Enriching the Detection of Mobile Malicious Webpages

[1]M.Madhuri, [2]Mr.G. Praveen Babu

[1]M.Tech Student, Computer Science, School of Information Technology, Jawaharlal Nehru Technological University Hyderabad, Telangana, India

[2]Associate Professor of CSE, School of Information Technology, Jawaharlal Nehru Technological University Hyderabad, Telangana, India

ABSTRACT: The widespread use of mobile devices to access the web in the present world has increased enormously. The security can be compromised by doing nothing more than visiting a webpage. A malicious webpage is a page that attempts to install malware onto your device. This usually requires some action on your part. Malicious webpages often look like legitimate webpages. Sometimes the malicious webpages will ask you to install software that your computer appears to need. For example, the webpage might ask for permission to install one program, but install a completely different one (one that you definitely do not want on your computer). The layouts and functionality of the webpages in mobiles are different from the desktop ones. The detection techniques which are existing are not efficient in detecting such malicious webpages. That is why it is proposed to introduce a mechanism called kAYO(Knockout in boxing terminology) .

This kAYO mechanism is used to differentiate the malicious webpages from the benign webpages based on the static features (such as number of iframes and known fraudulent phone numbers etc) of the webpages. This proposal will identify and enrich the range of new static features by signifying the need for such mobile specific techniques. It is proposed that the number of webpages that are missed by Google Safe Browsing and VirusTotal will be detected, characterized and reported by kAYO mechanism.

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

1. INTRODUCTION

The viewing of webpages over the mobile phones has largely increased, though the experience is still not highly appreciated due to various reasons like processor power and network bandwidth. The web assaults are the testing issues of the web group. At the point when the client visits the vindictive site the assault is started through different highlights (lexical, space, way, web substance and hyperlink and so on). To avoid the client against getting to the malignant sites, a few mechanized investigation and location strategies have been proposed. Notwithstanding, regardless of huge advances in processor power and transmission capacity, the perusing knowledge on cell phones is extensively unique. These contrasts can to a great extent be ascribed to the sensational diminishment of screen estimate, which impacts the substance, usefulness and design of portable site pages. Substance, usefulness, and design have

routinely been used to perform static investigation to decide noxiousness in the work area space. Highlights such as the recurrence of iframes and the quantity of redirections have generally filled in as solid markers of vindictive plan. Because of the critical changes made to oblige cell phones, such statements may never again be valid. For instance, though such conduct would be hailed as suspicious in the work area setting, numerous famous kind portable pages require various redirections before client's access content. The most usual scheme to make money is to install malignant software on a huge number of hosts. The installed malware programs typically connect to a command and control (C&C) infrastructure. In this way, the attackers fetch all the users' data and in sometimes they corrupt or modify the data. Noxious web content has turned out to be a standout amongst the best systems for digital offenders to disperse vindictive code. Specifically, aggressors much of the time utilize drive-by-download endeavors to trade off countless. To play out a drive-by-download assault, the aggressor first specialties vindictive customer side scripting code (ordinarily written in JavaScript) that objectives a powerlessness in a web program or in one of the program's module. This code is infused into traded off sites or is essentially facilitated on a server under the control of the lawbreakers. At the point when a casualty visits a noxious site page, the pernicious code is executed, and, if the casualty's program is helpless, the program is traded off.

In this paper, kAYO is introduced to identify the malignant web pages in mobiles. This mechanism detects all the malicious web pages which were not detected by using previous techniques. kAYO uses static features of mobile web pages derived from their HTML and JavaScript content, URL and advanced mobile specific capabilities. kAYO is used to identify the cross related attacks which takes place on mobiles and used to improve the efficiency of designing the websites. kAYO provides 90% accuracy in classification and also identifies the mischievous web pages which were not identified by previous techniques. In addition to this, kAYO provides 44 significant features and they were divided into four categories. They are Mobile specific features, java script, URL, Html. These are used to recognize all the malignant web pages in mobile websites.

2. RELATED WORK

Dynamic methodologies utilize honey client frameworks to visit site pages and decide whether they are pernicious or not. In high-connection honey clients, the investigation is performed by utilizing customary programs running in an observed situation what's more, identifying indications of a fruitful drive-by-download assault (e.g., changes in the record framework, the registry, or the arrangement of running procedures). In low-association honey clients, the examination depends on copied programs whose execution amid the visit of a website page is observed to recognize the indication of an assault (e.g., the summon of a defenseless technique in a module). Both high-and low-association frameworks require to completely executing the substance of a website page. This incorporates getting the page itself, every one of the assets that are connected from it and, in particular, translating the related dynamic substance, for example, and JavaScript code. These methodologies generally yield great recognition rates with low false positives, since, by performing dynamic investigation, they have finish "perceivability" into the activities performed by an assault. The drawback is that this examination can be moderately moderate, in view of the time required by the program (either reenacted or

genuine) to recover what's more, execute every one of the substance involving a site page, taking from a couple of moments to a few minutes, contingent upon the many-sided quality of the broke down page.

