Armano, Giovanni; Marchal, Samuel; Asokan, N

Real-Time Client-Side Phishing Prevention Add-On

Published in:
Proceedings - 2016 IEEE 36th International Conference on Distributed Computing Systems, ICDCS 2016

DOI:
10.1109/ICDCS.2016.44

Published: 10/08/2016

Document Version
Early version, also known as pre-print

Please cite the original version:
Armano, G., Marchal, S., & Asokan, N. (2016). Real-Time Client-Side Phishing Prevention Add-On. In *Proceedings - 2016 IEEE 36th International Conference on Distributed Computing Systems, ICDCS 2016* (Vol. 2016-August, pp. 777 - 778). [7536598] IEEE. https://doi.org/10.1109/ICDCS.2016.44

Real-Time Client-Side Phishing Prevention Add-on

Giovanni Armano
Aalto University
Email: giovanni.armano@aalto.fi

Samuel Marchal
Aalto University
Email: samuel.marchal@aalto.fi

N. Asokan
Aalto University and University of Helsinki
Email: asokan@acm.org

***Abstract*—Since existing solutions for steering users away from phishing websites are typically server-based, they have several drawbacks: they compromise user privacy, are not robust against adaptive attackers who serve different content at different times, and do not provide any guidance to users after flagging a website as a phish. To address these limitations, we present a new phishing prevention system implementing a fast and effective phishing detection technique we developed recently [1]. It is implemented as a client-side application and a browser add-on. It uses information extracted from website visited by the user to detect if it is a phish and warn the user. It also determines the target of the phish and offers to redirect the user there.**

I. Introduction

Popular phishing detection systems use the same underlying technique to detect phishs: they rely on URL blacklists built from off-line analysis of suspicious webpages and hosted on a server [2]. This approach has several major drawbacks. It compromises user privacy because the server learns every URL visited by users of the system. It is not robust against adaptive attackers who can rapidly change the content served from the website depending on time or the identity or location of the browser. Moreover, existing systems do not provide any relevant guidance to users in their warning message [3]. They neither give much information about the actual content of the webpage they get prevented connection to.

In this paper, we present a real-time phishing prevention add-on addressing these shortcomings. Its client-side-only implementation (Section II) copes with adaptative phishing attacks serving different content. The warning messages (Section III) are designed to integrate some findings of state-of-the-art usability studies on phishing prevention system [3]–[5] in order to maximize their efficacy. It has the following properties that differs from currently available solutions:

- It provides real-time protection and is privacy friendly, since implemented client-side only.
- Its user warnings partially display the phishing webpage content.
- It identifies the target of phishing webpages and propose redirection to the original website.

II. Implementation

The add-on implements a phishing detection system previously developed [1]. This system leverages machine learning techniques and computes features from data sources retrieved only by the web browser (starting URL, landing URL, redirection chain, logged links and HTML source code). The feature set models phisher limitations and measures the consistency in term usage in the different data sources. The system has two components: a phishing webpage detection system, classifying webpages as phish or legitimate, and a target identification system, inferring the potential targets of a phishing page. To reduce the workload, a whitelist of hashes built upon starting URL, landing URL and HTML source identifies trusted webpages that bypass analysis.

The application is composed of five components: *Background script, Foreground script, Dispatcher, Phishing detector and Target identifier*. The two former run in the browser (Javascript) while the three latter run in the operating system (Python). The architecture of the application and its interactions with the browser are depicted in Fig. 1.

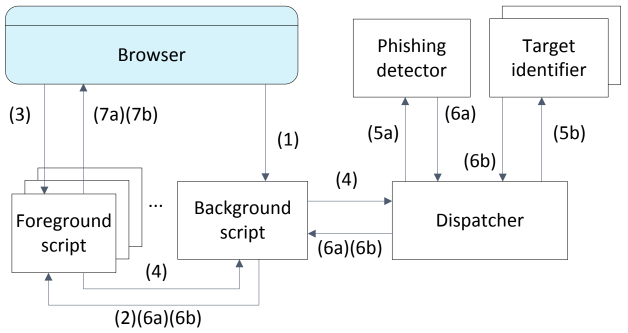


Fig. 1: Application architecture

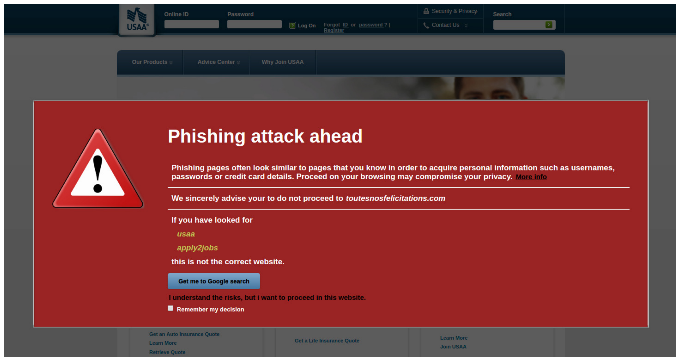
Background script runs concurrently with the browser application. It collects the logged links and redirection chain when a new webpage is loaded in the browser (1). It sends this information to the Foreground script (2) that will further send back the complete list of processed data sources for forwarding to the Dispatcher (4). It later forwards results from the Dispatcher to the Foreground script (6a)(6b).

