Predictable Testing of Phishing Websites Based on Deportment Model Approach

Sandhya R & Emilin Shyni C

Department of Computer Science, K.C.G College of Technology, Karapakkam, Chennai.
E-mail : sandhya_ravi@outlook.com, emilin@kcgcollege.com

***Abstract* – A phishing attack is a criminal activity which mimics a certain legitimate webpage using a fake webpage with an intention of luring end-users to visit the fake website thereby stealing their personal information such as usernames, passwords and other personal details such as credit card information.An attacker might generate input forms by injecting script code and steal credentials from the legitimate websites.A new approach called trustworthiness testing is used for testing suspected phishing websites.It checks whether the behavior (response) of websites matches with our knowledge of phishing or legitimate website behaviors to decide whether a website is phishing or legitimate. The model is described by capturing the form submission with random number of inputs and corresponding responses.A set of heuristic combination is maintained to test and decide whether websites are phishing or legitimate based on their behaviors.**

Keywords* – *Phishing, Heuristics, Legitimate website, Trustworthiness, Finite State Machine.

I. INTRODUCTION

Phishing is a web-based attack that allures end users to visit fraudulent websites and give away personal information (e.g., user id, password). The stolen information is the beginning point of many illegitimate activities such as online money laundering Phishing attacks cost billions of dollars in losses to business organizations and end users. There are two main activities performed by phishers to make an attack successful. They are (i) developing fraudulent websites, and (ii) motivating (or urging) users to visit those sites. The fake websites have similar look and feel of legitimate websites, which are owned by organizations such as banks, credit unions, and governments. Phishers download pages of legitimate websites and modify some parts of these pages.

These include detecting suspicious websites with heuristics, educating and training users, compiling white lists and blacklists, filtering emails, and customizing visual cues to distinguish legitimate websites from fake websites. Most browsers (e.g., Firefox, Internet Explorer) have built-in phishing attack detection abilities based on white and blacklisted websites.

II. BACKGROUND

Phishing is an important security problem. Although phishing is not new and, hence, should be well-known by Internet users, many people are still tricked into providing their confidential information on dubious web pages. To counter the phishing threat, a number of antiphishing solutions have been proposed, both by industry and academia.The key idea is that when a phishing website does not reach its victims, they cannot fall for the scam.Hence, Trustworthiness testing has been pioneered by the work of self-verifying data where explicit properties (e.g., source of input data, content layout, and format) of data structures are checked during testing as opposed to the verification of program outputs. Several works have tested non-testable programs through testing of trustworthiness.

Zhang et al. [1] develop the CANTINA tool which leverages the TF-IDF (term frequency and inverse document frequency) algorithm to identify most weighted texts (or words) and generates lexical signatures from the top five most important words. These signatures are searched through a trusted search engine such as Google. The resultant domain names are compared with the current domain. If there is no match, then the current page is identified as phishing.

Chou et al. [2] develop the SpoofGuard tool that detects phishing web pages based on heuristics and computes

spoof scores based on matched heuristics. The heuristics include the features of stateless evaluation (e.g., a page containing Amazon logo asking user id and password), stateful evaluation (e.g., whether a domain already visited before), and input data (e.g., data has been sent to a site before). If the score exceeds a threshold, a page is suspected to be phishing.

Xiang et al. [3] apply information extraction and retrieval techniques to detect phishing pages. The DOM of a downloaded page is examined to recognize its identity through different attributes (e.g., page title, copyright) and identify the actual domain name based on the identity. Next, they search the current domain of a suspected page and compare the result with theprevious search output. If there is no common result in the two sets, the downloaded page is suspected as phishing.

Hook and Kelly [4] apply a testing technique to verify program trustworthiness when exact outputs are unknown. They apply mutation-based analysis that generates mutants (source code modification based on a set of rules) of a given program followed by a set of test cases (random inputs). While mutants are killed by randomly generated test cases, the set of generated outputs of mutant programs are compared with the set of outputs of the original program. The idea is that if the maximum deviation between the outputs generated by a mutant and an original program is acceptable, then the implemented program is considered as trustworthy. We believe that testing of suspected phishing websites shares some common similarities of their work as we do not know the response pages before testing.

