Detecting Mobile Malicious Webpages In Real Time

Gadhamsetti Manikanta Kumar, Sri.S.K.Alisha, Sri.V.Bhaskara Murthy
Mca Student, Senior Assistant Professor, Associate Professor
Dept Of Mca
B.V.Raju College, Bhimavaram

ABSTRACT
Mobile specific webpages differ significantly from their desktop counterparts in content, layout and functionality. Accordingly, existing techniques to detect malicious websites are unlikely to work for such webpages. In this paper, we design and implement kAYO, a mechanism that distinguishes between malicious and benign mobile webpages. kAYO makes this determination based on static features of a webpage ranging from the number of iframes to the presence of known fraudulent phone numbers. First, we experimentally demonstrate the need for mobile specific techniques and then identify a range of new static features that highly correlate with mobile malicious webpages. We then apply kAYO to a dataset of over 350,000 known benign and malicious mobile webpages and demonstrate 90% accuracy in classification. Moreover, we discover, characterize and report a number of webpages missed by Google Safe Browsing and VirusTotal, but detected by kAYO. Finally, we build a browser extension using kAYO to protect users from malicious mobile websites in real-time. In doing so, we provide the first static analysis technique to detect malicious mobile webpages.

I.INTRODUCTION

Mobile devices are increasingly being used to access the web. However, in spite of significant advances in processor power and bandwidth, the browsing experience on mobile devices is considerably different. These differences can largely be attributed to the dramatic reduction of screen size, which impacts the content, functionality and layout of mobile webpages. Content, functionality and layout have regularly been used to perform static analysis to determine maliciousness in the desktop space [20], [37], [51]. Features such as the frequency of iframes and the number of redirections have traditionally served as strong indicators of malicious intent. Due to the significant changes made to accommodate mobile devices, such assertions may no longer be true. For example, whereas such behavior would be flagged as suspicious in the desktop setting, many popular benign mobile webpages require multiple redirections before users gain access to content. Previous techniques also fail to consider mobile specific webpage elements such as calls to mobile APIs. For instance, links that spawn the phone's dialer (and the reputation of the number itself) can provide strong evidence of the intent of the page. New tools are therefore necessary to identify malicious pages in the mobile web. In this paper, we present kAYO1 , a fast and reliable static analysis technique to detect malicious mobile web- pages. kAYO uses static features of mobile webpages derived from their HTML and JavaScript content, URL and advanced mobile specific capabilities. We first experimentally demonstrate that the distributions of identical static features when extracted from desktop and mobile webpages vary dramatically. We then collect over 350,000 mobile benign and malicious webpages over a period of three months. We then use a binomial classification technique to develop a model for kAYO to provide 90% accuracy and 89% true positive rate. kAYO's performance matches or exceeds

that of existing static techniques used in the desktop space. kAYO also detects a number of malicious mobile webpages not precisely detected by existing techniques such as VirusTotal and Google Safe Browsing. Finally, we discuss the limitations of existing tools to detect mobile malicious webpages and build a browser extension based on kAYO that provides realtime feedback to mobile browser users. We make the following contributions: • Experimentally demonstrate the differences in the "security features" of desktop and mobile webpages: We experimentally demonstrate that the distributions of static features used in existing techniques (e.g., the number of redirections) are different when measured on mobile and desktop webpages. Moreover, we illustrate that certain features are inversely correlated or unrelated to or non-indicative to a webpage being malicious when extracted from each space. The results of our experiments demonstrate the need for mobile specific techniques for detecting malicious webpages. • Design and implement a classifier for malicious and benign mobile webpages: We collect over 350,000 benign and malicious mobile webpages. We then identify new static features from these webpages that distinguish between mobile benign and malicious webpages. kAYO provides 90% accuracy in classification and shows improvement of two orders of magnitude in the speed of feature extraction over similar existing techniques. We further empirically demonstrate the significance of kAYO's features. Finally, we also identify 173 mobile webpages implementing cross-channel attacks, which attempt to induce mobile users to call numbers associated with known fraud campaigns. • Implement a browser extension based on kAYO: To the best of our knowledge kAYO is the first technique that detects mobile specific malicious webpages by static analysis. Existing tools such as Google Safe Browsing are not enabled on the mobile versions of browsers, thereby precluding mobile users. Moreover, the mobile specific design of kAYO enables detection of malicious mobile webpages missed by existing techniques. Finally, our survey of existing extensions on Firefox desktop browser suggests that there is a paucity of tools that help users identify mobile malicious webpages. To fill this void, we build a Firefox mobile browser extension using kAYO, which informs users about the maliciousness of the webpages they intend to visit in real-time. We plan to make the extension publicly available post publication. We note that we define maliciousness broadly, as is done in the prior literature on the static detection in the desktop space [20], [37], [51]. However, because driveby-downloads are not at all common in the mobile space at the time of writing, the overwhelming majority of detected pages are related to phishing.

II.EXISTING SYSTEM

- A popular approach in detecting malicious activity on the web is by leveraging distinguishing features between malicious and benign DNS usage.
- Both passive DNS monitoring and active DNS probing methods have been used to identify malicious domains. While some of these efforts focused solely on detecting fast flux service networks, another can also detect domains implementing phishing and drive-by-downloads.
- The best-known non-proprietary content-based approach to detect phishing webpages is Cantina

DISADVANTAGES OF EXISTING SYSTEM:

- Existing tools such as Google Safe Browsing are not enabled on the mobile versions of browsers, thereby precluding mobile users.
- DNS based mechanisms do not provide deeper understanding of the specific activity implemented by a webpage or domain.

