Detecting Malicious Web Pages to Provide User A Safe Browsing Experience

SYEDA MAHVEEN MINHAJ[1], T. SAI KUMARI[2]

[1]PG Scholar, Dept of CSE, Shadan Women's College of Engineering and Technology, Hyderabad, TS, India,

[2]Assistant Professor, Dept of DS & CE, JNTU College of Engineering and Technology, Karimnagar, Hyderabad, TS, India,

Abstract: Internet has become a crucial part of our life and using internet is a regular activity in our day to day life. With internet being accessed by millions of users, hacking systems have become easy for intruders. There are thousands of sites that contain malicious content thus providing user a safe browsing experience is very important. Though there are existing tools and techniques that use dynamic features to detect maliciousness of a desktop based website but the techniques used for desktop cannot be used for mobile as data, structure and functionality will be different. As the available techniques uses dynamic features to deliver deeper perceptibility into the behavior of the webpage and have less false positive rate and are more precise. However, performing actions on each and every page with this technique obstructs the performance. Thus a new technique called kAYO has been developed, that uses static features of the webpage to detect the whether a page is malevolent or benevolent and this technique can be used for both desktop and mobile. The technique warns the user about the maliciousness of the page and blocks the content that is unsafe for the user.

Keywords*:* Support Vector Machines (SVM), Internet Explorer (IE), URL, HTML.

I. INTRODUCTION

Cell phones are progressively being utilized to get to the web. Notwithstanding, disregarding huge advances in processor power and data transmission, the perusing knowledge on cell phones is significantly extraordinary. These qualifications can, as it were, be credited to the thrilling lessening of screen gauge, which impacts the substance, value and configuration of adaptable site pages. Content, Functionality and Layout have routinely been used to perform static examination to choose threat in the desktop space[4]. Features, for instance, the repeat of iframes and the amount of redirections have by and large filled in as strong pointers of threatening objective. Due to the significant changes made to oblige cell phones, such attestations may never again be valid. For instance, though such conduct would be flagged as suspicious in the desktop setting, numerous prominent favorable versatile website pages require different redirections before client's access content Past techniques furthermore disregard to consider adaptable specific site page parts, for instance, calls to compact APIs. For example, relates that convey the telephone's dialer (and the notoriety of the number itself) can give solid certification of the craving for the page. New instruments are in this way fundamental to perceive destructive pages in the versatile web. kAYO1, a speedy and strong static examination strategy to recognize malicious flexible pages. kAYO uses static features of convenient pages got from their HTML and JavaScript substance, URL and pushed adaptable specific limits. We first probably show that the transports of indistinct static features when removed from desktop and flexible site pages vary radically. We by then accumulate in excess of 350,000 convenient kind and harmful site pages over a period of three months. We by then use a binomial classification methodology to develop a model for kAYO to give 90% precision and 89% honest to goodness positive rate. kAYO's execution organizes or outperforms that of existing static strategies used in the desktop space. kAYO additionally distinguishes various malevolent portable website pages not definitely identified by existing strategies, for example, VirusTotal[2] and Google Safe Browsing. At long last, we examine the restrictions of existing devices to identify versatile malignant site pages and fabricate a program expansion in light of kAYO that gives real-time input to portable program clients.

We make the following contributions:

Experimentally Demonstrate The Differences In The "Security Features" Of Desktop And Mobile Web Pages: We tentatively exhibit that the circulations of static highlights utilized in existing techniques[5] (e.g., the number of redirections) are diverse when estimated on portable and desktop site pages. In addition, we outline that specific highlights are conversely related or disconnected to or non-characteristic to a website page being noxious when removed from each space. The outcomes of our examinations show the necessity for adaptable particular methodology for perceiving pernicious site pages.

Design And Implement A Classifier For Malicious And Benign Mobile Web Pages: We gather more than 350,000 amiable and noxious versatile website pages. We at that point recognize new static highlights from these site pages that recognize portable generous and malignant site pages. kAYO gives 90% precision in classification and shows change of two requests of greatness in the speed of highlight extraction over comparable existing strategies. At last, we additionally recognize 173 versatile pages executing cross-channel assaults, which endeavor to actuate portable clients to call numbers related with known misrepresentation crusades.

Implement a Browser Extension Based on kAYO: To the best of our insight kAYO is the first strategy that distinguishes portable specific noxious website pages by static investigation. Existing apparatuses, for example, Google Safe Browsing are not empowered on the versatile renditions of programs, in this way blocking portable clients. Additionally, the versatile specific outline of kAYO empowers identification of malevolent portable website pages missed by existing procedures. At long last, our study of existing expansions on Firefox desktop program proposes that there is a lack of instruments that assistance clients recognize portable noxious website pages. To fill this void, we fabricate a Firefox versatile program expansion utilizing kAYO, which educates clients about the malignance of the pages they plan to visit progressively. We intend to make the augmentation freely accessible post production.

We take note of that we define perniciousness extensively, as is done in the earlier writing on the static recognition in the desktop space[4]. In any case, in light of the fact that drive by-downloads are not in the slightest degree basic in the versatile space at the season of composing, the mind greater part of distinguished pages are identified with phishing.

