A Decision-Support Tool for Experimentation on Zero-Hour Phishing Detection

Pavlo Burda(✉), Luca Allodi, and Nicola Zannone

Eindhoven University of Technology, Eindhoven, The Netherlands
{p.burda,l.allodi,n.zannone}@tue.nl

Abstract. New, sophisticated phishing campaigns victimize targets in few hours from attack delivery. Some methods, such as visual similarity-based techniques, can spot these zero-hour attacks, at the cost of additional user intervention. However, more research is needed to investigate the trade-off between automatic detection and user intervention. To enable this line of research, we present a phishing detection tool that can be used to instrument scientific research in this direction. The tool can be used for experimentation on assisting user decision-making, evaluating user trust in detection, and keeping track of users' previous "bad" decisions.

1 Introduction

Research and industry have identified an increasing sophistication of phishing attacks in the last years [4,8,13]. The adoption of innovative detection evasion techniques and the velocity at which phishing attacks arrive and change form make it challenging to design early detection systems able to warn users of the suspicious nature of a visited website. New and unknown attack instances take their toll in the first few hours since delivery (hence *zero-hour* phishing), and attempt to bypass detection systems by fingerprinting user agents or concealing features of cloned pages in embedded objects, while preserving the visual similarity needed to persuade the end user they are indeed on the legitimate webpage [7,15]. For example, Fig. 1 shows a phishing website in which no textual reference to the Office 365 brand in the HTML page, or in single image resources, is present. This makes it hard to automatically extract relevant features by just relying on image resources or textual features from the Document Object Model (DOM).

To counteract evasion techniques, automatic detection methods can identify relevant visual features of a suspicious page (e.g., a logo) and find the corresponding legitimate page using search engines, without relying on slow-to-update block-lists [7]. Such systems, however, are less reliable than allow/block-lists (too many false alarms) and, therefore, require human intervention to effectively counter such attacks [5]. Yet users are often not considered in the design of the tools themselves [2,12]. On the other hand, humans may not heed the generated

G.-V. Jourdan et al. (Eds.): FPS 2022, LNCS 13877, pp. 443–452, 2023.
https://doi.org/10.1007/978-3-031-30122-3_27

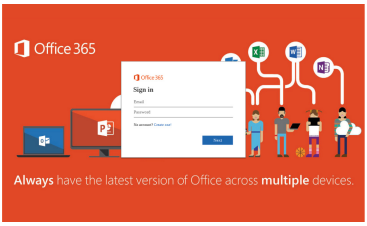

Fig. 1. Phishing website imitating Microsoft Office 365

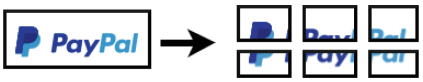

Fig. 2. Example of splitting an image of the PayPal logo to evade detection

warnings for many reasons, such as lack of trust in the decision support system or additional user interface fatigue, with consequent detrimental habitual patterns [14]. Moreover, the amount, type and even content of warnings can depend on the employed detection system and, consequently, affect warning effectiveness.

Assisting users in taking decisions on website legitimacy is in essence the goal of phishing detection and warning effectiveness research. However, gaps in this direction are mainly addressed separately by extant research, for example by improving detection accuracy through the usage of visual features in [7], or by investigating different warning types, as reported in [3]. As a consequence, existing methods and tools are often limited in applicability to experiments that capture the full process where the interaction between the phishing webpage and the user unfolds. For example, even the best detection methods can be ineffective when users do not trust and follow the tool's advice [6]. On the contrary, pitfalls of detection tools, such as false positives or long run-times, can be mitigated with effective risk communication. We argue that these limitations narrow the research possibilities where technology and automation can support individuals in avoiding phishing attacks. To address them, we need an *integrated research approach* that puts both phishing detection and Human Computer Interaction (HCI) ingredients together for an experimental tool to evaluate, characterize, and refine the interaction between zero-hour phishing decision support, and the final user.

