PHISHTOR

PHISHING WEBSITE DETECTION - CHROME PLUGIN

by

ASWIN SUNDAR - 2015103006

RAMKUMAR B - 2015103042

A project report submitted to

FACULTY OF DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

for

CREATIVE AND INNOVATIVE PROJECT

in the academic year of 2018-2019

ABSTRACT

This document proposes a phishing[1] detection plugin for chrome browser that can detect and warn the user about phishing web sites in real-time using random forest classifier. Based on the IEEE paper, Intelligent phishing website detection using random forest classifier[2], the random forest classifier seems to outperform other techniques in detecting phishing websites.

One common approach is to make the classification in a server and then let the plugin to request the server for result. Unlike the old approach, this project aims to run the classification in the browser itself. The advantage of classifying in the client side browser has advantages like, better privacy (the user's browsing data need not leave his machine), detection is independent of network latency.

This project is mainly of implementing the above mentioned paper in Javascript for it to run as a browser plugin. Since javascript doesn't have much ML libraries support and considering the processing power of the client machines, the approach needs to be made lightweight. The random forest classifier needs to be trained on the phishing websites dataset[3] using python scikit-learn and then the learned model parameters need to be exported into a portable format for using in javascript.

---

[1] Phishing is mimicking a creditable company's website to take private information of a user.

[2] https://ieeexplore.ieee.org/abstract/document/8252051/

[3] https://archive.ics.uci.edu/ml/datasets/phishing+websites

TABLE OF CONTENTS

ABSTRACT **2**

TABLE OF CONTENTS **3**

CHAPTER 1 **4**

INTRODUCTION **4**

1.1 PROBLEM DOMAIN 4

1.2 PROBLEM DESCRIPTION 5

1.3 SCOPE 6

1.4 CONTRIBUTION 6

1.5 SWOT ANALYSIS 7

1.6 PESTLE ANALYSIS 7

1.7 ORGANISATION OF THESIS 9

CHAPTER 2 **10**

RELATED WORKS **10**

2.1 DIRECTORY BASED APPROACHES 11

2.2 RULE BASED APPROACHES 11

2.3 ML BASED APPROACHES 12

2.4 DRAWBACKS 13

CHAPTER 3 **14**

REQUIREMENTS ANALYSIS **14**

3.1 FUNCTIONAL REQUIREMENTS 14

3.2 NON FUNCTIONAL REQUIREMENTS 14

3.3 CONSTRAINTS AND ASSUMPTIONS 15

3.4 SYSTEM MODELS 16

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DOMAIN

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords, and credit card details (and money), often for malicious reason. It is typically carried out by email spoofing or instant messaging, and it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site, the only difference being the URL of the website in concern. Communications purporting to be from social web sites, auction sites, banks, online payment processors are often used to lure victims. Phishing emails may contain links to websites that distribute malware.

Detecting phishing websites often include lookup in a directory of malicious sites. Since most of the phishing websites are short lived, the directory cannot always keep track of all, including new phishing websites. So the problem of detecting phishing websites can be solved in a better way by machine

learning techniques. Based on a comparison of different ML techniques, the random forest classifier seems to perform better.

Only way for an end user to benefit from this is to implement detection in a browser plugin. So that the user can be warned in real time as he browses a phishing site. However, browser extensions have restrictions such as they can be written only in javascript and they have limited access to page URLs and resources.

Existing plugins send the URL to a server, so that the classification can be done in the server and the result is returned to the plugin. With this approach, user privacy is questioned and also the detection may be delayed due to network latency and the plugin may fail to warn the user in right time. As it is an important security problem and also considering the privacy aspects, we decided to implement this on a chrome browser plugin which can do the classification without an external server.

1.2 PROBLEM DESCRIPTION

To develop a browser plugin which once installed, should warn the user on the event of he/she visiting a phishing website. The plugin should not contact any external web service for this which may leak the user's browsing data. The detection should be instant so that the user will be warned before entering any sensitive information on the phishing website.

1.3 SCOPE

According to wikipedia, In 2017, 76% of organisations experienced phishing attacks. Nearly half of information security professionals surveyed said that the rate of attacks increased from 2016. In the first half of 2017 businesses and residents of Qatar were hit with more than 93,570 phishing events in a three-month span. With increasing number of internet users, there is a prominent need for security solutions again attacks such as phishing. Hence this plugin would be a good contribution for the chrome users.

1.4 CONTRIBUTION

This is the first implementation of phishing website detection in browser plugin without use of an external web service. This makes use of existing works done on phishing detection and implements them in a manner that it will benefit end users. This involves porting the existing python classifier (random forest) to javascript. The plugin with an one time download of the learned model, will be able to classify websites in real time. This involves developing such a model (random forest) in javascript, as browser plugin supports only javascript. Thus this project contributes to better privacy and rapid detection of phishing.