II. RELATED WORK

Past strategies neglected to consider versatile particular site page parts acknowledge calls to portable arthropod sort. To illustrate, joins that brood the telephone's dialer (and the name of the sum itself) will offer hearty confirmation of the aim of the page. New apparatuses zone unit so important to spot vindictive pages inside the versatile web. The inclination to make kAYO1, a brisk and solid static investigation strategy to see malignant portable pages. Thump cool uses static choices of portable website pages got from their hypertext markup dialect and JavaScript content, PC address and propelled versatile particular abilities. We introductory by experimentation show that the appropriations of indistinguishable static choices once removed from desktop and portable site pages shift significantly. We gathered in excess of 350,000 portable benign and malicious site pages over a measure of 3 months. We tend to then utilize a binomial characterization strategy to build up a model for thump chilly to deliver ninetieth exactness and eighty-nine genuine positive rate. kAYO's execution coordinates or surpasses that of existing static strategies utilized in the desktop territory. Thump cool conjointly identifies assortment of malignant versatile site pages not precisely recognized by existing strategies acknowledge Infection Aggregate and Google Safe Perusing. At last, we tend to examine the confinements of existing instruments to see portable noxious website pages and construct a program expansion bolstered thump chilly that gives ongoing criticism to versatile program clients. We tend to make the resulting commitments. Mobile websites are significantly different from their desktop counterparts in content, functionality and layout. Consequently, existing tools using static features to detect malicious desktop web pages are unlikely to work for mobile web pages. We explain four factors that motivate building separate static analysis techniques to detect malicious mobile web pages.

A. Differences Between Mobile And Desktop Websites

We have focused on websites built for desktop browsers in the past. Mobile browsers have been shown to differ from their desktop counterparts in terms of security. Although differences in mobile and desktop websites have been observed before, it is unclear how these differences impact security. Furthermore, the threats on mobile and desktop websites are somewhat different. Static analysis techniques using features of desktop webpages have been primarily studied for drive-by-downloads on desktop websites, whereas, the biggest threat on the mobile web at present is believed to be phishing [1]. Efforts in mitigating phishing attacks on desktop websites include isolating browser applications of different trust level, email filtering[3], using content-based features and blacklists. The best-known non-proprietary content-based approach to detect phishing webpages is Cantina[6]. Cantina suffers from performance problems due to the time lag involved in querying the Google search engine. Moreover, Cantina does not work well on webpages written in languages other than English. Finally, existing techniques do not account for new mobile threats such as known fraud phone numbers that attempt to trigger the dialer on the phone. Consequently, whether existing static analysis techniques to detect malicious desktop websites will work well on mobile websites is yet to be explored.

- **Differences in Content:** Mobile websites are often simpler than their desktop counterparts. Therefore, the distribution of content-based static features (such as the number of JavaScripts) on mobile webpages differs from that of desktop webpages. Mobile webpages do not have any iframes, whereas the corresponding desktop webpages have multiple iframes. Desktop webpages have more Javascripts than mobile webpages. Due to the simplicity of mobile webpages, the majority of other content related static features used in existing techniques[5] including, the number of images, page length, the number of hidden elements, and the number of elements with a small area also differ in magnitude.
- **Infrastructure:** Website providers use JavaScript or user agent strings to identify and then redirect mobile users to a mobile specific version. Even the most popular mobile websites show multiple redirects, which has traditionally been a property of desktop websites hosting malware. However, multiple redirects do not necessarily indicate bad behavior for mobile websites due to the characteristics of their hosting infrastructure. We note that not all static features used in existing techniques[5] differ when measured on mobile and desktop webpages.
- **Impact of Screen Size:** The screen size of a mobile phone is significantly smaller than that of a desktop computer. Therefore, a mobile user only sees a part of the URL of a webpage. Intuitively, the author of a mobile phishing webpage may only need to include misleading words at the beginning of the URL and a short URL might suffice to trick a user.
- **Mobile Specific Functionality:** Mobile websites enable access to a user’s personal information and advanced capabilities of mobile devices through web APIs. Existing static analysis techniques do not consider these mobile specific functionalities in their feature set. We argue and later demonstrate that accounting for the mobile specific functionalities helps identify new threats specific to the mobile web. For example, the presence of a known ‘bank’ fraud number on a website might indicate that the webpage is a phishing webpage imitating the same bank.

II. MOTIVATION

A. System Architecture

Dynamic methodologies utilizing virtual machines and honey client frameworks give further perceivability into the conduct of a page. Along with this, such structure has a low false positive rate and are more accurate. Nonetheless, downloading and executing every site page impacts execution and frustrates adaptability of dynamic methodologies. This execution punishment can be kept away from by utilizing static methodologies. Static methodologies depend on the basic and lexical properties of a website page and don't execute the substance of the page. One such procedure of distinguishing pernicious URLs is utilizing factual techniques for URL classification in view of a URL's lexical and host-based properties. Utilizing HTML and JavaScript highlights separated from a site page notwithstanding URL classification helps address this downside and gives better outcomes. Static methodologies maintain a strategic distance from execution punishment of dynamic methodologies. Furthermore, utilizing quick and solid static ways to deal with distinguish considerate site pages can stay away from costly top to bottom examination all things considered. Mobile websites are significantly different from their desktop counterparts in content, functionality and layout. Consequently, existing tools using static features to detect malicious desktop webpages are unlikely to work for mobile webpages. We explain four factors that motivate building separate static analysis techniques to detect malicious mobile webpages.

