for analysis: URL or webpage content, that must be analyzed by the centralized service to render the decision. This endangers privacy. Requesting a distant service implies an additional communication delay in decision response.

Client-side solutions require only a piece of software to be installed on the client machine. The decision is computed without relying on external sources.

Pros: Users do not have to share any browsing history, preserving their privacy. Client-side solutions render faster decisions since no unnecessary communication with a remote service is required. The availability of the service is guaranteed.

Cons: They admit only lightweight detection techniques due to performance limitation of the client device. Nevertheless, it causes a computation overhead on the client device that may impact user experience. The requirement to install software (e.g., browser extension) on the client can also be a major limitation in certain settings.

URL blacklists require a centralized architecture for webpage analysis and blacklist composition, although the blacklist can be locally stored and updated. *URL* and *webpage content* based methods admit a fully client-side implementation if 1) they do not use external information i.e. other than webpage content and 2) their computation is lightweight. URL methods tend to have low accuracy, leading to false alarms [13, 17], and are hence not useful by themselves. With their dependency on a centralized architecture and the several days of delay, URL blacklists have several drawbacks.

From a theoretical perspective, webpage content techniques with a client-side implementation present the best trade-off, guaranteeing high detection accuracy and preserving privacy. They raise some constraints on the detection technique design that must provide lightweight computation and sufficient speed to act in real-time (less than 1 second [18]) without impacting client device performance. Nevertheless, a centralized solution is usually favored by security vendors for two main reasons: 1) the detection solution is easier to maintain and update and 2) any impact on user device performance is usually imperceptible. This choice may be reconsidered in the near future though, considering the increasing attention given to privacy and the corresponding legal measures taken to protect it, e.g., EU General Data Protection Regulation (GDPR) [7].

3 Ground Truth Collection

Comparing the accuracy of phishing detection techniques is challenging due to limited reproducibility of the results. Detection algorithms are often considered as a competitive advantage and not made publicly available. Academic publications often lack detailed description of the proposed methods, which hinders reproducibility. Evaluation is done on different datasets and do not present the same accuracy metrics.

To address these limitations we introduce several recommendations in this section for ground truth data collection and propose a standardized evaluation methodology in Section 4. We list best practices to ensure 1) effectiveness of phishing detection methods in real-world applications and 2) unbiased and relevant comparison of accuracy between different phishing detection techniques. The recommendations primarily address the use-case of supervised machine learning based phishing detection [18, 20, 21]. Some of the recommendations are generally applicable for the evaluation of any machine learning technique, while the rest are specific to phishing detection.

3.1 Selection Process

Reference ground truth datasets exist to evaluate phishing detection techniques. However, they contain mostly outdated entries because of phishing websites having a short lifetime [10], which prevents availability of their information over time. Rather than having a static reference dataset, it is better to focus on reference sources providing an evolving but consistent dataset with up-to-date instances that ensure current representativity. We identified several dataset selection practices to follow in order to ensure the representativity of evaluation results with respect to real-world phishing detection scenarios.

3.1.1 Legitimate dataset selection

1. Select webpages developed in multiple languages and alphabets. Phishing detection methods rely largely on lexical anlaysis of URLs and content [18, 21, 23]. With the possibility to use or encode unicode characters in both content and URLs, evaluating methods on webpages developed in, e.g., the Roman alphabet do not assess their efficacy on webpages developed in Chinese, Japanese or Cyrilic alphabet. Around 50% of webpages have English content on the Internet, followed by German (6%), Russian (6%), Spanish (5%) and Japanese (4%)[1]. We observed that accuracy results obtained from

[1] W3Techs - `https://w3techs.com/technologies/overview/content_language/all` (accessed 06/22/2018)

evaluation of phishing detection technique depend on the languages and alphabets of the webpage analysed [15].