In this work, we propose a new experimentation approach to conduct research on zero-hour attack detection and to inform users about related risks. In particular, we present a tool, implemented as a browser extension, to support users in the detection of zero-hour phishing websites, with a particular focus on websites aiming to steal user credentials. The tool relies on a visual similarity-based method for detection and leverages various warning methods for user notification. Thus, our tool enables an integrated research line on zero-hour phishing that allows, for instance to:

- Assess user aids supporting decision-making on website legitimacy.
- Evaluate user trust in a detection system's risk advice.
- Explore new risk communication methods by keeping track of past decisions and associated risks.

2 Need for Zero-Hour Phishing Detection Experimentation

According to industry reports [1,13], a significant fraction of phishing attacks are zero-hour, i.e., when deployed with a variation on previously-observed features (e.g. domains or DOM elements), they cannot be easily linked back to previously-seen attacks (e.g. a similar phishing landing page). Several approaches have been proposed to detect phishing attacks and to communicate risk advice to users. Next, we review existing methods and discuss their drawbacks w.r.t. zero-hour attacks, and argue on the need for more experiments (involving end-users) to investigate their actual effectiveness.

Warnings. Warnings are the primary means to communicate security risks to users [14]. Two main categories are often employed in web browsers: *passive warnings*, which warn users without blocking the content area of a webpage, and *active interstitial warnings*, which block the content area and require an active interaction from the user to be bypassed [10]. Active warnings are more likely to be heeded by users than passive ones and, therefore, considered more effective in averting phishing attacks. Nonetheless, more experiments are needed to understand the effects of these warnings, which still suffer clickthrough rates between 9–18% for the phishing warnings and up to 70% for SSL related warnings [3]. Active warnings carry the risk of disrupting applications' usability too often, to a point where users can develop habitual and detrimental behavior patterns (such as overriding security settings), nullifying warning effectiveness altogether [14]. Moreover, user compliance is very sensitive to the context where warnings are triggered; for example, higher compliance was observed in online banking than in an e-commerce context [6]. Recent work has investigated how to nudge users to pay attention to warnings, for example, with *just-in-time*, *just-in-place* tooltips that elicit a more systematic cognitive response without blocking users completely [17]. This recent line of research integrates multiple disciplines and yields promising results, further signalling the need of new and innovative experimentation in this direction. Overall, research on warnings tends to disregard the internal mechanisms of phishing detection methods. On the other side, users are often not considered in the design of such methods. This has the side effect of limiting methods and tools' applicability to experiments that capture the full process where the interaction between phishing webpages and users unfolds.

Phishing Detection. Automated detection methods of phishing websites can be broadly grouped into three main classes: list-based, heuristic-based, and visual similarity-based [9]. List-based approaches operate by comparing the URL a user visits against a (*block*) list of known phishing websites or an (*allow*) list of legitimate websites. These lists are typically maintained and updated by relying on external crowdsourcing or reputation systems sources, such as PhishTank and Google Safe Browsing. While these solutions have proven to be effective against known threats, they face significant limitations when URLs are yet unknown (new or compromised websites) [7], or due to the time it takes to update block

lists [15]. Especially the slow update of such lists (approximately nine hours [15]) makes these methods ineffective against zero-hour attacks, which trigger victim responses in the first few hours since delivery [5,15]. On the other hand, heuristic-based approaches analyze features extracted from the webpage using predefined rules to determine its legitimacy. These approaches often rely on features such as SSL certificates, anomalies in the DOM or URL, etc. [9] which have however proven to be unreliable as attackers can forge relevant features invisible to heuristic rules [9,10].

