An Efficient and Usable Client-Side Cross Platform Compatible Phishing Prevention Application

FINAL YEAR PROJECT REPORT

Submitted by

N. Dhanush(2016103021)

G. Santhosh(2016103057)

S. Ben Stewart(2016103513)

College of Engineering, Guindy

ANNA UNIVERSITY: CHENNAI 600 025

SEPTEMBER 2019

1 PROBLEM STATEMENT

Phishing is a crime where the victim is contacted by an attacker posing as a trustworthy source and lure them into providing sensitive information like credit card details and personal identification numbers. These attacks are currently being blocked by web browsers that have a list of such phishing links. It takes several days and intense computing resources to prepare the list. Having a time lag in this process means that many victims are vulnerable at that point in time to such an attack. Inorder to make the process more efficient, the functionality required will be ported to the client side of the web browser. This makes sure that the time delay is averted and the phishing attack can be thwarted with fewer computational resources.

To solve the above mentioned problem, we will be implementing a web browser add-on that works as a background script on the client side. All the background scripts required will be made cross platform compatible to make the development easier and more efficient.

2 RELATED WORK

Phishing is the attempt to get the confidential information from vulnerable victims by having them to provide those information on websites that impersonate the real trustworthy websites. These fraudulent websites are passes through the electronic mail or other such communication methods, which the victims access using their web browser. Such links are blocked by the browser using a list that is created offline and that requires a lot of computing power but does not protect the victim against a dynamic phishing attack. To solve this problem, i.e. to reduce the computational resources required on the server side and to prevent dynamic phishing attacks, the previous work by Samuel Marchal, Giovanni Armano, Tommi Grondahl, Kalle Saari, Nidhi Singh, and N. Asokan implemented a client-side phishing prevention application named Off-the-hook.

We found a few issues with Off-the-hook, with the most important and glaring issues being that it had to be implemented individually for each platform. Also the extraction technique they used was rising a few issues because of the stop word removal they used and the fact that once the attacker gets to know the implementation schema he can circumvent it by using a free subdomain name that can be used to trick Off-the-hook into marking the site as legitimate.

Thus in this project we attempt to fix the problems that Off-the-hook has mainly Temporal resilience in which the accuracy of the application must not degrade overtime and aso to make the application cross platform.

3 SYSTEM ARCHITECTURE

The below figure gives an overall idea of how the components work together with the pipeline complete.

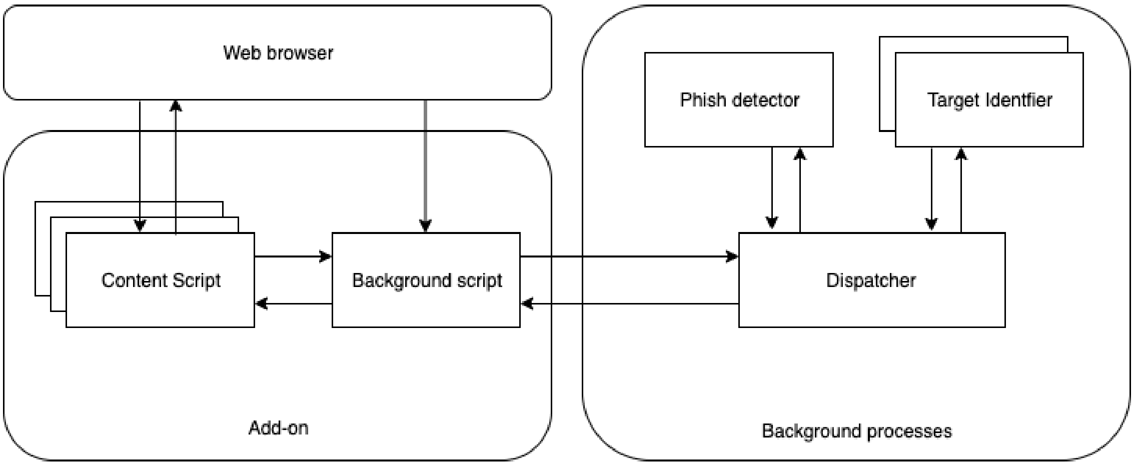

Figure 3.1 Detailed System Architecture

4 FINAL DEMONSTRATION

At the end of this project a working implementation of the above cross platform compatible add-on for phishing prevention will be submitted. It will be made to work with different OS platforms and several web browsers. The add-on will have a phishing site to be accessed using a web browser and the success condition would be that the add-on successfully identifies it to be a phishing link and notify the user.

REFERENCES

[1] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, N. Asokan, "Off-the-hook: An efficient and usable client-side phishing prevention application", *IEEE Trans. Comput.*, vol. 66, no. 10, pp. 1717-1733, Oct. 2017.

[2] A. K. Jain, B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list", *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, Dec. 2016.

[3] G. Xiang, J. Hong, C. P. Rosé, L. Cranor, "CANTINA: A feature-rich machine learning framework for detecting phishing Web sites", *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, 2011.

[4] *Implementation for the Usage of Google Safe Browsing APIs (v4)*, 2019, [online] Available: https://github.com/google/safebrowsing.

[5] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic clas- sification of phishing pages,” in Proc. Netw. Distrib. Syst. Security Symp., 2010, pp. 1–14.