2. Select webpages of diverse popularity. Do not limit the dataset to high popularity websites. Features of many classification techniques [18, 20, 21] rely on popularity directly (e.g. Alexa ranking) or indirectly (e.g. DNS information). Hence, a dataset must not be biased by this factor where legitimate instances will have only high popularity and phishing webpages obviously low. Most low popularity websites are legitimate and such instances must be represented in the legitimate set. Hence, it is not recommended to take, e.g., Alexa as the source for legitimate instances as observed in some work [5]. We recommend to use a balanced legitimate set comprising 50% high popularity websites and 50% low popularity websites. High popularity websites should represent services most targeted by phishing attacks, i.e., online banking, online gaming, payment service, email provider, retail websites, etc.
3. Include URLs representative of user requests. For example, sources like Alexa list [5] or DMOZ [14] contain only Fully Qualified Domain Names without any path, e.g, *www.example.com*. While surfing, users visit many long links that do not follow this pattern. A classification model can be biased towards this criteria and render good results during evaluation while performing poorly in reality. We recommend recording websites visited by users while surfing to gather a representative legitimate dataset. This needs to be manually sanitized to remove any sensitive user information and ensure that no malicious webpage is included.

Most of the aforementioned recommendations relate to the representativity of datasets. As illustrated in Figure 1, the representativity of a dataset impacts the accuracy of a detection technique and its evaluation results. If a detection technique is trained using a non-representative dataset, it may be ineffective at detecting phishes that are not represented in the dataset. Nevertheless, the evaluation may report good accuracy because of using the same type of non-representative data.

3.1.2 Phishing dataset selection

1. Compose a dataset from publicly available sources that update their entries on a regular basis. Even though listed elements evolve over time, the same source will provide the same diversity and consistency of data, which ensures reproducibility. These sources also provide freshly detected phishes, which represent the latest trend in techniques used for creating phishing webpages . The alternative of using a static dataset is not a good approach since most phishing websites go offline after a few hours [10]. Examples of recommended sources for phishing webpages are PhishTank[2], used for many phishing detection evaluations [17, 18, 23], or OpenPhish[3]. Both sources maintain up-to-date lists of active phishing websites that are refreshed on a regular basis. We recommend to scrape websites listed on these sources in an automated manner and on a regular basis to create offline caches of the phishes. The caches must contain all the necessary information to evaluate the detection technique and to perform data sanitization.
2. Manually sanitize the dataset to ensure it is composed of valid, active phishes. The main challenge is to get accurate phishing labels. Even though websites are marked as phish, this labeling can be erroneous or outdated, i.e., a URL can be marked as phish while its content is not available or not malicious anymore. As later discussed in Section 3.2, many phishing blacklist entries from reliable sources are not malicious and either point to empty content, domain parking websites or legitimate websites. This highlights the need for data sanitization. We introduce a protocol for sanitizing phishing datasets in Section 3.2.
3. Ensure that the dataset does not contain too many replicas. The same websites are often used in sev-

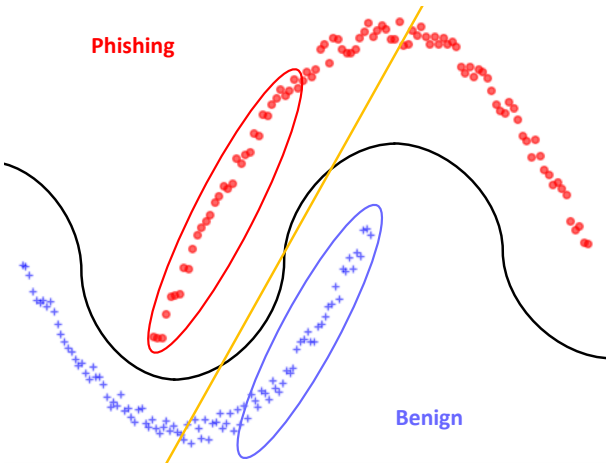

Figure 1: Simplistic illustration of dataset representativity and its impact on phishing detection models. The non-representativity of a dataset (points selected in ovals) leads to building a wrong decision boundary for the phishing detection model (orange line). A good representativity (the remaining points) would lead to building an appropriate boundary as represented by the black line.

[2]PhishTank - `https://www.phishtank.com/`
[3]OpenPhish - `https://openphish.com` (accessed 06/22/2018)

eral phishing campaigns and pointed by slightly different URLs. Including several variants of the same phish can bias evaluation results by overestimating the detection capabilities and the scalability of a given detection technique, e.g., by having replicas in the training set and the testing set. Splitting the phishing dataset chronologically (for training and testing) mitigates the impact of replicas on evaluation results.

4. Select phishes that target a large variety of websites and services to ensure representativity. The same languages and alphabets diversity considerations as for the legitimate dataset apply to the phishing dataset.