These issues are addressed by visual similarity-based approaches, which use content rendered in the browser to determine the (non)legitimacy of a website. These techniques use features such as the logo, the screenshot of the webpage or other features to compare two websites and determine whether one imitates the other [2,11,12]. The advantage of visual similarity-based techniques is that the replacement of text by other objects (such as images and other embedded objects) cannot circumvent the detection technique [7]. However, their ability to detect phishing attacks depends on their ability to find the impersonated legitimate website [10,11]. This important limitation is evident in the state-of-the-art [2,11,12] where it is often addressed by narrowing the scope to a predetermined target list of sites or brands that covers specific classes of phishing attacks.[1]

Zero-Hour Detection. Whether previous work use corpora of predetermined URLs, webpages, screenshots or combinations thereof, these approaches are fundamentally limited in detecting zero-hour attacks not present in the given corpus [10]. Therefore, the identification of a page resembling the page under analysis has to be performed using external sources. Visual similarity-based approaches often apply keyword extraction methods to extract relevant terms from the webpage metadata (e.g., title-tag) and image resources (e.g., logos) in the DOM, which are then fed to a search engine [7]. The underlying assumption is that search engines place benign websites on top [16]. However, the brand name cannot be detected when it occurs only in embedded objects, as it is the case for the webpage in Fig. 1. Similarly, adversaries can compose images from several sub-images as in Fig. 2 or generate them with CSS. Therefore, more robust ways of extracting search terms, which goes beyond applying text mining or extracting images from the DOM for a reverse image search, are needed to effectively enable zero-hour detection capability.

Need for Experimentation. Overall, most of the research in phishing detection deals with improving accuracy and devising new methods for attack detection. However, this often happens without integrating the constraints of the human in the loop. As a consequence, the proposed methods and tools are often limited in applicability to experiments that capture the full cycle of a phishing attack. For example, when (re)producing experiments with such tools, end-user components

[1] Password managers can act as detection methods by flagging mismatched locations of used credentials; however, they are not the de-facto authentication method and still act similarly to an allow-list of previously saved websites.

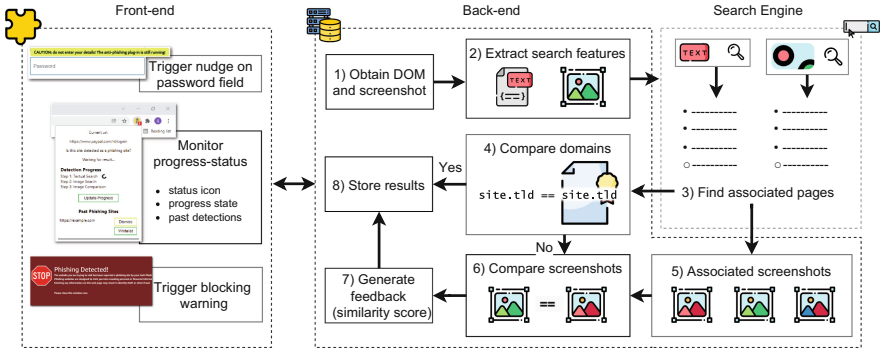

Fig. 3. Overview of the phishing website detection tool

that should connect to the phishing detection method, e.g. browsers or other UI components displaying warnings, are often not provided. This gap affects the possibility to measure or manipulate human-computer interaction (HCI) factors, also in relation to tool capabilities and warning features. When experimenting with phishing detection, we need an integrated research approach that puts all ingredients together in an experimental tool that allows to investigate the complex interaction between users and (risk-based) decision support tools for zero-hour phishing detection.

3 A Tool for the Early Detection of Phishing Websites

To enable experimentation in the context of early phishing detection, we designed and developed a tool that employs a visual similarity-based phishing website detection method as the backend and leverages a variety of warning mechanisms to inform the user about the identified risks posed by the webpage they are visiting. An overview of the overall tool architecture is presented in Fig. 3. The tool is available at: https://github.com/paolokoelio/zerohour-decisionsupport-phishing.