Yue and Wang [5] develop the BogusBiter tool that intercepts credential of users, generates a large number of fake credentials, and places the credential among the fake credentials to nullify the attack. A similar approach has been applied by Joshi et al. [6] (the PhishGuard tool) who intercept user submitted credentials. However, to hide an actual supplied credential, they send another set of fake credentials at the end. Krida and Kruegel [7] save a mapping between supplied credentials and corresponding trusted domains during the learning phase. In a detection phase, a submitted credential is matched with the saved credentials, and the current domain name is compared with the saved domain names. If there is no match, a website is suspected as phishing.

Pan and Ding [8] detect phishing web pages by identifying the anomalies in declared identities (e.g., keyword, copyright related text present in HTML) and observing how anomalies manifest through DOM objects and HTTP transactions (e.g., server form handler). Dong et al. [9] develop user profiles by creating or updating binding relationships that relate user supplied personal information and trusted websites. When a user is about to submit his credentials, a detection engine generates a warning, if there is no match between the current and the previously learned binding relationship.

Leung et al. [10] develop a set of test criteria for trustworthiness testing of embedded programs (mobile handsets). Their developed criteria are based on the intrinsic properties of production environment (e.g., network wave should fluctuate, the signal strength should be within a specified limits), operation environment (e.g., a mobile station sometimes receives signal from a faraway base station as opposed to the nearest station), and benchmark results (e.g., output of two mobile handsets should be close or similar).

In contrast, the proposed approach tests the trustworthiness of suspected websites based on known phishing and legitimate website behaviors. The heuristics are developed based on a FSM model representing behaviors related to form submissions with random inputs.

III. DEPORTMENT MODEL AND TESTING

The model provides us the flexibility to detect phishing websites that might steal information through an uncertain number of pages containing forms and employ various types of form generation techniques (e.g., non XSS-based, XSS-based). Apply offline analysis approach to navigate and download all the accessible pages by submitting random inputs and observe interesting responses.

A. State-based heuristics:

The sate based heuristics consist of three set of heuristics they are no loop, single loop, multiple loop.

1) No loop (H1): If a website does not travel to the same page again it is indicates as no loop.

2) Single loop (H2): If a website travels to the same page again it is indicated as the single loop. Meeting the single loop heuristic might not always indicate that a site is phishing.

3) Multiple loops (H3): If a website travels to the same page more than one time it is indicated as multiple loop. Multiple loop is suspected as phishing site.

B. Response-based heuristics:

It contains three types of heuristics that relate the response of form submissions. These include maximum form submission, maximum error message, and the presence of supplied input.

1) *Maximum form submission (H4):* A phishing website might be designed to accept unlimited number of form submissions. The heuristic checks the maximum number of form submission during testing (H4). If a website exceeds the maximum number of form submissions with random inputs during testing, it is most likely to be a phishing website as opposed to a legitimate site. We consider the maximum number of form submission value as six in this paper.

2) *Maximum error message (H5):* A legitimate website might block a user for providing random inputs. This observation leads to the development of a heuristic based on the maximum number of error message observed during form submission. If providing random inputs result in the maximum number of error messages, it indicates that a website is likely to be a legitimate as opposed to a phishing. We consider the maximum number of error message as three.

3) Presence of supplied input (H6): This heuristic is satisfied, if the response of a form submission contains part of the random inputs that have been provided. A legitimate website often greets a user after a successful login or registration with the supplied name, user id, or email. On the other hand, a phishing website does not generate a response page that contains the supplied inputs.

C. Form-based heuristics:

It consist of two form feature-based (also denoted as form-based) heuristics. These include No form, Common form.

1) *No form (H7):*This heuristic checks whether a form submission results in a page that has no input form (or no hyperlink to proceed further). In other words, there is no way to proceed further from the response page. Note that if a response page is nonexistent, we consider it as an example of no form (i.e., the heuristic is satisfied).

2) Common form (H8): This heuristic is satisfied, if a current form being submitted with random inputs matches with any of the forms that have been submitted before. Two forms are considered as common form if their field name and types are similar.

IV. HEURISTIC COMBINATIONS

The heuristics combinations are maintained form submission response, and form-based heuristics. Using state, response, and form-based heuristics allow us to distinguish all the phishing websites from legitimate websites. For example, a legitimate website can be detected by checking two different combinations of heuristics: (i) H1, H2, H6, and H7, or (ii) H1, H2, H5, and H8. On the other hand, if a website behavior satisfies H1, H2, and H7 then it can be concluded as phishing. The state-based heuristics are generalized by the notation {H1, H2, H3}. This implies that a website might satisfy one or more state-based heuristics. However, the form submission response based heuristics are specific towards phishing (H4) and legitimate websites (H5 and H6).