Dataset composition is tied by many constraints that may introduce a bias. The removal of personal information from the legitimate dataset reduces its representativity, considering that many links and webpages contain personal information related to, e.g., user account or tracking information. Using public blacklists to compose the phishing dataset may deprive it of short-lived phishing websites. An alternative to get such instances is to scrape a large amount of websites that would be further checked against blacklists to determine if they have been malicious at scraping time. Another solution is to mount honeypot web servers that intend to be compromised by phishers [10] and capture the phishing websites that will be hosted on honeypots. This assumes prior knowledge of the detection method to collect all the information needed to render the decision. It prevents though the usage of the same dataset to assess other techniques.

3.2 Phishing Dataset Sanitization

Composing a ground truth phishing dataset from online sources is the best practice. However, a careful, manual sanitization must be performed because labels may be invalid. To demonstrate this phenomenon, we studied the validity of phish entries listed in PhishTank. It is the most used online source of phishing websites for evaluating phishing detection techniques. To provide some context, we present the validation process of PhishTank for adding an entry to its blacklist. Suspicious URLs are first submitted by any user to PhishTank and added to an "unverified" list. Entries in this list are later verified in a crowdsourced manner and added to the "valid" list if confirmed as phish. A flag indicates if a given website is currently online or offline. Phishes from the *valid/online* list are typically selected for the evaluation of phishing detection techniques.

We automatically collected URLs newly added to the PhishTank *valid/online* list on an hourly basis over several weeks in the course of 2016. We scraped the webpage pointed by each URL in an automated manner, instantly after we collected them, using an instrumented web browser (Selenium in Python). We saved the source code, starting URL, landing URL and a screenshot of the scraped webpages. This collection gathered 23,118 webpages listed as valid and online according to PhishTank. We manually verified these webpages and identified 13,646 as valid phishes. 41% of the blacklisted URLs (9,472) were not malicious. Such a proportion of mislabeled data would compromise the training of a phishing detection technique and heavily bias the evaluation results. This is why we report the procedure we followed to identify this mislabeled data and we advise it for phishing dataset sanatization.

The verification was done by an operator who manually analyzed all the collected webpages according to the following procedure. We advise this procedure for phishing dataset sanatization.

1. **Screenshot analysis**: The operator visualized the screenshot of the webpages. Screenshots depicting pages for parked domain names, content unavailable websites, Error 404, etc. were identified as mislabeled phishes. These websites were not online anymore when listed by PhishTank. The remaining entries were analyzed in the next step.
2. **Domain name / page content match**: The landing domain name (depicted in the address bar of the screenshot) was compared to the content of the webpage for a match, i.e., to identify if the webpage corresponds to its domain name. If a match was found, we identified a legitimate website, i.e., a mislabeled entry in the blacklist. If no match was found, the entry was confirmed as phish. This match was determined in the following manner. For popular websites like Ebay, PayPal, Amazon, etc., the design of these websites as well as their expected domain names were known to the operator. The operator considered there was no match if, e.g., a PayPal website was hosted under an unknown domain name. For websites that were not known to the operator, the landing domain name was manually visited again at the time of sanitization to verify if the current content was the same as the screenshot we captured. If it was not the same (no match), the entry was confirmed as phish. If it was the same (match), we made a web search in Google using a few prominent terms present in the webpage screenshot. We visited the websites representing the top results from the search until finding the website that had the same design as the screenshot we captured. This website was always found. If the domain name for this website was the same as the landing domain name we recorded during scraping (match), the entry was identified as mislabeled phish. If it was different (no match), the entry was confirmed as phish.

4 Evaluation

We present a systematic methodology for evaluating machine learning-based phishing webpage detection techniques. It consists of recommendations for dataset usage, the evaluation metrics to present and the evaluation of temporal resilience. Temporal resilience refers to the phishing detection technique maintaining a constant accuracy overtime.

4.1 Dataset Usage

Once the ground truth is selected it must be used in a proper manner to ensure that the evaluation results are representative. Evaluation of supervised machine learning requires splitting the ground truth into a training and a testing set. The training set is used to train the classification model while the testing set is used to evaluate the accuracy of the classification model (the phishing detection technique).