3.1 Backend

The backend of our tool implements the machinery for early detection of phishing websites, which operates on a remote server exposing a REST-like API. In particular, we extended and enhanced a visual similarity-based detection approach that employs *both* textual and visual features of an arbitrary webpage as search terms for the identification of the original webpage from [7]. The overall idea is to combine textual and visual features extracted from a screenshot of the rendered webpages and to evaluate an unknown webpage against the results of the search engines. The output of the evaluation is then used to generate feedback to the user (see Sect. 3.2). Figure 3 (Backend) illustrates how the approach operates. By

relying on search engines and the visual features of a *rendered* webpage (rather than only on features of the DOM), the tool allows a zero-hour protection by avoiding the maintenance of benign allow-lists, and is robust against resource evasion techniques, such as image splitting (Fig. 2), image replacement by pure CSS and image distortions.

As shown in Fig. 3, our approach takes as input a website and obtains the DOM and a screenshot of the rendered webpage (1). Textual features are extracted from the DOM (e.g., title-tag) in a similar fashion to other techniques (cf. Sect. 2). On top of it, visual regions potentially containing identifiable information are extracted from the screenshot (2). These features include, but are not limited to, logos, slogans, parts of header images, and other visual information that is likely to be found in the corresponding legitimate website. Such visual regions are extracted by means of serial image processing steps that rely on region characteristics, such as saliency and high contrast with other elements in a webpage. Region extraction can return several regions, where some of them might not contain information useful to identify the mimicked website. To retain only regions with relevant information, we employ a random forest classifier to filter regions based on regions' features, such as dimensions, coordinates on the page, color properties, and energy-entropy characteristics. The rationale for using certain properties stems from their ability to store "constant" brand/logo-like characteristics [18]. We also rely on the Clearbit public API as an additional region candidate. This API allows to retrieve the logo of a company by sending a request with a URL.

Together with the title-tag, the extracted regions are used as (reverse) search terms to find websites similar to the current webpage through a search engine (*associated pages* in step 3). Based on the intuition that search engines most likely place benign results at the top [16], the top results of both searches are marked as candidate associate pages and used to determine whether the current page is legitimate or not. To determine the legitimacy of the current website, its domain name is checked against the domain names listed in the "Subject Alt Names"-field of the associated pages' SSL certificates (4) (this field contains domain localizations of the current website, e.g., amazon.com, amazon.co.uk, etc.). If the current domain is in this list, the website is marked as "legitimate". Otherwise, a screenshot of the associated pages is obtained (5). Each screenshot is automatically compared with the screenshot of the current webpage using a number of image similarity algorithms (Earth Moving Distance, Discrete Cosine Transformation, etc.) (6) and, depending on their degree of similarity, it is classified as "phishing" or "legitimate". The similarity scores are then used to generate feedback about the legitimacy of the current webpage (7).

3.2 Frontend

To enable experimentation in which the human is in the loop, we also realized a frontend interface as a Chromium browser extension. This allows to include HCI factors in the experimentation ingredients and facilitate experiment deployment. The modules of the plug-in are shown in Fig. 3 (Frontend).

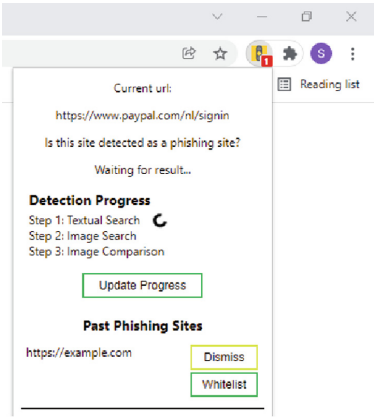

Fig. 4. Extension status pop-up (past phishing sites are displayed)