Limitations of Existing Techniques: These disparities among versatile and desktop website pages' request examination. Existing static examination frameworks and gadgets for recognizing noxious site pages are based on desktop site pages. Along these lines, they can't recognize flexible specific perils with high precision. Also, a few website pages fabricated specifically for versatile, return void pages when rendered in a desktop program. Thusly, despite existing one of a kind examination methodology that execute locales in desktop programs on virtual machines, are incapable on such convenient destinations. Finally, signature based instruments, for instance, Google Safe Perusing at present simply work with desktop programs. We physically visited five portable specific known vindictive pages gathered from PhishTank, from the Google Chrome versatile program. We saw that these pages are flagged as malignant on the Chrome desktop program, yet not on the Chrome versatile program whose clients are the genuine focuses of the mobile noxious site pages. Albeit empowering Google Safe Browsing in portable chrome is a building exertion, we contend and later exhibit that a versatile specific static strategy can likewise recognize new dangers beforehand inconspicuous by such administrations.

III. METHODOLOGY

The goal is to outline and build up a method to recognize portable specific malignant pages continuously. We extricate static highlights from a website page and make forecasts about its potential noxiousness.

A. kAYO Feature Set

A site page has a few parts including HTML and JavaScript code, pictures, the URL, and the header. Mobile specific website pages additionally get to applications running on a client's gadget utilizing web APIs (e.g., the dialer). We remove basic, lexical and quantitative properties of such segments to produce kAYO's list of capabilities. We center around separating portable pertinent highlights that take insignificant extraction time. Our theory is that such highlights are solid markers of whether a site page has been worked for helping a client in their web perusing knowledge or for pernicious purposes. The list of capabilities comprises of 44 highlights, 11 of which are new and not beforehand identified or utilized. A subset of highlights in kAYO have been utilized in static review of desktop site pages previously. Nonetheless, it is critical that these highlights in versatile site pages and desktop site pages' contrast in size (e.g., number of iframes) and show fluctuating relationship with the idea of the website page (i.e., malignant/amiable). kAYO's 44 features into four classes:

Mobile Specific: Eight mobile specific features have been collected to capture the advanced capabilities of mobile web pages. Mobile websites enable access to personal data from a user's phone, an experience not offered by desktop websites. For instance, portable web APIs, example, tel: and sms: produce the dialer and the SMS submissions independently on a mobile phone. To portray the conduct of mobile Programming interface calls, we separated the quantity of Programming interface calls tel:, sms:, smsto:, mms: and mmsto: from every versatile site page.

JavaScript: JavaScript empowers customer side client connection, non-concurrent correspondence with servers, and modification of the DOM objects of pages on the fly. 10 features have been extracted that captures the JavaScript applicable static conduct of a website page, two of which are new. Every one of the highpoints is faster to rescue than the highlights in vision of JavaScript complication. JavaScript found on malevolent webpages can be obscured. In place of de-obfuscating every JavaScript, we excerpt simple JavaScript related structures from a webpage. The basic reason in choosing this method is that a great quantity of benevolent webpages includes potentially unsafe JavaScript code.

HTML: 14 features have been included altogether from the HTML code of every site page. Well known site pages incorporate various pictures, and inside and outer HTML joins for better client encounter.

URL Features: Structural and lexical features of a URL have been used to discriminate between malevolent and benign web pages. However, using only URL features for such differentiation leads to a high false positive rate.

B. Methodologies Used

Information gathering process included amassing named amiable and malignant versatile specific pages. At first, a test that identifies and defines 'portable specific site pages'. By then the data gathering process have been driven. After that these drags have been used specifically in light of the way that they are close to the dissemination of the related work, making them as close corresponding as would be realistic. Topmost webpages of well-known websites have been creeped from Alexa.com using Android mobile phones and desktop IE browsers. Then each pair of final URLs have been analyzed manually for the same set of URLs for the desktop. Before grouping a URL as portable specific, it has been confirmed that the final URLs for desktop and versatile were distinctive for a similar seed URL. It at that point analyzed the substance of each match of desktop and portable site pages, and guaranteed that the two substance were extraordinary. All the seed URLs that prompted an indistinguishable final URL when crept from the desktop and the portable program have been disregarded. Our investigation identified nine subdomains (e.g., m.) and seven URL way prefixes (e.g., /versatile) in the URLs of prominent sites to speak to their portable specific website pages. An extensive variety of paired grouping methods in machine learning have been considered, yet for space talk about three alternatives are utilized: Support Vector Machines (SVM), naive Bayes and strategic relapse.