1.5 SWOT ANALYSIS

STRENGTHS	WEAKNESSES
• Enables user privacy. • Rapid detection of phishing. • Can detect new phishing sites too. • Can interrupt the user incase of phishing.	• Javascript limits functionality. • Cannot use features that needs a external service such as SSL, DNS, page ranks. • No library support.
OPPORTUNITIES	**THREATS**
• Everyone conscious of privacy and security can use this plugin. • Non technical people who do business. transactions are vulnerable to phishing and they are potential end users for this.	• Server side classification plugins may perform better than this and users without privacy concerns may opt of those. • Chrome Plugin API will be continuously changed and effort needs to taken in updating the plugin regularly to avoid deprecated APIs.

1.6 PESTLE ANALYSIS

1.6.1 Political

This project is rarely controlled by political factors. One such scenario is that the government may take any measure to

prevent phishing and in such cases the plugin may lose its potential.

1.6.2 Economical

The plugin is completely based on public dataset and open chrome plugin API. Thus it has no Economical factors controlling it.

1.6.3 Social

The social factor that will have effect on this plugin is awareness of the user. Phishing detection systems aims to aid an user in finding a phishing sites. At least the users need to be aware about phishing to install this plugin.

1.6.4 Technological

Technological advancements or improved techniques for phishing detection can make this plugin become outdated and are a major threat to this plugin.

1.6.5 Legal

Legal policies those in concern of user privacy such as GDPR will enhance the potential of this plugin as user will be moving to more privacy based products.

1.6.6 Environmental

This plugin has no environmental factors affecting it.

1.7 ORGANISATION OF THESIS

Chapter 2 discusses the existing approaches to phishing detection in greater detail. Chapter 3 gives the requirements analysis of the system. It explains the functional and non-functional requirements, constraints and assumptions made in the implementation of the system and the various UML diagrams. Chapter 4 explains the overall system architecture and the design of various modules along with their complexity. Chapter 5 gives the implementation details of each module. Chapter 6 elaborates on the results of the implementation. Chapter 7 concludes the thesis and gives an overview of its criticisms.

CHAPTER 2

RELATED WORKS

This chapter gives a survey of the possible approaches to phishing website detection. This survey helps to identify various existing approaches and to find the drawbacks in them. The difficulty in most of the approaches is that they are not implemented in real time so that an end user will benefit from it.

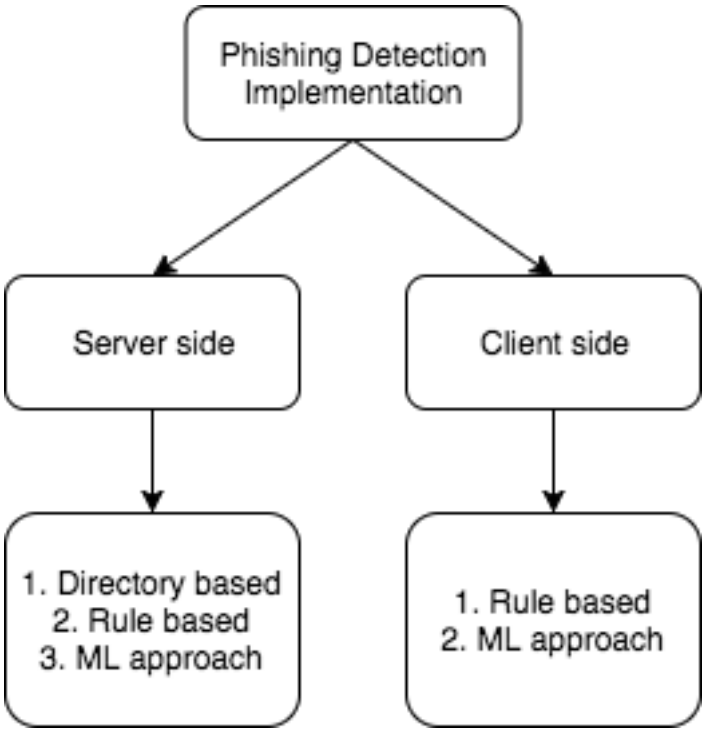

Figure 2.1 Approaches to phishing detection

2.1 DIRECTORY BASED APPROACHES

Most popular one of this kind is PhishTank. According to PhishTank[4], it is a collaborative clearing house for data and information about phishing on the Internet. Also, PhishTank provides an open API for developers and researchers to integrate anti-phishing data into their applications at no charge. Thus PhishTank is a directory of all phishing websites that are found and reported by people across the web so that developers can use their API for detecting phishing websites.