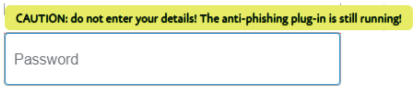

Fig. 5. Passive, just-in-place tooltip when selecting a password field

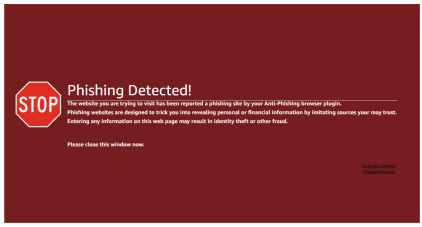

Fig. 6. Active, full-screen blocking warning upon a successful detection

The extension is configured to only scan pages that contain a password field, given the focus on phishing websites aiming to steal user credentials. Upon visiting a page, the detection process starts in the background (cf. Sect. 3.1), and a traffic light icon in the address bar signals the current status, as shown in Fig. 4. The user may click on the icon for more details. On the top, the current URL is displayed together with the outcome, the center contains information on the progress, i.e., the textual/image search and image comparison steps, and the bottom shows the past phishing discoveries.

Whenever users select the password field, the extension triggers the just-in-time just-in-place passive warning in Fig. 5 to remind that the detection in not complete. Researchers can personalize the warning behavior to steer user attention with different designs or impede certain actions by, e.g., temporarily blocking the "Submit" button. When a webpage is detected as a phishing webpage, a full-page blocking warning (Fig. 6) blocks the user if she is still on that page, akin to current browsers' behavior. To ignore this active warning or remember this choice the user must click locate and click on the respective links. Contents and design of the message can be customized to, for example, embed information on the used search features or alter interaction paths to dismiss the message.

To cover cases where the user acts on the webpage before the analysis is complete, past phishing websites are displayed in a retrospective fashion, as shown at the bottom of Fig. 4: even if the user navigates away from the not-yet-detected phishing page, the system will alert the user retrospectively in the status icon and in the pop-up of a detection. Users can dismiss or label as "legitimate" a previously detected phishing URL. The displayed information and interactive elements of the pop-up can be altered to give less or more insights and control to the user, such as near real-time data, history of detection or (de)activation of features.

4 Discussion and Conclusions

As zero-hour phishing detection methods can generate false positives, human intervention is often needed in the decision making process. This, however, places additional burden on the user. To this end, research should assess the best ways to avoid too much strain on the user while keeping them safe. Our work presents a visual similarity-based phishing detection tool that enables this line of research. The tool is packaged into a usable and upgradable browser extension and a web API. This allows an easy deployment of experiments with a scalable number of participants to investigate research gaps in this area. We identify three main research directions that could be supported, experimentally, by the proposed tool:

Assessing User Aids Supporting Decision-Making on Website Legitimacy. Thanks to prior research in usable security, passive indicators have been replaced with blocking warnings. Nonetheless, new experiments can shed light on the gaps not filled by active warnings, such as the circumstances of warning triggering. Our tool can be used to evaluate (types of) warnings in the context of different website categories, such as e-commerce, social media or banking. Similarly, different implementations of nudges, such as dynamic notifications or timed blocking of the "Submit" button, can be tested in various circumstances. For example, experiments can be set up within an organization's embedded phishing training, thus allowing warning efficacy to be tested in an ecologically valid setting.

Evaluating User Trust in a Detection System's Risk Advice. The efficacy of decision-support systems depends on the balance between system's capabilities and users trust [6]. Our tool can help investigating the calibration between the perceived trust and the tool's risk advice by dynamically customizing the warning contents. For example, effects of user calibration on the final decision can be measured by presenting further details on where, how and when a warning has been generated or by displaying the tool's detection statistics. Research on indicator proxies for the inner processes of the tool, such as progress bars or status indicators, has the potential to steer user perceptions and, eventually, improve user choices. Experiments can benefit from the dynamic interaction of the plug-in and the underlying detection logic where, for example, experiment designs may vary the content and placement of status indicators in the browser UI at detection run-time.

Exploring New Risk Communication Methods by Keeping Track of Past Decisions and Associated Risks. Whereas visual similarity-based detection tools are able to detect zero-hour attacks, they have typically long runtimes, which can significantly affect a user's reliance on such tools. Our implementation takes an original approach to this problem: instead of blocking users before the detection is complete (as done by, e.g., Microsoft SafeLink), users are notified retrospectively of the past phishing encounter and, thus, can remediate 'bad' decisions by changing their credentials. While a similar approach has been successfully applied against credential stuffing attacks, it is unclear if this concept is

effective in a near real-time setting. Our tool enables further research in this direction, for example, user studies on the efficacy of retrospective notifications to reduce attack success rates.