- Support Vector Machine is a well-known paired classifier. Be that as it may, it functions admirably just on a couple of thousand examples of information. Because of the scaling issue of SVMs and our huge dataset, SVM was not the best decision for kAYO.
- Naive Bayes is generally used when the values of different features are mutually independent. Many features that we extracted were mutually dependent. For example, the number of scripts in a webpage was dependent on the number of internal, external and embedded JavaScript in the webpage, which were three other features of our model. Since the assumptions required for optimal performance of naive Bayes did not hold for our dataset, we could not use the naive Bayes classifier.
- Logistic Regression is a scalable classification technique and makes no assumption about the distribution of values of the features. Therefore, this technique was the best fit for our dataset. We used the binomial variation of logistic regression to model kAYO and employed `1-regularization to avoid overfitting of the data.

C. Data Collection

Our data gathering process included accumulating labeled benign and malicious mobile specific webpages. First, we describe an experiment that identifies and defines ‘mobile specific webpages’. We then conduct the data collection process. We use these crawls specifically because they are close to the publication of the related work, making them as close to equivalent as possible.

Identification of Mobile Specific Webpages: We crawled the top-level webpage of the 1,000 most popular websites from Alexa.com using the Android mobile and desktop Internet Explorer (IE) browsers. We used Android mobile version 4.0 and IE desktop version 9.0 for Windows 7. We then manually analyzed each pair of final URLs for the same seed URL when crawled from each browser. Before classifying a URL as mobile specific, we confirmed that the final URLs for desktop and mobile were different for the same seed URL. We also compared the contents of each pair of desktop and mobile webpages, and ensured that the two contents were different. We ignored all the seed URLs that led to an identical final URL when crawled from the desktop and the mobile browser. Our analysis identified nine sub domains (e.g., m.) and seven URL path prefixes (e.g., /mobile) in the URLs of popular websites to represent their mobile specific webpages. Additionally, we considered all URLs with the ‘.mobi’ Top Level Domain (TLD) to be mobile sites. We defined a mobile specific webpage as one containing differences in the content from the corresponding desktop webpage.

Building the Dataset: To generate training data for our model, we statically crawled the top-level webpage of the top most popular websites from Alexa from an Android mobile browser. We then extracted the mobile specific. We have gone through many of the webpages to detect the maliciousness and found out malicious webpages that can harm the system of user without him having any knowledge about it. And after finding that we found out a way for providing user a safe browsing experience.

IV. IMPLEMENTATION AND EVALUATION

We describe the machine learning techniques we considered to tackle the problem of classifying mobile specific webpages as malicious or benign. We then discuss the strengths and weaknesses of each classification technique, and the process for selecting the best model for kAYO. We build and evaluate our chosen model for accuracy, false positive rate and true positive rate. Finally, we compare kAYO to existing techniques and empirically demonstrate the significance of kAYO’s features. We note that where automated analysis is possible, we use our full datasets; however, as is commonly done in the research community, we use randomly selected subsets of our data when extensive manual analysis and verification is required.

A. Model Selection and Implementation

We treated the problem of detecting malicious webpages as a binary classification problem. We considered each known benign mobile webpage as a negative sample and each known malicious mobile webpage as a positive sample. We considered a wide range of popular binary classification techniques in machine learning, but for space discuss three popular options: Support Vector Machines (SVM), na¨ıve Bayes and logistic regression.

Support Vector Machines (SVM) is a popular binary classifier. However, it works well only on a few thousand samples of data. Due to the scaling problem of SVMs and our large dataset, SVM was not the best choice for kAYO.

Naı¨ve Bayes is generally used when the values of different features are mutually independent. Many features that we extracted were mutually dependent. For example, the number of scripts in a webpage was dependent on the number of internal, external and embedded JavaScript in the webpage, which were three other features of our model. Since the assumptions required for optimal performance of na¨ıve Bayes did not hold for our dataset, we could not use the na¨ıve Bayes classifier.

Logistic Regression is a scalable classification technique and makes no assumption about the distribution of values of the features. Therefore, this technique was the best fit for our dataset. We used the binomial variation of logistic regression to model kAYO and employed `1-regularization to avoid overfitting of the data.

We used the web scraping framework to crawl the collected mobile URLs. We then built a parser for extracting features from each input webpage dynamically. The crawler and feature extraction scripts were implemented in Python. We used logistic regression on the extracted features for training and testing.

B. Evaluation

Our dataset contained benign URLs and malicious URLs. We divided our dataset into three subsets: training, cross-validation and test. We first randomly shuffled the data and set aside 10% of the data as the test set. The remaining 90% of data was used for training and 10-fold cross-validation. For each validation round we calculated the accuracy, the false positive rate and the true positive rate on the validation set. We further used `1-regularization to avoid overfitting. We varied the regularization parameter from 0 to 1,000 in the intervals of 10 and chose the best kAYO's evaluation is very appropriate and we found this after testing so many benign and malicious webpages. kAYO not only provide the user a safe browsing experience but even alert the user with the malicious webpages which can harm the user. kAYO provided 91% true positive rate and 7% false positive rate on the cross-validation set. We used the best parameters obtained from the training and cross validation steps to test the 10% labeled dataset set aside. Our test set shows 90% accuracy, 8% false positive rate and 89% true positive rate. We believe that this rate is equivalent to that of the desktop-specific schemes, which use dramatically smaller datasets than our own, as accuracy falls significantly when datasets increase in size. We also anticipate that the false positive rate on the test set would be lower than what was found using the labeled samples because kAYO detected a number of malicious mobile URLs in the wild that we hand verified, and were not detected by tools that we used for establishing ground truth of our dataset.

