BTP Report

Patel Shahil Manishbhai - 200010039

Aim:

To classify URLs as Legitimate or phishing, using image processing and favicon similarity.

Objective:

As a part of my summer project, I employed machine learning techniques to classify websites as potentially phishing or legitimate.

As part of that project, we used certain features based on URL processing, source code processing, and 3rd party features to classify a website as legitimate or phishing. Also, we used various classification algorithms such as Random Forest, Decision Trees, Logistic Regression, KNN, Neural Classifier, and SVM.

In continuing that project, the present study focuses on an extended approach. This comprehensive approach involves the use of website favicons and screenshots to extract a variety of features. We aim to generate a cumulative score for each website by combining the individual scores of these features and applying various parameters. The ultimate goal is to employ this cumulative score to classify websites as 'phishing' or 'legitimate.'

Approach:

My approach can accept a single URL as well as a CSV file containing multiple URLs as input. The process uses the detection of logo similarity with legitimate sites, detection of input boxes in a suspicious web page, and matching the domain name with the domain names of the top (3) URLs obtained by searching the 'Suspicious Domain + Top 5 most frequently repeated terms' on the search engine. I will give a negative score if any phishy feature is detected and a positive if the features are legitimate. A cumulative score will be maintained, which will help to classify the website as legitimate or phishing.

Data collection:

For data collection, we initiated the process by procuring an extensive list of the top 1 million domain names from Tranco, a renowned and reputable source known for its expertise in ranking web domains. This meticulously curated list forms the fundamental cornerstone of our dataset, underpinning our research.

To acquire the associated website favicons, we harnessed the capabilities of web scraping techniques. Our choice of tool for this endeavor was the widely acclaimed BeautifulSoup library, a Python library expressly designed for the task of web scraping. This library enabled us to extract the website favicons from the diverse domain names proficiently.

Our data collection efforts commenced with a targeted focus on extracting favicons from the top 200 domains listed in our dataset. This subset of domains represents a strategically chosen selection, comprising some of the most prominent and recognizable websites in the digital landscape. Subsequently, we expanded our scope to encompass the top 1000 domains, thereby broadening the scope of our data collection initiative. This expansion was driven by our aspiration to create a more expansive and comprehensive dataset.

The decision to commence with the retrieval of favicons from the top 200 domains was rooted in empirical findings gleaned from existing research in the domain of website classification utilizing deep learning techniques. Notably, these studies have underscored a common tactic adopted by phishing websites in their endeavor to impersonate legitimate online platforms. This strategy often involves the usage of favicons that closely resemble those of well-established and trusted websites.

The top 200 domains, in particular, emerged as prime targets for phishing attempts due to their heightened visibility and recognition. Consequently, our data collection approach prioritized this subset to discern discernible trends and similarities in utilizing favicons by both genuine and potentially malicious websites.

Subsequently, the ambit of our data collection was expansively broadened to encompass the top 1000 domains. This extension was executed to engender a dataset that offered a broader spectrum of websites and their corresponding favicons for analysis. While our initial focal point pertained to the top 200 domains, we acknowledge that phishing endeavors can encompass a more extensive array of websites. Therefore, we designed our approach to be inherently scalable, with the potential for further expansion to encompass logos from the top 1 million domains. This prospect would facilitate an even more exhaustive examination of website favicons and their potential role in phishing attempts.

In total, the types of favicons extracted encompassed a diverse range, including ***'icon,' 'apple-touch-icon,' 'shortcut icon,' 'mask-icon,' 'fluid-icon,' 'manifest,' 'yandex-tableau-widget,' 'apple-touch-startup-image,' 'apple-touch-icon-precomposed,' 'ICON,' 'SHORTCUT ICON,'***

'APPLE-TOUCH-ICON,' 'MANIFEST,' 'MASK-ICON,' and ***'FLUID-ICON.***' These various types were meticulously collected and cataloged as part of our comprehensive data collection initiative."

Methodology:

- ***Initiation and Preprocessing:***
    - Commencing the process, we initiate the classification task by addressing the input URL (a single URL from the user or a CSV file containing multiple URLs), which we seek to categorize as either legitimate or potentially phishing. We eliminate any leading or trailing whitespace within the domain name to ensure data consistency.
- ***Full-Page Screenshot Retrieval:***
    - We acquire a full-page screenshot of the target URL, facilitated through Selenium and Chrome. The code operates in headless mode, that is it runs in the background without popping up a visible browser window. The captured screenshot is stitched together from multiple viewport screenshots to create a full-page image.
    - The stitched screenshot is then systematically archived within the designated directory.
- ***Keyword Detection via Optical Character Recognition (OCR):***
    - Utilizing Pytesseract OCR and OpenCV, we identify keywords within the captured screenshot. The recognized keywords are pivotal indicators that prompt user interactions, an integral attribute of phishing websites. The key terms encompass elements such as 'username,' 'password,' 'email,' 'input,' 'textbox,' 'form,' 'sign in,' 'sign-in,' 'log-in,' 'login,' 'phone,' 'OTP,' 'CVV,' 'pin,' 'card,' and 'account.'
    - The OCR technique, combined with openCV, facilitates image-to-text conversion, thus enabling the identification of the specified keywords.
- ***Extraction of Most Occurring Terms:***
    - Our subsequent endeavor involves retrieving the top five most recurring terms from both the screenshot and the source code of the URL. While extracting the most frequent terms, we won't consider the most common terms, such as pronouns, prepositions, and articles.
    - The screenshot text is again subjected to Pytesseract for conversion while the source code is analyzed to identify high-frequency terms.
    - Integrating screenshots and source code serves as a comprehensive strategy, enhancing the efficiency of phishing website detection. This approach proves particularly valuable when encountering deceptive websites that employ images as a primary communication medium rather

than textual content. Capturing the visual attributes of a webpage extends our detection capability, especially when dealing with deceptive, image-centric elements.

- ***Google Search Query:***
    - Upon obtaining the top occurring terms from the screenshot and source code, we compile a unique search query for the search engine. This query, composed of the input domain name and the identified top terms, is then submitted for search engine query execution.
- ***Search Engine Results:***
    - The search results yield the top three outcomes, and the respective domain names are registered within our records.
- ***Checking the presence of the suspicious domain name in the search engine results:***
    - Once the results returned by the search engine are saved in a set, we check if the suspicious domain is present in the search results; if not, we decrease the weighted score. If the suspicious domain is present in the search results, we increment the weighted score.
- ***Favicon Retrieval:***
    - We scrape and download a website's favicon (shortcut icon) from the URL. We locate and download the favicon from various sources, including HTML metadata and common default paths.
- ***Favicon Similarity Detection:***
    - In our system, we utilize the Structural Similarity Index (SSIM) to gauge the similarity between the favicon of a suspicious URL and the favicon stored in our database, which exclusively contains favicon from legitimate websites. This metric assists us in evaluating the extent of resemblance between the two images.
    - To maintain a high level of accuracy and reliability in our assessments, we have set a predefined threshold of 75%. Specifically, if the SSIM score, which quantifies the structural similarity, surpasses this threshold, we mention it in the report.
    - The report will include both the similarity score itself and the name of the legitimate domain from our database that corresponds to the favicon responsible for crossing the threshold.
    - This approach serves as an integral part of our detection measures, enabling us to swiftly identify any potential anomalies or fraudulent activities associated with URLs and their respective favicons. By establishing a strict threshold and cross-referencing the data with our database of known legitimate sites, we enhance the precision of our detection system, aiding in the prompt identification of suspicious URLs.

- ***Comprehensive Reporting:***
    - Concluding the evaluation, we compile comprehensive reports for each URL (whether classified as legitimate or phishy). The report is helpful to educate the user on why the URL is considered Phishy or Legitimate. It also helps to cross-verify the final classification by manually checking the features reported in the file.
      The report will encompass the following:
        - URL classification as legitimate or phishing based on the final weighted score.
        - A delineation of the parameters influencing the classification.

Future work:

- Instead of converting the image to text and subsequently identifying keywords for user input, developing an algorithm for language detection within the screenshot is recommended. Later, based on the detected language, the algorithm should ascertain keywords that necessitate user input on the website.
- Following executing a search query on the search engine, combining the input domain name with the top-occurring terms, the current approach primarily focuses on assessing domain name similarity. However, plans involve extending this approach to encompass a comparative analysis of screenshots derived from the top URL results provided by the search engine in conjunction with the screenshot of the input URL.
- Instead of using the structural similarity approach to calculate the similarity score between the suspicious URL's favicon and the favicons of the legitimate domains, we can use deep learning approaches to calculate the similarity score. (Earlier, I used a deep learning approach to calculate the similarity score. However, the model performed poorly and detected similarity of around 90-95% in two favicons which aren't similar visually, So this issue has to be addressed).
- As the present approach is inherently time-consuming, a proposal is to introduce user-configurable flags that permit enabling or disabling specific functionalities. This feature becomes particularly relevant when users need more computational resources to execute deep learning techniques or seek expeditious results.
- Put this all together to make a browser extension. This extension's primary objective is to classify the URL visited by the user as phishing or legitimate and impart knowledge regarding the mechanics of phishing attacks while concurrently highlighting the distinctive features that indicate a potential phishing website.