V. PROCESS FLOW

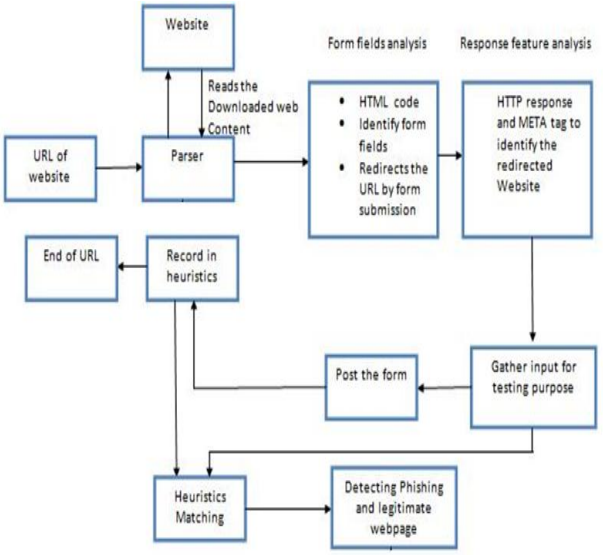

Fig .1 : System Architecture

The Parser downloads the html content from the website when the URL is fed to the browser. The parser initially reads the content and downloads the webpage. The parser analyses the form fields and the Response fields.

All the input is being gathered for the testing purpose. Form post is done after the random form filling. All the form post is recorded in heuristics. The heuristics recording is done till the end of URL. The heuristic matching is done; the legitimate and phishing WebPages can be identified based on the heuristics combinations.

VI. RESULT AND EVALUATION

We perform experiments in this section to identify two issues:

A. *Comparative evaluation*

To validate the testing approach for detecting phishing attacks that might be launched through XSS-based forms in trusted websites. XSS enables attackers to inject client-side scriptinto Web pages viewed by other users. A cross-site scripting vulnerability may be

used by attackers to bypass access controls such as the same origin policy.

The SSL based attacks is man-in-the-middle attacks in which the attacker makes independent connections to the victims and relays messages between them, making them believe that they are talking directly to each other over a private connections. There is some non-SSL page which links/redirects to an SSL page. SSL sees that and changes the link/redirect to a non-secured link/redirect. For example, you might have index.php linking to https://domain.com/login.php. SSL changes it to http://domain.com/login.php.

NetCraft and Spoof Guard will not support the XSS and SSL based attacks. The proposed system support both XSS and SSL based attacks.