- Downloading and executing each webpage impacts performance and hinders scalability of dynamic approaches.
- URL-based techniques usually suffer from high false positive rates.
- 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.

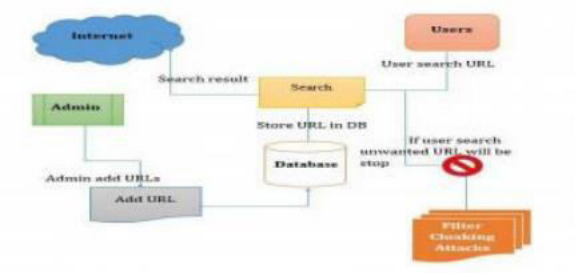
III.PROPOSED SYSTEM

- In this paper, we present kAYO, a fast and reliable static analysis technique to detect malicious mobile web-pages. kAYO uses static features of mobile webpages derived from their HTML and JavaScript content, URL and advanced mobile specific capabilities.
- We first experimentally demonstrate that the distributions of identical static features when extracted from desktop and mobile webpages vary dramatically
- We experimentally demonstrate that the distributions of static features used in existing techniques (e.g., the number of redirections) are different when measured on mobile and desktop webpages. Moreover, we illustrate that certain features are inversely correlated or unrelated to or non-indicative to a webpage being malicious when extracted from each space.

ADVANTAGES OF PROPOSED SYSTEM:

- kAYO also detects a number of malicious mobile webpages not precisely detected by existing techniques such as VirusTotal and Google Safe Browsing.
- The results of our experiments demonstrate the need for mobile specific techniques for detecting malicious webpages.
- To the best of our knowledge kAYO is the first technique that detects mobile specific malicious webpages by static analysis.
- Moreover, the mobile specific design of Kayo enables detection of malicious mobile webpages missed by existing techniques.
- Finally, our survey of existing extensions on Firefox desktop browser suggests that there is a paucity of tools that help users identify mobile malicious webpages.

IV.SYSTEM ARCHITECTURE



V.MODULES:

- System Model
- Malicious Pages
- Identifying relevant static features
- Detect malicious mobile webpages

MODULES DESCSRIPTION:

System Model

In the first module, we develop the System environment model. Website providers use JavaScript or user agent strings to identify and then redirect mobile users to a mobile specific version. We note that not all static features used in existing techniques differ when measured on mobile and desktop webpages. 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

Malicious Pages

We argue that benign webpage writers take effort to provide good user experience, whereas the goal for malicious webpage authors is to trick users into performing unintentional actions with minimal effort. We therefore examine whether a webpage has noscript content and measure the number of noscript. Intuitively, a benign webpage writer will have more noscript in the code to ensure good experience even for a security savvy user.

Identifying relevant static features

We extract static features from a webpage and make predictions about its potential maliciousness. We first discuss the feature set used in kAYO followed by the collection process of the dataset. Structural and lexical properties of a URL have been used to differentiate between malicious and benign webpages. However, using only URL features for such differentiation leads to a high false positive rate.

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.

Detect malicious mobile webpages

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.

VI.CONCLUSION

Mobile webpages area unit considerably completely different than their desktop counterparts in content, practicality and layout. Therefore, existing techniques exploitation static options of desktop webpages to notice malicious behavior don't work well for mobile specific pages. We have a tendency to designed and developed a quick and reliable static analysis technique referred to as knock cold that detects mobile malicious webpages. Knock cold makes these detections by activity forty four mobile relevant options from webpages, out of that eleven area unit recently known mobile specific options. Knock cold provides ninetieth accuracy in classification, and detects variety of malicious mobile webpages within the wild that don't seem to be detected by existing techniques appreciate Google Safe Browsing and Virus Total. Finally, we have a tendency to build a browser extension exploitation knock cold that gives time period feedback to users. we have a tendency to conclude that knock cold detects new mobile specific threats appreciate internet sites hosting identified fraud numbers and takes the primary step towards distinguishing new security challenges within the trendy mobile web.

REFERENCES

[1] Chaitrali Amrutkar, Young Seuk Kim and Patrick Traynor, “Detecting Mobile Malicious Webpages in Real Time,” IEEE Trans. Services Computing, IEEE, 2017.

[2] Shuang Liang, Yong Ma and Yong Ma, “The Scheme of Detecting Encoded Malicious WebPages Based on Information Entropy”, IEEE, 2016.

[3] Xi Xiao Ruibo Yan, and H. Yan Runguo Ye, “Detection and Prevention of Code Injection Attacks on HTML5-based Apps,” IEEE,2016.

[4]Lookout.https://play.google.com/store/apps/details?hl=en&id=c om.lookout.

[5] Malware Domains List. http://mirror1.malwaredomains.com/files/domains.txt.

[6] Phish tank. http://www.phishtank.com/.

[7] Pin drop phone reputation service. http://pindropsecurity.com/phone-fraudsolutions/phone reputation service prs/.

[8] Scrapy — an open source web scraping framework for python. http://scrapy.org/.

[9] Virus Total. https://www.virustotal.com/en/.

[10] Google developers: Safe Browsing API. https://developers.google.com/Safe browsing/, 2012.