Acknowledgment. The authors thank Ardela Isuf and Sam Cantineau for the implementation of the tool. This work is supported by the ITEA3 programme through the DEFRAUDIfy project funded by Rijksdienst voor Ondernemend Nederland (grant no. ITEA191010).

References

1. APWG: Phishing activity trends reports. https://docs.apwg.org/reports/apwg_trends_report_q2_2022.pdf
2. Afroz, S., Greenstadt, R.: PhishZoo: detecting phishing websites by looking at them. In: Int. Conference on Semantic Computing, pp. 368–375. IEEE (2011)
3. Akhawe, D., Felt, A.P.: Alice in warningland: a large-scale field study of browser security warning effectiveness. In: USENIX Security, pp. 257–272 (2013)
4. Allodi, L., Chotza, T., Panina, E., Zannone, N.: The need for new antiphishing measures against spear-phishing attacks. IEEE Secur. Priv. **18**(2), 23–34 (2020)
5. Burda, P., Allodi, L., Zannone, N.: Don't forget the human: a crowdsourced approach to automate response and containment against spear phishing attacks. In: European Symposium on Security and Privacy Workshops, pp. 471–476 (2020)
6. Chen, Y., Zahedi, F.M., Abbasi, A., Dobolyi, D.: Trust calibration of automated security IT artifacts: a multi-domain study of phishing-website detection tools. Inf. Manag. **58**(1), 103394 (2021)
7. van Dooremaal, B., Burda, P., Allodi, L., Zannone, N.: Combining text and visual features to improve the identification of cloned webpages for early phishing detection. In: International Conference on Availability, Reliability and Security, pp. 1–10. ACM (2021)
8. Google: Understanding why phishing attacks are so effective and how to mitigate them (2019). https://security.googleblog.com/2019/08/understanding-why-phishing-attacks-are.html
9. Jain, A., Gupta, B.: Phishing detection: analysis of visual similarity based approaches. Secur. Commun. Netw. **2017**, 1–20 (2017)
10. Khonji, M., Iraqi, Y., Jones, A.: Phishing detection: a literature survey. IEEE Commun. Surv. Tutor. **15**(4), 2091–2121 (2013)
11. Lin, Y., et al.: Phishpedia: a hybrid deep learning based approach to visually identify phishing webpages. In: USENIX Security, pp. 3793–3810 (2021)
12. Liu, R., Lin, Y., Yang, X., Ng, S.H., Divakaran, D.M., Dong, J.S.: Inferring phishing intention via webpage appearance and dynamics: a deep vision based approach. In: USENIX Security, pp. 1633–1650 (2022)
13. Microsoft: Microsoft digital defense report (2021). https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report-2021
14. Modic, D., Anderson, R.: Reading this may harm your computer: the psychology of malware warnings. Comput. Hum. Behav. **41**, 71–79 (2014)
15. Oest, A., et al.: Sunrise to sunset: analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In: USENIX Security, pp. 361–377 (2020)
16. Panum, T.K., Hageman, K., Hansen, R.R., Pedersen, J.M.: Towards adversarial phishing detection. In: Cyber Security Experimentation and Test Workshop, p. 7 (2020)

17. Volkamer, M., Renaud, K., Reinheimer, B.: TORPEDO: TOoltip-poweRed Phishing Email DetectiOn. In: Hoepman, J.-H., Katzenbeisser, S. (eds.) SEC 2016. IAICT, vol. 471, pp. 161–175. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33630-5_12
18. Ye, Q., Jiao, J., Huang, J., Yu, H.: Text detection and restoration in natural scene images. J. Vis. Commun. Image Represent. **18**(6), 504–513 (2007)