Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 4, April 2018, pg.76 – 81

DETECTION OF ANOMALOUS WEBPAGES

Arpitha G, Radhakrishna Dodmane
[1]Computer Science and Engineering, NMAMIT, Nitte, VTU, India
[2]Associate Professor, Computer Science and Engineering, NMAMIT, Nitte, VTU, India
[1] arpithag.achar@gmail.com
[2] radhakrishna@nitte.edu.in

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. 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. 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.

Keywords- Mobile security, webpages, web browsers, machine learning.

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 [2], [3]. 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. 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.

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.

II. RELATED WORK

Dynamic approaches using virtual machines [5] and honeyclient systems [3], [2] provide deeper visibility into the behavior of a webpage. Therefore, such systems have a very low false positive rate and are more accurate. However, downloading and executing each webpage impacts performance and hinders scalability of dynamic approaches. This performance penalty can be avoided by using static approaches. Static approaches rely on the structural and lexical properties of a webpage and do not execute the content of the webpage. One such technique of detecting malicious URLs is using statistical methods for URL classification based on a URL's lexical and host-based properties [7]. Static approaches avoid performance penalty of dynamic approaches. Additionally, using fast and reliable static approaches to detect benign webpages can avoid expensive in-depth analysis of all webpages. Although differences in mobile and desktop websites have been observed before [9], it is unclear how these differences impact security. Furthermore, the threats on mobile and desktop websites are somewhat different [6]. Static analysis techniques using features of desktop webpages have been primarily studied for drive-by-downloads on desktop websites [10], [11], whereas, the biggest threat on the mobile web at present is believed to be phishing. Efforts in mitigating phishing attacks on desktop websites include isolating browser applications of different trust level, email filtering, using content-based features and blacklists.

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.

III. ARCHITECTURE

A webpage has several components including HTML and JavaScript code, images, the URL, and the header. Mobile specific webpages also access applications running on a user's device using web APIs (e.g., the dialer). We extract structural, lexical and quantitative properties of such components to generate kAYO's feature set. We focus on extracting mobile relevant features that take minimal extraction time. Our hypothesis is that such features are strong indicators of whether a webpage has been built for assisting a user in their web browsing experience or for malicious purposes.

Our feature set consists of 44 features, 11 of which are new and not previously identified or used. Mobile websites enable access to personal data from a user's phone, an experience not offered by desktop websites. For example, mobile web APIs such as tel: and sms: spawn the dialer and the SMS applications respectively on a mobile device. In order to characterize the behavior of mobile API calls, we extracted the number of API calls tel:, sms:, smsto:, mms: and mmsto: from each mobile webpage. We further extracted the target phone numbers from these API calls. We ran the commercially available Pindrop Security Phone Reputation System (PRS) [7] on each phone number. Based on the results of the PRS, we gave the score of 1/0 (known fraud/benign) to each phone number scraped from the mobile API calls, and added the score as a feature in kAYO.

JavaScript enables client-side user interaction, asynchronous communication with servers, and modification of the DOM objects of webpages on the fly. The primary reason in choosing this approach is that a large number of benign webpages include potentially dangerous JavaScript code as shown by Yue et al. [5]. Secondly, external JavaScript can be very large, sometimes of the order of a few megabytes. Our goal is to build a real-time browser extension based on kAYO.

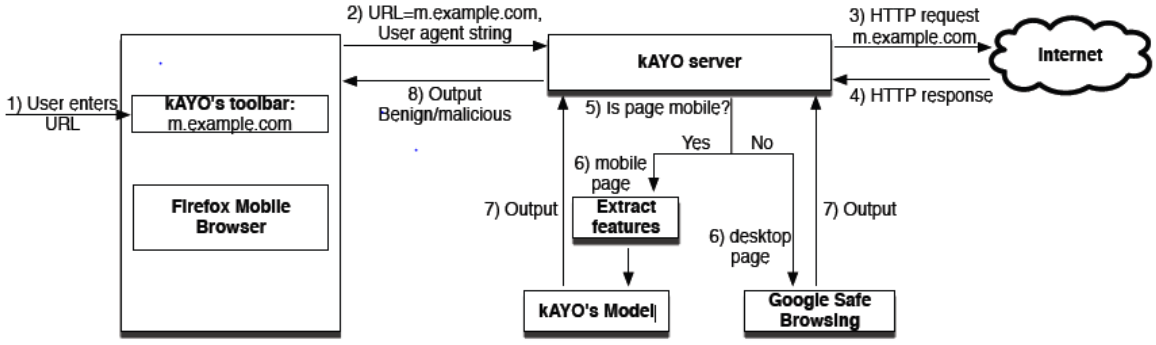
We extract 14 features in total from the HTML code of each webpage. Popular webpages include a number of images, and internal and external HTML links for better user experience. We extract other features such as the percentage of white spaces in the HTML content, the number of cookies from the header, the number of secure and HTTPOnly cookies, and whether the webpage is served over an SSL connection.