1. Training and testing sets must be fully disjoint. The testing set must neither be used for training nor for classification parameters tuning or accuracy results optimization. This leads to knowledge transfer from testing set to classification model that overestimate the predictive performance. This is a basic requirement in any classification evaluation.
2. Use the oldest collected data for training, and the newest collected data for predicting. Phishing campaigns follow temporal trends regarding webpages composition. The widespread classification evaluation practice of cross-validation (used in e.g. [5, 14, 20]) that randomly splits a dataset into a training and a testing set is not recommended. Different phishes developed at the same time period are likely to have the same composition pattern and can be assigned to a training and a testing set. This may overestimate the predictive performance of an overtrained classification model. Previous work has already shown the gap between cross-validation and real-world evaluation results [1]. The evaluation setting must reproduce real-world scenarios where models are trained on data seen until the current point in time and used on newly developed webpages.
3. Use a testing set that is larger than the training set. Learning with fewer instances and getting good evaluation results on a larger dataset ensures generalizability of the detection technique. It means that the features and the classification model capture relevant characteristics of legitimate and phishing webpages and that the method will scale to the large number of online webpages.
4. The dataset must present a real-world distribution of phishing to legitimate webpages as observed in the Internet ($\approx 1/100$) [21, 23]. Since there are far more legitimate than phishing webpages, using a different distribution such as a balanced dataset ($1/1$) leads to results that suffer from base rate fallacy [3]. Using real-world distribution ensures that evaluation metrics, presented in the next section, are representative and relevant.

4.2 Accuracy Metrics

In evaluation of machine learning techniques, two classes, *positive* (P) and *negative* (N), are defined to evaluate accuracy. Here we assume positive to be a phish and negative to be a benign webpage. Consequently a *true positive* (TP) is a detected phish, a *false negative* (FN) is a phish missed by the detection system, a *false positive* (FP) is a benign page detected as phish and a *true negative* (TN) is a benign wepbage identified as benign. Many evaluation metrics (*Accuracy*, *F-Measure*, *AUC*, *MCC*, etc.) can be computed to assess the accuracy of a detection technique. On the one hand, presenting too many metrics can lose a reader in this wealth of information, implying a selection process. On the other hand, some paramount metrics can be omitted while being relevant. We identified the following metrics as the most relevant to assess the detection performance and the usability of a phishing detection technique. Their computation is detailed in Equation 1.

- *True Positive Rate* (TPR) or *Recall* denotes the phishing detection capability of a method, i.e., how good is a technique at protecting from phishing. TPR computes the ratio of phishes that will be detect by the method and it must be maximized.
- *False Positive Rate* (FPR) denotes the reliability of the phishing decision, computing the ratio of legitimate webpages that will be wrongly identified as phish. This value must be minimized.
- *Precision* is a key value highlighting the usability of a method and the unnecessary annoyance it generates while deployed. It computes the ratio of detected phishes that are actual phishes with respect to the total count of detected phishes. It is paramount to have a real-world distribution dataset ($\approx$1 phish/100 benign) to avoid base rate fallacy and get a relevant precision value. Precision gives the ratio of phishing warnings depicted for actual phishes, it must be maximized.

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{TN+FP} \qquad Precision = \frac{TP}{TP+FP} \tag{1}$$

Table 1: Design and evaluation setup for some landmark academic phishing detection systems. Column headers provide the design or evaluation setup reported as well as the section where we presented it. *Design* is limited to client-side (client) or centralized (central.). *Dataset split* refers to split between training and testing set, "old/new" means that the oldest data was used for training and the newest for testing."All" for *accuracy metrics* means TPR, FPR and Precision. Different work used different evaluation methodologies and none of them fully follow the recommendations we presented. We refer readers interested in a more extensive design and performance comparison to [15].

		Sect. 2.2	Sect. 3.1	Sect. 4.1			Sect. 4.2	Sect. 4.3	
Technique	**Year**	**Design**	**Dataset language**	**Dataset split**	**Train /test**	**Leg/ phish**	**Accuracy metrics**	**Long. study**	**Adversary resilience**
Cantina [23]	2007	client	English	none	-	110/1	TPR/FPR	no	discussion
Ma *et al.* [14]	2009	central.	English	cross-valid	1/1	3/4	TPR/FPR	no	discussion
Xiang *et al.* [22]	2009	client	English	none	-	2/1	TPR/FPR	no	no
Whittaker *et al.* [21]	2010	central.	several	old/new	6/1	90/1	all	no	discussion
Monarch [20]	2011	central.	several	cross-valid	4/1	1/1	TPR/FPR	no	discussion
Chen *et al.* [5]	2014	central.	English	cross-valid	9/1	1/5	TPR/FPR	yes	no
PhishStorm [17]	2014	client	English	cross-valid	9/1	1/1	all	no	no
DeltaPhish [6]	2017	central.	English	cross-valid	3/2	4/1	TPR/FPR	no	yes
Off-the-Hook [15]	2017	client	6	old/new	1/20	100/1	all	yes	discussion