Comparison with Existing Static Techniques: We have identified and used 11 new mobile-relevant features previously not studied. We note that none of the existing techniques account for mobile specific features considered in kAYO. The non-commercial static analysis technique closest to kAYO is Cantina[6]. It detects phishing webpages in real-time using static features of webpages. We obtained the desktop malicious webpages by monitoring public blacklists and crawling live URLs two links deep. We verified ground truth of these URLs using Google Safe Browsing and VirusTotal[2]. We randomly shuffled the webpages and chose few webpages while keeping the proportion of benign and malicious webpages in the dataset equivalent to the mobile dataset. We extracted 33 out of the 44 static features in kAYO from each webpage in the desktop and mobile datasets. We disregarded the 11 new mobile features used in kAYO and instead focused our analysis on the 33 features previously used in similar desktop static techniques. We note that the goal of this experiment is not to extract all desktop relevant features used earlier, but demonstrate that a model trained on features extracted from desktop webpages does not perform well when applied to mobile webpages. We believe that these 33 features accurately represent the static features used in earlier techniques to detect malicious desktop webpages. We used logistic regression with regularization to train a model on the desktop webpage dataset and tested the model on the mobile dataset. Fig.1 below shows the results of our experiments.

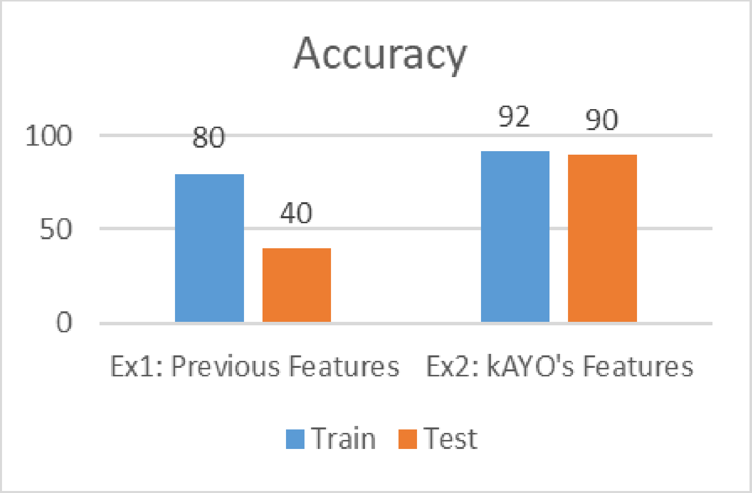


Fig.1. results of our experiments.

Ex1 shows that using 33 features, we achieved 77% accuracy in training on desktop webpages. However, when the parameters obtained from this model were applied to the mobile dataset, the accuracy reduced significantly to 40%. The difference between the accuracy of the training and testing dataset is the important comparison metric in this experiment as it demonstrates the inability of previous desktop-only models to accurately characterize mobile webpages. Ex2 simply shows kAYO’s results (discussed in Section 5.2) of training and testing on mobile webpages considering mobile specific features. Both training (91%) and testing (90%) dataset accuracies improve notably. More importantly, the accuracies of the training and testing datasets in Ex2 are comparable unlike those in Ex1. These results confirm our intuition that mobile specific static techniques are necessary - without using the new mobile features, previously proposed techniques perform poorly.

Significance of kAYO’s Feature Set: It is important to observe that kAYO’s feature set has been carefully created to ensure relevance to mobile webpages and negligible extraction time. We experimentally demonstrate the significance of kAYO’s features using the Pearson product-moment Correlation Coefficient (PCC). PCC is a measure of the linear dependence between two variables giving a value between +1 and −1 inclusive. In other words, PCC provides information about the predictive power of a feature over the classification result. The larger the absolute value of the PCC of a feature, the more its predictive power. For example, a feature with PCC -0.6 is a better predictor of whether a webpage is malicious than a feature with PCC 0.21. It is important to note that identifying features with very high PCC values is extremely difficult given the hundreds of different components of webpages and the diversity of threats. We find the PCC between each feature in kAYO’s feature set and the label (benign/malicious), from the test set used for evaluation. Intuitively, if kAYO’s features are significant, then the absolute value of the PCC of each feature with the label must be non-zero. Figure 5 shows the plot of the PCC of each of the 44 features of kAYO with the label. The circles show the PCC of the newly identified features of kAYO and the Xs depict the PCC of features adopted from earlier works.