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. We extract 12 URL features in total. Words such as login and bank are commonly used in the URL of the login webpage for benign websites that are highly prone to imitation. Only a part of the URL is visible to the user of a mobile phone due to the small screen [13]. Therefore, intuitively, the author of a phishing webpage will include misleading words at the beginning of the URL.

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 crawled the top-level webpage of the 1,000 most popular websites from Alexa.com [11] 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. Our analysis identified nine subdomains (e.g., m.) and seven URL path prefixes (e.g., /mobile) in the URLs of popular websites to represent their mobile specific webpages.

To generate training data for our model, we statically crawled the top-level webpage of the top 1,000,000 most popular websites from Alexa from an Android mobile browser. Gathering data for malicious mobile URLs was challenging since

the mobile web is still evolving and new threats are emerging. We monitored several public blacklists [2], [3], [5] continually for three months and extracted mobile specific URLs from the blacklists.



Architecture of the mobile browser extension based on kAYO. User enters the URL he wants to visit in the extension toolbar and receives a response in real-time from our backend server about the maliciousness of the URL.

IV. IMPLEMENTATION

We build and evaluate our chosen model for accuracy, false positive rate and true positive rate. 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.

We build and evaluate our chosen model for accuracy, false positive rate and true positive rate. 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.

Software Requirements:

- Operating System: Windows 7 or above
- Programming Language: Python
- Database:MySQL
- Database interface: SQLYog
- IDE:pycharm

Hardware Requirements:

- Processor: dual core 2.0GHz
- RAM: 2GB
- HardDisk:20GB

- Input Device: Standard mouse and keyboard
- Output Device: VGA and HR Monitor

V. RESULTS

We developed a browser extension using kAYO for Firefox Mobile[9], which informs users about the maliciousness of the webpages they intend to visit. Our goal was 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. 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.

(a): Chrome desktop browser informing the user of a potentially malicious webpage. The webpage is a known mobile phishing webpage.

(b): kAYO extension running on the Firefox mobile browser detects the webpage as malicious and warns the user.

VI. CONCLUSION

We designed and developed a fast and reliable static analysis technique called kAYO that detects mobile malicious webpages. kAYO makes these detections by measuring 44 mobile relevant features from webpages, out of which 11 are newly identified mobile specific features. kAYO provides 90% accuracy in classification, and detects a number of malicious mobile webpages in the wild that are not detected by existing techniques such as Google Safe Browsing and VirusTotal. Finally, we build a browser extension using kAYO that provides real-time feedback to users. We conclude that kAYO detects new mobile specific threats such as websites hosting known fraud numbers and takes the first step towards identifying new security challenges in the modern mobile web.

REFERENCES

[1] Gnu octave: high-level interpreted language. http://www.gnu.org/ software/octave/.

[2] hphosts, a community managed hosts file. http://hphosts.gt500.org/hosts. txt.

[3] Joewein.de LLC blacklist. http://www.joewein.net/dl/bl/dom-bl-base. txt.

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

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

[6] Phishtank. http://www.phishtank.com/.

[7] Pindrop phone reputation service. http://pindropsecurity.com/ phone-fraud-solutions/phone reputation service prs/.

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

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

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

[11] Alexa, the web information company. http://www.alexa.com/topsites, 2013.

[12] dotmobi. internet made mobile. anywhere, any device. http://dotmobi. com/, 2013.