4.3 Temporal Resilience

The evaluation of efficacy for a phishing detection technique is usually a one time operation, especially for solutions presented in the academic literature. It is performed on a static dataset collected over a short period of time. However, techniques for crafting a phishing webpage evolve overtime. This evolution is driven by three major trends: (1) the targeting of new websites to mimic, (2) new technology features used in designing phishing websites and (3) adaptation of attackers to circumvent new phishing protection mechanisms. The evaluation of the resilience to this evolution is paramount for a phishing webpage detection technique. (1) and (2) are natural trends in phishing techniques evolution. The evaluation of accuracy for phishing detection techniques with regards to this evolution can be assessed using a *longitudinal study*. The resilience to circumvention by adversaries (3) must be assessed using adversarial techniques.

4.3.1 Longitudinal study

A longitudinal study is realized by repeatedly evaluating the accuracy of the detection technique at fixed time interval (e.g., every month) over an extended period of time (e.g., one year) [15]. The datasets used for this evaluation must be composed of new data freshly collected before evaluation. The evaluation must be performed with and without retraining of the detection technique to evaluate the difference in accuracy metrics between these two strategies. This study is used to infer a maximum *retraining period* (time between two retrainings of the detection technique), for which the accuracy of the system does not degrade. The retraining period provides an estimated maintenance cost for a security vendor deploying the phishing detection technique. Retraining requires data labeling, which is time consuming and costly as discussed in Section 3.2. The longitudinal study demonstrates the readiness of a phish detection technique for real-world deployment:

- If accuracy remains consistently high without retraining, the detection technique is suited for real-world deployment and it has low maintenance cost.
- If accuracy degrades slowly without retraining but improves with retraining, the technique is suited for real-world deployment. The maintenance cost is inversely proportional to the retraining period.
- If accuracy degrades despite retraining, the detection technique is not suited for real-world deployment because of design flaws.

Longitudinal studies are rarely performed in an academic context because of a rush to publication mindset. Nevertheless, they are paramount to the industry and for technology transfer consideration.

4.3.2 Resilience to adversaries

Knowing a detection technique, phishers will try to evade it by adapting their phishes. In the case of machine learning-based detection techniques, the resilience to adversaries is evaluated by performing adversarial machine learning attacks [11], such as techniques for crafting adversarial examples [4, 8]. These adversarial methods can algorithmically modify a phish such that it is misidentified as benign by the detection technique. Such attacks must be simulated against phishing detection techniques,

while evaluating their impact on accuracy metrics. As for the longitudinal study, the accuracy must be evaluated with and without retraining. Similar conclusions about the readiness for real-world deployment can be drawn.

An additional criterion to evaluate is the manipulability of features used by the detection technique. Phishing detection systems use features extracted from webpages as input. The crafting of adversarial examples requires being able to manipulate and modify these features. If these features were selected because of being constrained or not under the control of the phisher, it increases the resilience of the detection technique to adversaries [18]. The adversary cannot freely craft adversarial examples as he cannot modify all features at will.

5 Conclusion

Phishing attacks remain a concerning problem despite the several theoretical solutions proposed in the literature for two main reasons: *design limitations* and *biased evaluation*. To cope with these issues, we presented a number of implementation recommendations to drive design choices of phishing detection techniques in order to improve deployability and usability. We introduced a list of guidelines to evaluate proposed solutions following the selection of a representative ground truth, appropriate dataset usage and relevant metrics to present. These recommendations can also enable fair comparison of phishing detection technique accuracy.

As an illustration of the state-of-the-art in phishing detection, Table 1 provides a summary for the design and evaluation setup used for some landmark work in phishing webpage detection. None of the setups used to evaluate this work fully meet the recommendations we presented in this paper. Academic research in phishing detection must better adopt design and evaluation methods that are relevant to real-world deployment. We hope this paper will make research in phishing detection more impactful, leading to more technology transfer.