As malware on the Internet spreads and becomes more sophisticated, anti-malware techniques need to be improved in order to be able to identify new threats in an efficient, and, most important, automatic way. D. Canali, M. Cova et al. developed Prophiler, a system whose goal is to provide a solution that is filter which can minimize the number of web pages that need to be analyzed dynamically to identify malicious web pages. They have deployed our system as a front-end for Wepawet, a well known, publicly-available dynamic analysis tool for web malware. The results show that Prophiler is able to dramatically reduce the load of the Wepawet system with a very small false negative rate.

The domain service (DNS) is a crucial component of the Internet. DNS provides a two-way mapping between domain names and their IP addresses. For instance, DNS is a critical service for the functioning of benign Internet services; it has also started to play the significant role for detection of malicious activities. For example, bots resolve DNS names to locate their command and control servers, and spam mails contain URLs that link to domains that resolve to scam servers. In this paper, L. Bilge, E. Kirda et al. introduced EXPOSURE, a system that employs passive DNS analysis techniques to identify malicious domains. The main achievement is that it is beneficial to monitor the use of the DNS system on a large-scale for signs that indicate that a certain name is used as part of a malicious operation. Compared to related work, our approach is generic, and does only focus on a specific class of threat (e.g., such as Fast-Flux botnets). They believe that EXPOSURE is a useful system that can help security experts and organizations in their fight against cyber-crime. As future work, we plan to release EXPOSURE to the public as a community service.

Mobile devices contains small screens, so that users are not able to see the whole URLs and are very likely to click on the links without enough forethought of possible phishing attacks. Moreover, users download and install applications without realizing that installed applications may not be a copy of legitimate official applications, a problem which overwhelmingly targets financial institutions. This paper describes an overview of different types of mobile phishing attacks. M. Boodaei et al. Also discuss some mitigation approaches and their limitations. They suggest some best practices. The following approach recommends users are careful while downloading apps and the websites which are designed by the attackers. There is a broad scope of further research can be done to develop novel mitigation approaches, especially considering the variation of devices and accessibility of application market.

3. FRAMEWORK

Web page consists of several components such as HTML, java script, the URL, the header and images. Versatile particular website pages additionally get to applications running on a client's gadget utilizing web APIs (e.g., the dialer). We separate auxiliary, lexical and quantitative properties of such segments to produce kAYO list of capabilities. We center on separating versatile important highlights that take insignificant extraction time.

kAYO feature set is divided into four classes. The first one is Mobile specific features and the following are

Java script, HTML and URL features. These are described as shown in Fig 1.

Category	Features	Total # of features
Mobile specific	# of API calls to tel:, sms:, smsto:, mms:, mmsto:, geolocation; # of apk, # of ipa	8
JavaScript	presence of JS, noscript, internal JS, external JS, embedded JS; # of JS, noscript, internal JS, external JS, embedded JS	10
HTML	presence of internal links, external links, images; # of internal links, external links, images # of cookies from header, secure and HTTPOnly cookies, presence of redirections and iframes, # of redirects and iframes, whether webpage served over SSL, % of white spaces in the HTMl content	14
URL	# of misleading words in the URL such as *login* and *bank* , length of URL # of forward slashes and question marks, digits, dots, hyphens and underscores, # of equal signs and ampersand, subdomains, two letter subdomains, semicolons, presence of subdomain, % of digits in hostname	12
	Total:	44

Fig 1: Table summarizes the 8 mobile, 10 JavaScript, 14 HTML and 12 URL features.

3.1. Mobile Specific Features

We gather eight portable particular highlights to catch the propelled abilities of portable site pages. Versatile sites empower access to individual information from a client's telephone, an affair not offered by work area sites. For instance, versatile web APIs, for example, tel: and sms: generate the dialer and the SMS applications separately on a cell phone. Keeping in mind the end goal to portray the conduct of versatile API calls, we separated the quantity of API calls tel, sms, smsto, mms: and mmsto: from each versatile website page. We additionally extricated the objective telephone numbers from these API calls. We ran the monetarily accessible Pindrop Security Phone Reputation System (PRS) on each telephone number. In view of the consequences of the PRS, we gave the score of 1/0 (known extortion/considerate) to each telephone number scratched from the versatile API calls, and included the score as a component in kAYO. We as it were removed telephone numbers with API prefixes that could trigger an application introduced on a client's telephone.