Comparison with Existing Browser Tools: Browser extensions and plugins help protect users from visiting malicious websites. The most prevalent threat on the mobile web at present is phishing. Therefore, we surveyed the most popular anti-phishing Firefox desktop extensions for comparison with kAYO. These 33 extensions were selected by searching for the keyword ‘phishing’ on the Firefox extension store. Most of the extensions were certificate verifiers, password protectors or file protectors. We did not find any extensions performing content-based static analysis. We disregarded extensions that were built only for one specific website or were no longer supported. We also tested the freely available trial version of the Lookout safe browsing tool. Lookout is one of the most popular security applications available for mobile devices. This tool protects users of the Android mobile and the Chrome mobile browsers from phishing scams and malicious links on the mobile web. We browsed the same 10 known malicious URLs from both the Android mobile and Chrome mobile browser on a device running the Android 4.0 operating system. We were presented with alerts for only two out of the 10 URLs by Lookout, while kAYO detected eight out of the 10 webpages. Given the paucity of a working extension to detect different threats on mobile webpages, and the unavailability of signature-based tools such as Google Safe Browsing for mobile browsers, we developed a mobile browser extension using kAYO.

C. Browser Extension

Building a browser extension based on kAYO adds value for two reasons. First, the mobile specific design of kAYO existing services (e.g., pages including spam phone numbers). Second, building an extension allows immediate use of our technique. We discuss other potential avenues of adopting kAYO. We can develop a browser extension using kAYO for Firefox Mobile, which informs users about the maliciousness of the webpages they intend to visit. Our goal should be to build an extension that runs in real-time. Therefore, instead of running the feature extraction process in a mobile browser, we outsourced the processing intensive functions to a backend server.

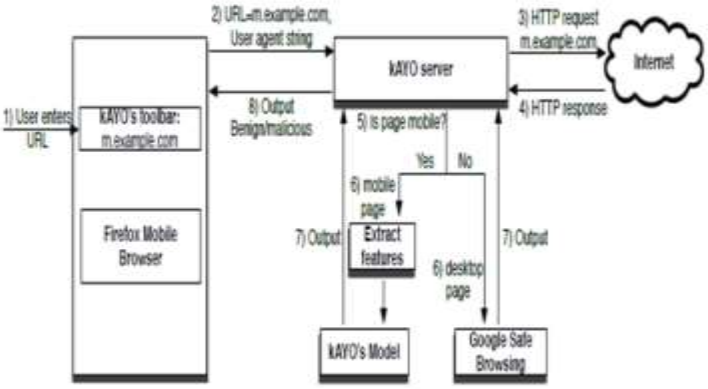


Fig.2. architecture of the extension.

Fig.2 above shows the architecture of the extension. User enters the URL he wants to visit in the extension toolbar. The extension then opens a socket and sends the URL and user agent information to kAYO’s backend server over HTTPS. The server crawls the mobile URL and extracts static features from the webpage. This feature set is input to kAYO’s trained model, which classifies the webpage as malicious or benign. The output is then sent back to the user’s browser in real-time. If the URL is benign according to kAYO, the extension renders the intended webpage in the browser automatically. Otherwise, a warning message is shown to the user recommending them not to visit the URL. Users of the extension will browse both mobile specific and desktop webpages since not all websites offer a mobile specific version. Recall that being a mobile specific technique, kAYO does not perform well on desktop webpages. Consequently, processing all pages of interest through kAYO might output incorrect results for desktop webpages. To address this problem, the backend server first detects whether the intended webpage is mobile specific using the same method explained in Section.

D. Discussion

kAYO detected a number of malicious webpages in the wild that were not found by existing techniques. We investigate these webpages in detail and then describe the limitations and future work of kAYO.

Limitations and Future Work:

- These disparities among versatile and desktop website pages' request examination. Existing static examination frameworks and gadgets for recognizing noxious site pages are based on desktop site pages. Along these lines, they can't recognize flexible specific perils with high precision. Also, a few website pages fabricated specifically for versatile, return void pages when rendered in a desktop program. Thusly, despite existing one of a kind examination methodology that execute locales in desktop programs on virtual machines, are incapable on such convenient destinations.
- Finally, signature based instruments, for instance, Google Safe Perusing at present simply work with desktop programs. We physically visited five portable specific known vindictive pages gathered from PhishTank, from the Google Chrome versatile program. We saw that these pages are flagged as malignant on the Chrome desktop program, yet not on the Chrome versatile program whose clients are the genuine focuses of the mobile noxious site pages. Albeit empowering Google Safe Browsing in portable chrome is a building exertion, we contend and later exhibit that a versatile specific static strategy can likewise recognize new dangers beforehand inconspicuous by such administrations.
- In-depth dynamic analysis of webpages may provide additional important details. However, because such approaches incur significantly higher costs, this approach conflicts with our design goal of creating a real-time detector. Accordingly, we leave the significant challenge of efficient operation of such tools to future work.
- Using signature based blacklist approaches such as Google Safe Browsing might improve the performance of kAYO's browser extension. A blacklist can be synchronized with kAYO's extension server and enforced locally. Although such techniques might reduce the average delay in page rendering, they will also preclude from protection against webpages that change dynamically defeating kAYO's goal of real-time evaluation. We plan to investigate performance enhancing designs that preserve real-time evaluation in future work shown in Figs.3 to 8.