Acknowledgment: This work was supported in part by the Academy of Finland under the WiFiUS program (grant 309994) and by Intel Collaborative Research Institute ICRI-CARS.

References

[1] ALLIX, K., BISSYANDÉ, T. F., JÉROME, Q., KLEIN, J., STATE, R., AND TRAON, Y. L. Large-scale machine learning-based malware detection: confronting the "10-fold cross validation" scheme with reality. In *ACM CODASPY* (2014), pp. 163–166.

[2] APWG. Phishing Activity Trends Report. Tech. Rep. 4Q2017, 2018.

[3] BORGIDA, E., AND BREKKE, N. The base rate fallacy in attribution and prediction. *New directions in attribution research 3* (1981), 63–95.

[4] CARLINI, N., AND WAGNER, D. Towards evaluating the robustness of neural networks. In *IEEE S&P* (2017), pp. 39–57.

[5] CHEN, T.-C., STEPAN, T., DICK, S., AND MILLER, J. An anti-phishing system employing diffused information. *ACM Trans. Inf. Syst. Sec. 16*, 4 (2014), 16:1–16:31.

[6] CORONA, I., ET AL. Deltaphish: Detecting phishing webpages in compromised websites. In *European Symposium on Research in Computer Security* (2017), Springer, pp. 370–388.

[7] DE HERT, P., AND PAPAKONSTANTINOU, V. The new general data protection regulation: Still a sound system for the protection of individuals? *Computer Law & Security Review 32*, 2 (2016), 179–194.

[8] GOODFELLOW, I. J., SHLENS, J., AND SZEGEDY, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[9] GUPTA, 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 28*, 12 (2017), 3629–3654.

[10] HAN, X., KHEIR, N., AND BALZAROTTI, D. Phisheye: Live monitoring of sandboxed phishing kits. In *ACM CCS* (2016), pp. 1402–1413.

[11] HUANG, L., JOSEPH, A. D., NELSON, B., RUBINSTEIN, B. I., AND TYGAR, J. Adversarial machine learning. In *ACM AISec* (2011), ACM, pp. 43–58.

[12] KHONJI, M., IRAQI, Y., AND JONES, A. Phishing detection: A literature survey. *Commun. Surveys Tuts. 15*, 4 (2013), 2091–2121.

[13] LE, A., MARKOPOULOU, A., AND FALOUTSOS, M. PhishDef: URL names say it all. In *Proceedings of IEEE INFOCOM* (2011), pp. 191–195.

[14] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *ACM SIGKDD* (2009), pp. 1245–1254.

[15] MARCHAL, S., ARMANO, G., GRÖNDAHL, T., SAARI, K., SINGH, N., AND ASOKAN, N. Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Transactions on Computers 66*, 10 (2017), 1717–1733.

[16] MARCHAL, S., FRANÇOIS, J., STATE, R., AND ENGEL, T. Proactive discovery of phishing related domain names. In *Research in Attacks, Intrusions, and Defenses* (2012).

[17] MARCHAL, S., FRANCOIS, J., STATE, R., AND ENGEL, T. PhishStorm: Detecting phishing with streaming analytics. *IEEE Trans. Netw. Serv. Manag. 11*, 4 (2014), 458–471.

[18] MARCHAL, S., SAARI, K., SINGH, N., AND ASOKAN, N. Know your phish: Novel techniques for detecting phishing sites and their targets. In *IEEE ICDCS* (2016).

[19] THOMAS, K., ET AL. Data breaches, phishing, or malware?: Understanding the risks of stolen credentials. In *ACM CCS* (2017), pp. 1421–1434.

[20] THOMAS, K., GRIER, C., MA, J., PAXSON, V., AND SONG, D. Design and evaluation of a real-time url spam filtering service. In *IEEE S&P* (2011), pp. 447–462.

[21] WHITTAKER, C., RYNER, B., AND NAZIF, M. Large-scale automatic classification of phishing pages. In *NDSS* (2010).

[22] XIANG, G., AND HONG, J. I. A hybrid phish detection approach by identity discovery and keywords retrieval. In *WWW* (2009), pp. 571–580.

[23] ZHANG, Y., HONG, J. I., AND CRANOR, L. F. CANTINA: A content-based approach to detecting phishing web sites. In *WWW* (2007), pp. 639–648.