Google has a API called Google Safe Browsing API which also follows directory based approach and also provides open API similar to PhishTank.

This kind of approach clearly can't be effective as new phishing web sites are continuously developed and the directory can't be kept up to date always. This also leaks users browsing behaviour as the URLs are sent to the PhishTank API.

2.2 RULE BASED APPROACHES

An existing chrome plugin named PhishDetector[5] uses a rule based approach so that it can detect phishing without external web service. Although rule based approaches support easier implementation on client side, they can't be accurate compared to Machine Learning based approaches. Similar work by Shreeram.V

---

[4] http://phishtank.com/

[5] https://chrome.google.com/webstore/detail/phishdetector-true-phishi/kgecldbalfgmgelepbblodfoogmjdgmj

on detection of phishing attacks using genetic algorithm[6] uses a rule that is generated by a genetic algorithm for detection.

PhishNet is one such Predictive blacklisting approach. It used rules that can match with TLD, directory structure, IP address, HTTP header response and some other.

SpoofGuard[7] by Stanford is a chrome plugin which used similar rule based approach by considering DNS, URL, images and links.

2.3 ML BASED APPROACHES

Intelligent phishing website detection using random forest classifier (IEEE-2017) by Abdulhamit Subasi, Esraa Molah, Fatin Almkallawi and Touseef J. Chaudhery discusses the use the random forest classifier for phishing detection. Random Forest has performed the best among the classification methods by achieving the highest accuracy 97.36%.

PhishBox: An Approach for Phishing Validation and Detection (IEEE-2017) by Jhen-Hao Li, and Sheng-De Wang discusses ensemble models for phishing detection. As a result, The false-positive rate of phishing detection is dropped by 43.7% in average.

Real time detection of phishing websites (IEEE-2016) by Abdulghani Ali Ahmed, and Nurul Amirah Abdullah discusses an approach based on features from only the URL of the website.

---

[6] https://ieeexplore.ieee.org/document/5670593/

[7] https://crypto.stanford.edu/SpoofGuard/

They were able to come up with a detection mechanism that is capable of detecting various types of phishing attacks maintaining a low rate of false alarms.

Netcraft[8] is one popular phishing detection plugin for chrome that uses server side prediction.

2.4 DRAWBACKS

Based on the above mentioned related works, It can be seen that the plugins either use rule based approach or server side ML based approach. Rule based approach doesn't seem to perform well compared to ML based approaches and on the other side ML based approaches need libraries support and so they are not implemented in client side plugin. All the existing plugins send the target URL to an external web server for classification. This project aims to implement the same in browser plugin removing the need of external web service and improving user privacy.

[8] https://toolbar.netcraft.com

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The plugin warns the user when he/she visits a phishing website. The plugin should adhere to the following requirements:

- The plugin should be fast enough to prevent the user from submitting any sensitive information to the phishing website.
- The plugin should not use any external web service or API which can leak user's browsing pattern.
- The plugin should be able to detect newly created phishing websites.
- The plugin should have a mechanism of updating itself to emerging phishing techniques.

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 User Interface

There must be a simple and easy to use user interface where the user should be able to quickly identify the phishing website. The input should be automatically taken from the

webpage in the current tab and the output should be clearly identifiable. Further the user should be interrupted on the event of phishing.

3.2.2 Hardware

No special hardware interface is required for the successful implementation of the system.

3.2.3 Software

- Python for training the model
- Chrome browser

3.2.4 Performance

The plugin should be always available and should make fast detection with low false negatives.

3.3 CONSTRAINTS AND ASSUMPTIONS

3.3.1 Constraints

- Certain techniques use features such as SSL, page rank etc. Such information cannot be obtained from client side plugin without external API. Thus those features can’t be used for prediction.
- Heavy techniques can’t used considering the processing power of client machines and the page load time of the website.

- Only Javascript can be used to develop chrome plugins. Machine learning libraries support for javascript is far less compared to python and R.

3.3.2 Assumptions

- The plugin is provided with the needed permissions in the chrome environment.
- The user has a basic knowledge about phishing and extensions.

3.4 SYSTEM MODELS

3.4.1 Use Case Diagram

The overall use case diagram of the entire system is shown in figure 3.1. The user can install the plugin and then can continue his normal browsing behaviour. This plugin will automatically check the browsing pages for phishing and warns the user of the same.