Foreground script has one instance injected in each open tab of the browser and runs for every loaded webpage. It combines the data sources collected by the Background script (2) and the HTML source code of the current webpage (3). This processed information is sent back for analysis (4). When it receives the Phishing detector response (6a), in case of phishing attack detected, it displays a warning message in its associated browser tab (7a), this warning is further updated with the result of the Target identifier (7b). This two-steps warning is due to computation of the Phishing detector being faster than the Target identifier (cf. Section IV). This implementation choice allows to faster provide a phishing warning message.

Dispatcher receives the data sources collected by the Foreground Script (4) and computes the hash of the webpage for checking against the whitelist. If no match is found, it forwards the data to the Phishing detector (5a) and to one instance of the Target identifier (5b) for processing. It later addresses the results of each operation to the Foreground script (6a)(6b).
Phishing detector performs the phishing detection based on the data sources (5a). The decision is sent to the corresponding instance of the Foreground script (6a).
Target identifier infers the potential targets of the phishing webpage from the received data sources (5b). The targets are sent back to the Foreground Script (6b). If one identified target matches the main level domain of the current website, it is considered safe regardless the decision of the Phishing detector and the warning message will therefore disappear. Otherwise, the phishing message is updated with the potential targets. Due to the time consumed for this task, two Target identifier processes are involved to share the workload.

III. User Warning

We designed active warnings that pop up, filling the whole browser window, in case of detected phishing webpage. Active warnings interrupting user's task catch better their attention than passive warnings [3], [4]. A warning message is displayed in the foreground while the background displays the phishing webpage darkened by a black semi-transparent layer preventing interactions with the website, as depicted in Fig. 2. Displaying the phishing site is a design choice to 1) raise awareness about phishing websites look and feel, and 2) better inform users for taking their decision to proceed or not. User training is an effective protection against phishing attacks [5].



Phishing attack ahead

Phishing pages often look similar to pages that you know in order to acquire personal information such as usernames, passwords or credit card details. Proceed on your browsing may compromise your privacy. More info

We sincerely advise your to do not proceed to *toutesnosfelicitations.com*

If you have looked for
usaa
apply2jobs
this is not the correct website.

I understand the risks, but i want to proceed in this website.

Remember my decision

Fig. 2: Warning message

Users read warning messages but experience difficulties to understand them [3]. Hence, our message contains intelligible information about phishing attacks, and offers three alternative actions. The user can "understand the risks, [...]" and proceed to the website by clicking on the black link, and additionally include this webpage in the whitelist by ticking the checkbox below. She can be directed to Google search engine with a query including the identified potential targets by clicking the blue button. If no target is identified it simply directs to *google.com*. Alternatively, she can choose to visit the official website of any identified target (up to 3) by clicking on one of them (in yellow).

IV. Performance Evaluation

To assess the usability of the system we evaluate its speed and memory usage. Its accuracy (99.9%) has already been assessed [1]. Table I presents the median processing time for the full process of phishing detection (1-7a) and target identification (1-7b). Detailed results excluding data collection (1-4) are presented as well: (5-7a) and (5-7b). The processing time was evaluated on 100 phishing websites (Phish) randomly picked from PhishTank and the top 100 Alexa websites (Leg). The required time for processing phishing websites is lower than legitimate websites and phishing warnings are displayed in less than half a second (473 ms). The target of phish is globally identified in less than 2 seconds after a page is loaded (1915 ms). In case of contradictory decision of the Target identifier regarding false positive of the Phishing detector, the warning message is removed in less than 2 seconds ($2469 - 644 = 1825$ ms).

TABLE I: Processing time (milliseconds)

		(1-4)	(5-7a)	(5-7b)	(1-7a)	(1-7b)
Leg	**Median**	222	405	1994	**644**	**2469**
	Stdev	466	570	1435	822	1551
Phish	**Median**	221	153	1620	**473**	**1915**
	Stdev	600	121	1112	626	1330

TABLE II: Memory usage (Mb)

	Dispatcher	Phish detector	Target identifier	Overall
Average	22.27	298.33	176.10	496.72
Stdev	1.29	3.14	4.43	8.41

Table II presents the average memory usage of the Dispatcher, Phishing detector and Target identifier individually and combined. The Phish detector (298 Mb) and the two Target identifier processes (176 Mb) have the highest memory usage and the complete system requires around 496 Mb of memory.
Acknowledgments: This work was supported in part by the Academy of Finland (grant 274951) and Intel Collaborative Research Center for Secure Computing (ICRI-SC).

References

[1] S. Marchal, K. Saari, N. Singh, and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in *Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016.
[2] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proceedings of the 2010 Network and Distributed System Security (NDSS) Symposium*, 2010.
[3] D. Akhawe and A. P. Felt, "Alice in warningland: A large-scale field study of browser security warning effectiveness," in *Proceedings of the 22nd USENIX Conference on Security*, 2013, pp. 257–272.
[4] S. Egelman, L. F. Cranor, and J. Hong, "You've been warned: An empirical study of the effectiveness of web browser phishing warnings," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1065–1074.
[5] P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A. Blair, and T. Pham, "School of phish: A real-world evaluation of anti-phishing training," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, 2009, pp. 3:1–3:12.