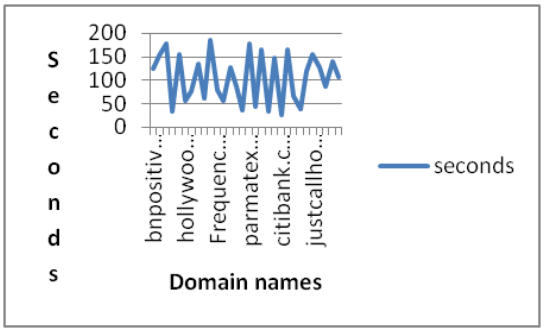

Fig.2 : Time duration for detecting legitimate websites by Spoof Guard

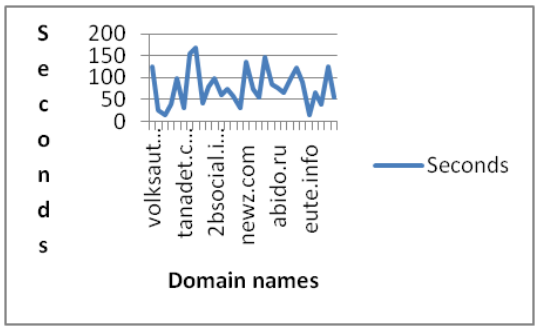

Fig.3 : Time duration for detecting phishing websites by Spoof Guard

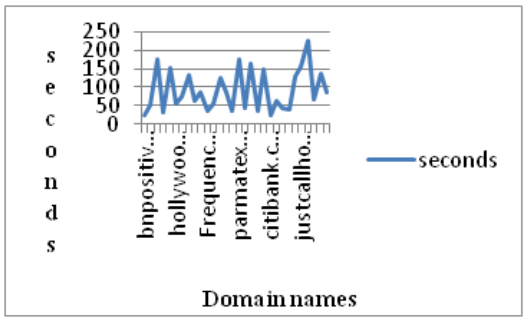

Fig. 4 : Time duration by the Phish detector

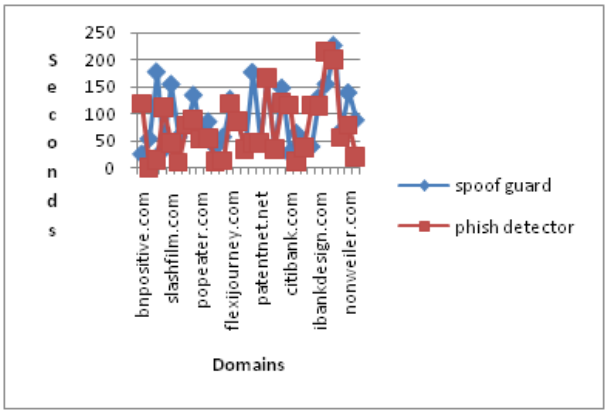

Fig.5 : Comparison between spoof guard and phish detector

VII. CONCLUTION AND FUTURE WORK

Testing of phishing websites is challenging due to uncertainty of website behaviors for random inputs and naturally fits well with the notion of trustworthiness testing. Here, trustworthiness testing is intended to check whether a program is performing functionalities against a set of benchmark or human knowledge of program behaviors. The proposed trustworthiness testing of suspected phishing websites to mitigate these issues. Consider the suspected websites as web-based programs. These programs should demonstrate different behaviors or responses with respect to random input submissions among phishing and legitimate websites. Such behaviors are denoted with the notion of Finite State Machine (FSM). The eight heuristics is developed based on state, submission response, and form-based features. To facilitate the testing, further seven heuristic combinations is developed for testing phishing and legitimate websites.

The future work includes Retrieval-based identity Recognition which can able to detect the phishing websites. The Website brand names usually appear in a certain parts of a webpage such as title, copyright field, etc., which renders the website identity searchable and recognizable.

VIII. REFERENCE

[1] Hossain Shahriar, Mohammad Zulkernine, trustworthiness testing of phishing websites: a behavior model-based approach, future generation computer systems 28 (2012) 1258–1271

[2] Y. Zhang, J. Hong, L. Cranor, CANTINA: a content-based approach detecting phishing websites, in: Proc. of the 16th Intl. Conf. on World Wide Web, Banff, Alberta, May 2007, pp.

639–648.

[3] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, J. Mitchell, Client-side defense against web-based identify theft, in: Proc. of the 11th Annual Network and Distributed System Security Symposium, NDSS'04, San Diego, CA, February 2004.

[4] G. Xiang, J. Hong, A hybrid phish detection approach by identity discovery and keywords retrieval, in: Proceedings of the 18th Madrid, Spain, April 2009, pp. 571–580.

[5] International Conference on World Wide Web, Hook, D. Kelly, Testing for trustworthiness in scientific software, in: Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, Vancouver, Canada, May 2009, pp. 59–64.

[6] C. Yue, H. Wang, Anti-phishing in offense and defense, in: Proc. of the Annual Computer Security Applications Conference, ACSAC, Anaheim, California, December 2008, pp. 345–354.

[7] Y. Joshi, S. Saklikar, D. Das, S. Saha, PhishGuard: a browser plug-in for protection from phishing, in: Proc. of the 2nd International Conference on Internet Multimedia Services Architecture and Applications, Bangalore, India, December 2008, pp. 1–6.

[8] Krida, C. Kruegel, Protecting users against phishing attacks with AntiPhish, in: Proc. of the 29th Annual International Computer Software and Applications Conference, Edinburgh, Scotland, July 2005,

[9] Y. Pan, X. Ding, Anomaly-based web phishing page detection, in: Proc. of the 22nd Annual Computer Security Applications Conference, Miami, Florida, December 2006, pp. 381–392.

[10] X. Dong, J.A Clark, J. Jacob, User behavior-based phishing websites detection, in: Proc. of International Multiconference on Computer Science and Information Technology, Wisla, Poland, October 2008, ,pp. 783–790.

[1] K. Leung, J. Ng, W. Yeung, Embedded program testing in untestable mobile environment: an experience of trustworthiness approach, in: Proceedings of the 11th Asia-Pacific Software Engineering Conference, Busan, Korea, November 2003, pp. 430–437.

◈◈◈