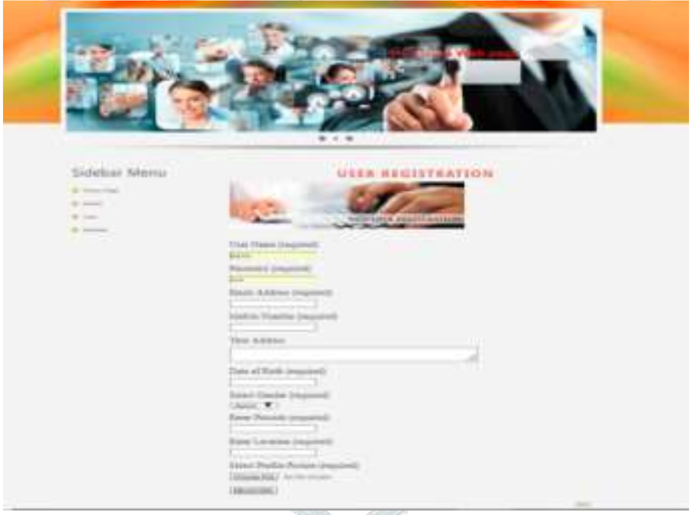


Fig.3. User Registration Page.

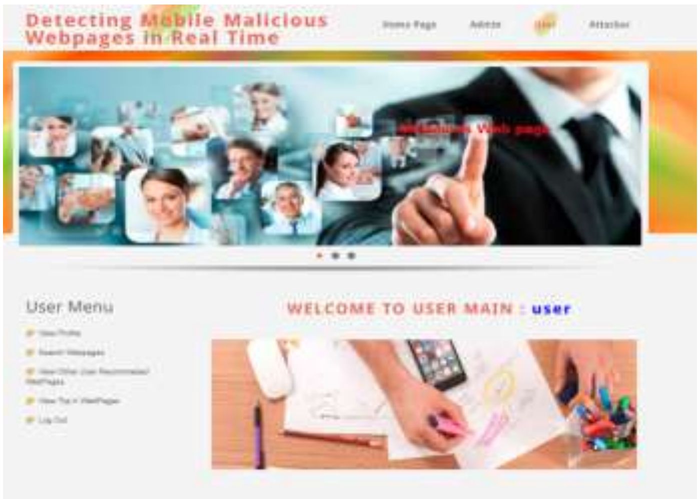

Detecting Mobile Malicious Webpages in Real Time
Home Page
Admin
User
Attacker
Malicious Web page
User Menu
View Profile
Search Webpages
View Other User Recommended WebPages
View Top K WebPages
Log Out
WELCOME TO USER MAIN : user

Fig.4. User Home Page.

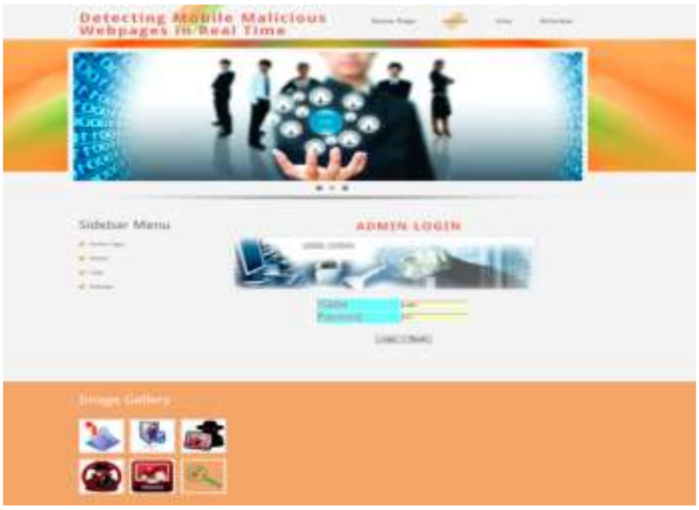


Fig.5. Admin Login Page.

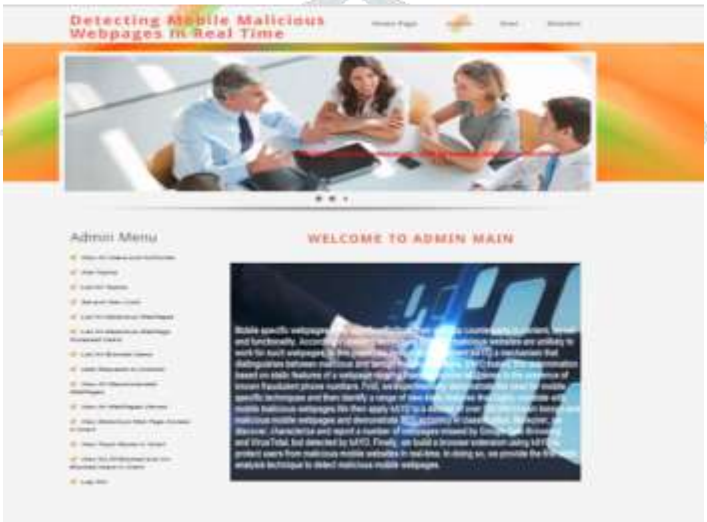


Fig.6. Admin Home Page.

