Chapter 1

Introduction

Phishing is an attempt to obtain sensitive information such as usernames, passwords, credit card details, often for malicious reasons by disguising as a trustworthy entity in an electronic communication. Phishing 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 almost identical to the legitimate one. Communications purporting to be from social web sites, auction sites, banks, online payment processors or IT administrators are often used to lure victims. Phishing emails may contain links to websites that are infected with malware.

1.1 Description

In this project, we are going to implement a cloud-based approach for detecting phishing websites using data mining. Here, we are going to build a **chrome extension**, which must be added on the users web browser to use our functionality. Secondly, we are going to build a **classifier model**, where we can test the authenticity of the URL. This model will then be deployed on the **cloud**. Now, the main task of the chrome extension is to fetch the URL immediately when the user visits a particular website and fetch the URL attributes. These attributes are nothing but the test data to our model. The model will then give the result to the chrome extension which will prompt the user regarding the same.

1.2 Problem Formulation

Phishing detection is the major problem faced rather than its elimination, because once detected, phishing websites can be added to the blacklist, thereby avoiding any of the attacks.
The problem with the blacklisted approach is that it can detect only those websites which had performed malicious activities in the past. If there is any naive website just born to lure the attackers, it cannot be detected using blacklist approach.

1.3 Motivation

2016 saw a variety of changes in spam flows, with an increase in the number of malicious mass mailings containing ransomware being the most significant. Phishing is now the #1 delivery vehicle for ransomware and other malware [6]. Such an extensive use of ransomware may be due to the availability of this sort of malware on the black market. Currently, cybercriminals can not only rent a botnet to send out spam, they can also use Ransomware-as-a-Service. This means that the attacker may not be a hacker in the traditional sense, and may not even know how to code. In 2017 the volume of malicious spam is unlikely to fall. The entry point for the ransomware is phishing itself. This approach will make phishing detection dynamic, since newly generated phishing websites can also be detected using this approach.

1.4 Proposed Solution

In the solution mentioned, we are going to use a cloud based model for phishing website detection. The model will be based on Random Forest algorithm and it will be trained using a training dataset and this model will be deployed on the cloud, which directly communicates with the chrome extension (or users web browser). Random Forest Classifier is explained in the Appendix. The detection of the phishing website will be based on URL attributes such as age of domain, PageRank, Google Index, Web Traffic, having @ symbol, number of dots, and many more. Our dataset consists of total of 29 attributed divided into four categories (Address bar based features, Abnormal based features, HTML and JavaScript based features and domain based features).
Following are the features of our dataset:

1. having_IP_Address {-1,1}: checks whether domain part of the URL has an IP address or not.
2. URL_Length {1,0,-1}: checks whether length of the URL is greater than or equal to 54 characters.
3. Shortening_Service {1,-1}: checks whether the URL uses shortening services.
4. having_At_Symbol {1,-1}: checks the presence of '@' symbol in the URL.
5. double_slash_redirecting {-1,1}: checks if the position of '//' in URL is greater than 7.
6. Prefix_Suffix {-1,1}: checks the presence of '-' in the URL.
7. having_Sub_Domain {-1,0,1}: checks if there exists 2 or more than 2 dots in the URL.

8. SSLfinal_State {-1,1,0}: checks if the URL uses HTTPS and the issuer is trusted and the age of certificate is greater than 1 year.

9. Domain_registeration_length {-1,1}: checks if the domain expires in less than 1 year.

10. Favicon {1,-1}: checks whether the favicon is loaded from external URL.

11. port {1,-1}: checks if a particular service (e.g. HTTP) is up or down on a specific server.

12. HTTPS_token {-1,1}: checks the presence of 'HTTPS ' token in domain part of a URL

13. Request_URL {1,-1}: checks whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain.

14. URL_of_Anchor {-1,0,1}: checks whether the anchor tag links to another domain.

15. Links_in_tags {1,-1,0}: checks whether the meta, script and link tags are linked to the same domain.

16. SFH {-1,1,0}: checks whether the SFH contains an empty string or 'about:blank' or it is handled by external domains.

17. Submitting_to_email {-1,1}: checks whether the personal information is submitted to an email.

18. Abnormal_URL {-1,1}: checks absence of hostname in the URL.

19. Redirect {0,1}: checks the number of times the website is redirected.

20. on_mouseover {1,-1}: checks if the onMouseOver event makes any changes to the status bar.

21. RightClick {1,-1}: checks if right click is disabled.

22. popUpWindow {1,-1}: checks if a popup windows contains text field.

23. Iframe {1,-1}: checks if the borders of the iframe are invisible.

24. age_of_domain {-1,1}: checks if the age of domain is less than 6 months.

25. DNSRecord {-1,1}: checks the absence of DNS record in the WHOIS database.

26. web_traffic {-1,0,1}: checks if the website rank is more than 1,00,000.

27. Page_Rank {-1,1}: checks if the page rank is less than 0.2.

28. Google_Index {1,-1}: checks whether the website is in Google index or not.

29. Result {-1,1}:[class label] determines whether the site is phishing or genuine.

For our system, we have chosen a cloud based model since we are using the concept of data mining for phishing detection and not the traditional blacklist approach. Also, phishing detection would be dynamic in this kind of model, since our model can also detect newly generated phishing websites. The websites detected as phished will not be added to the blacklist, rather next time when the user encounters with the same website, the testing will be done once again for that website. This is the advantage of the cloud based model. Also, the complete load of the processing is on cloud and not on the client, which will give better efficiency. Another advantage of using a cloud based model is that multiple users can access the system simultaneously.

1.5 Scope of the Project

This project is intended to be used to detect phishing websites and not to prevent them. Phishing attack prevention is solely on the user. Our job is just to alert the user whenever we find the URL as malicious. Further tasks are completely under users responsibility. Secondly, our project does not cover phishing email detection. Our project solely focuses on phishing website detection. Since we are building a chrome extension, it will work only on Chrome. Also, the project will not work on the URL requests made from mobile phones.