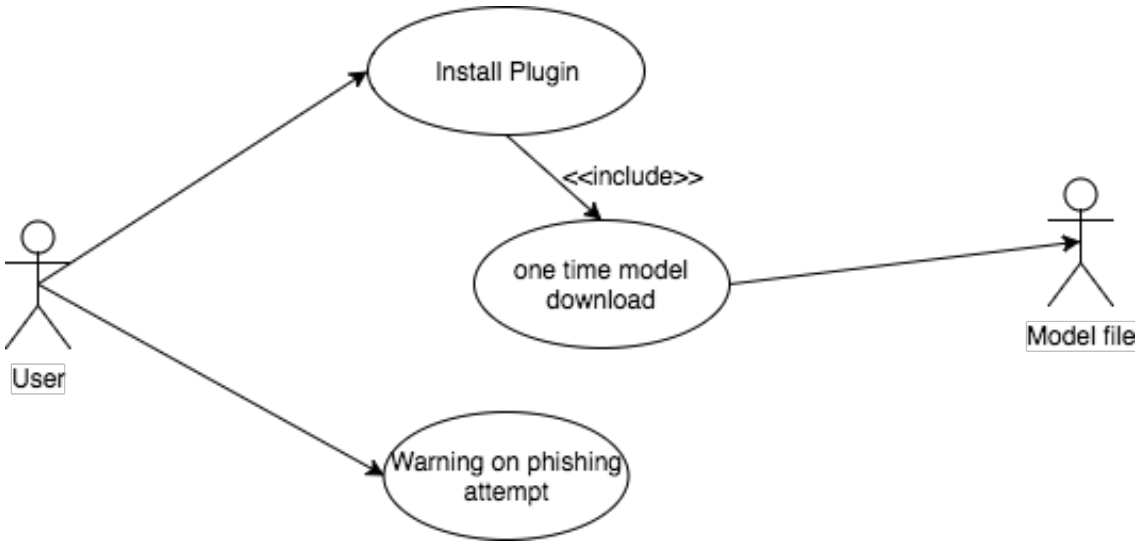

Figure 3.1 Use case diagram of the system

Pre condition: The user visits a website and have plugin installed.

Post condition: The user is warned incase it's a phishing website.

3.4.2 Sequence diagram

The sequence of interactions between the user and the plugin are shown in the figure 3.2

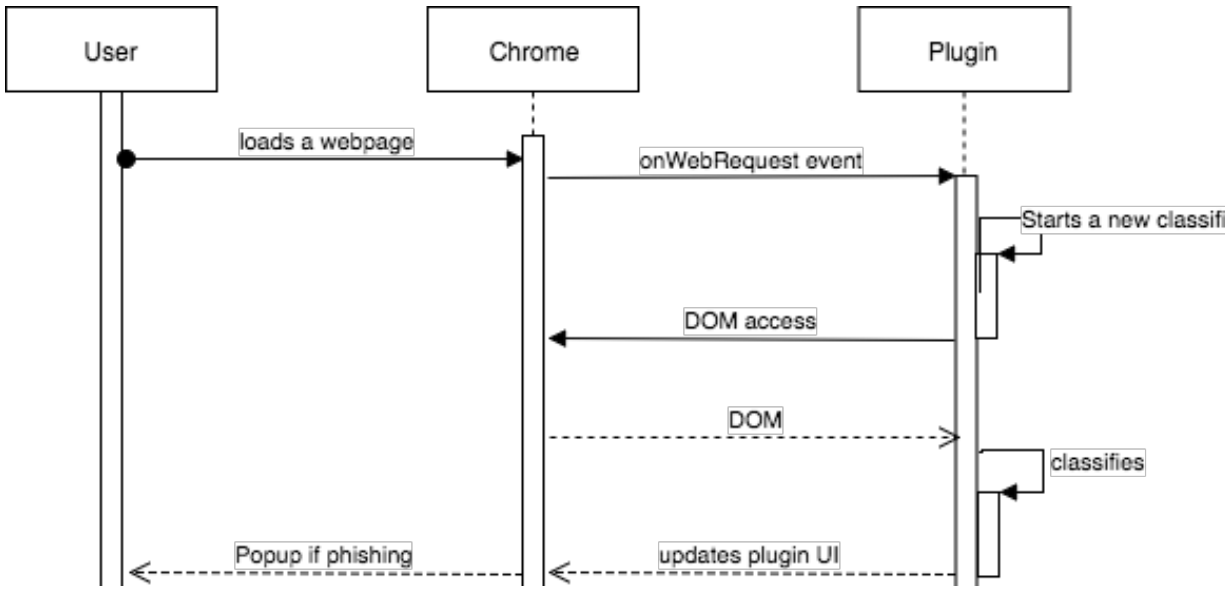

Figure 3.2 System Sequence diagram