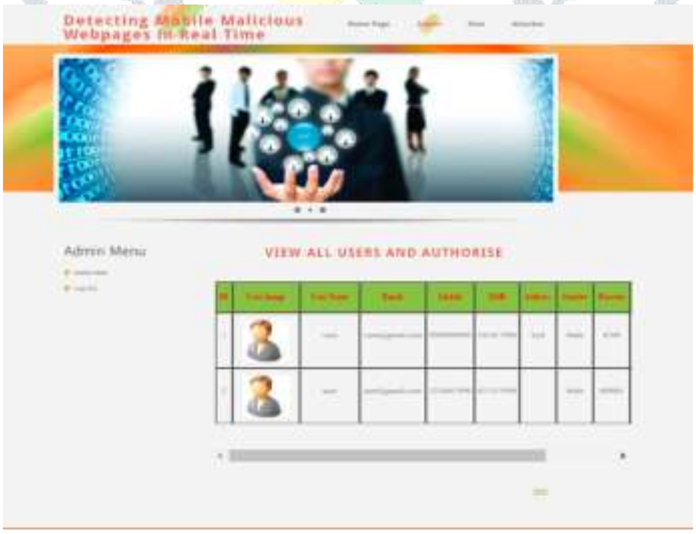
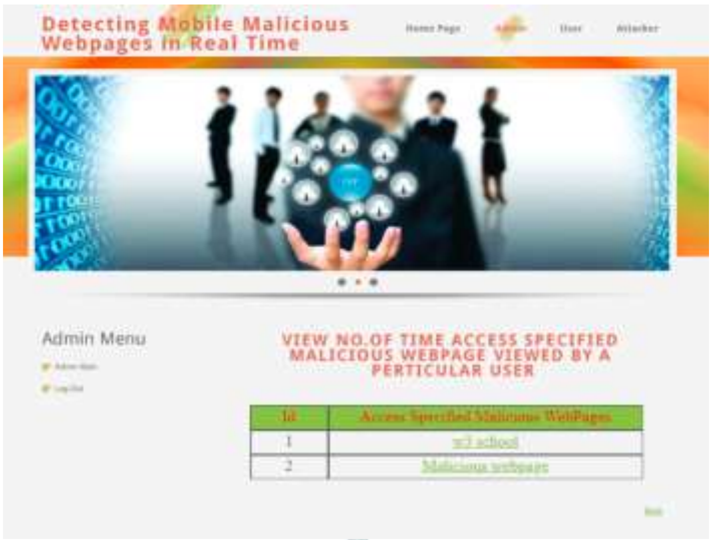

Detecting Mobile Malicious Webpages in Real Time
Admin Menu
VIEW ALL USERS AND AUTHORISE

Fig.7. Authorization Page.



Detecting Mobile Malicious Webpages in Real Time

VIEW NO.OF TIME ACCESS SPECIFIED MALICIOUS WEBPAGE VIEWED BY A PERTICULAR USER

Id	Access Specified Malicious WebPages
1	w3 school
2	Malicious webpage

Fig.8. Admin Viewing Malicious Webpages.

V. CONCLUSION

Mobile webpages are very different from desktop webpages in content, functionality and layout. Hence, the techniques available to find the maliciousness of the desktop webpage that uses static features are not effective enough to detect the maliciousness of mobile webpages. Thus, a fast and reliable technique called kAYO has been developed, which utilizes static features to find the maliciousness of the mobile webpage. kAYO detects the maliciousness by considering 44 mobile specific features from webpages, out of which 11 features are identified as newly identified mobile oriented features. kAYO gives 90% of the precision in classification and detects number of toxic webpages that are not identified using existing techniques like Google Safe Browsing and VirusTotal[2]. With this, it has been determined that kAYO perceives new mobile specific threats such as website hosting known numbers and takes the first step towards identifying new security challenges in the modern mobile world.

VI. REFERENCES

[1] M. Boodaei. Mobile users three times more vulnerable to phishing attacks. http://www.trusteer.com/blog/mobile-users-threetimes-more-vulnerable-to-phishing-attacks, 2011.

[2] VirusTotal. https://www.virustotal.com/en/.

[3] [28] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In Proceedings of the 16th International Conference on World Wide Web (WWW), 2007.

[4] P. Likarish, E. Jung, and I. Jo. Obfuscated malicious javascript detection using classification techniques. In Proceedings of Malicious and Unwanted Software (MALWARE), 2009.

[5] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In Proceedings of the SIGKDD Conference, 2009.

[6] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In Proceedings of the 16th international conference on World Wide Web (WWW), 2007.

Author's Details:

Ms. SYEDA MAHVEEN MINHAJhas completed her BE from Stanley college of Engineering and Technology for Women, Hyderabad. OU University, Hyderabad. Presently, She is pursuing her Masters in Computer Science from Shadan women's college of Engineering and technology, Hyderabad, TS. India.

Ms. T SAI KUMARI has completed B.Tech (CSE) from Shadan Women's college of engineering and technology, JNTUH University, Hyderabad. M.Tech (CSE) from Shadan women's college of engineering and technology, JNTU University, Hyderabad, Currently she is working as an Assistant Professor of IT Department in Shadan women's college of Engineering and technology, Hyderabad, TS. India.