3.2. JavaScript Features

JavaScript empowers customer side client connection, non concurrent correspondence with servers, and alteration of the DOM objects of site pages on the fly. We remove 10 includes that catch the JavaScript important static conduct of a site page, two of which are new. All the highlights are speedier to separate than the highlights in view of JavaScript deobfuscation. JavaScript found on pernicious website pages can be jumbled. We will likely form an ongoing program expansion in view of kAYO. As needs be, we abstained from utilizing highlights that would back off the element extraction process. We contend that kind website page essayists require push to give great client encounter, while the objective for vindictive page writers is to trap clients into performing inadvertent activities with negligible exertion. We hence inspect whether a site page has noscript substance and measure the quantity of noscript. Naturally, a generous website page essayist will have more noscript in the code to guarantee great experience notwithstanding for a security adroit client.

3.3. HTML Features

We separate 14 includes altogether from the HTML code of every website page. Prevalent site pages incorporate a number of pictures, and interior and outside HTML joins for better client encounter. For instance, the best level page of m.cnn.com incorporates connections to different news articles distributed by CNN (inside HTML joins), ads for a nearby eatery (outside HTML connection) and picture identified with the most recent breaking news. In like manner, we first decide if a page has any pictures, inward what's more, outside HTML joins. We at that point extricate the quantity of inward connections, outside connections and pictures from a site page as highlights of kAYO. Noxious pages (particularly those executing drive-by-downloads and clickjacking) incorporate connect to terrible substance in iframes.

3.4. URL Features

Basic and lexical properties of a URL have been used to separate amongst malevolent and kindhearted pages. In any case, utilizing just URL highlights for such separation prompts a high false positive rate. We remove 12 URL includes altogether. Creators of phishing website pages frequently misuse the recognition of clients to a site page by including words in the URL that can delude a client into trusting that the phishing website page is the true blue site page. Words for example, login and bank are ordinarily utilized as a part of the URL of the login site page for kind sites that are profoundly inclined to impersonation. Just a piece of the URL is noticeable to the client of a cell phone because of the little screen. In this manner, naturally, the creator of a phishing website page will incorporate deceiving words toward the start of the URL. We consider the nearness of such words in the URL as another component in kAYO.

4. EXPERIMENTAL RESULTS

Several experiments were conducted to differentiate the malicious webpages from the benign webpages.

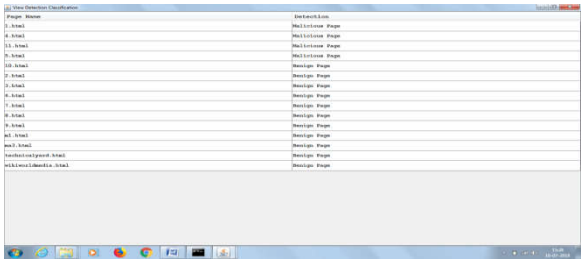

Fig 1: Detection Classification

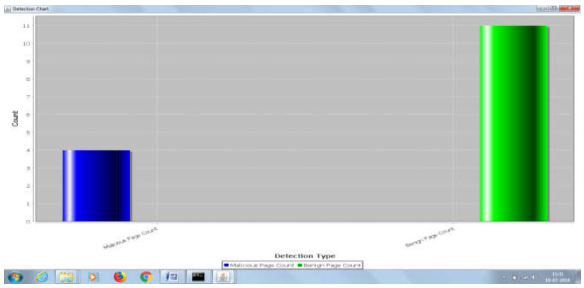

Fig.2: Chart for Detection Classification

The fig.2 shows the detection classification of malicious webpages and benign webpages. The results of the following approach such as kAYO are considering the mobile web pages using mobile features.

5. CONCLUSION

Versatile site pages are fundamentally not quite the same as their work area partners in substance, usefulness and design. Accordingly, existing strategies utilizing static highlights of work area pages to distinguish vindictive conduct don't function admirably for versatile particular pages. We outlined and built up a quick and solid static investigation strategy called kAYO that recognizes versatile malignant site pages. The evaluation is based on the true positive and false positive rates and kAYO gives the lowest false positive rates when compared to the existing static analysis techniques. Thus kAYO gives 90% exactness in characterization, and identifies a number of noxious versatile website pages in the wild that are most certainly not recognized by existing systems, for example, Google Safe Perusing and Virus Total. At long last, we assemble a program augmentation utilizing kAYO that gives constant criticism to clients. We presume that kAYO identifies new versatile particular dangers, for example, sites facilitating known extortion numbers and ventures out

distinguishing new security challenges in the advanced portable 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 LL 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] Virus Total. https://www.virustotal.com/en/.

[10] Google developers: Safe Browsing API. https://developers.google.com/ safe-browsing/, 2012. [11] Alexi, the web information company. http://www.alexa.com/topsites, 2013.

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