Identification of Mobile Malicious Application using KAYO Technique

Doguparthi Madhuri & K.Ramesh babu
[1] PG Scholar Department of CSE, GVR & S College Of Engineering & Technology, Guntur (D.T),Andhra Pradesh

[2]Asst.Professor, Department of CSE, GVR & S College Of Engineering & Technology, Guntur (D.T), Andhra Pradesh

Abstract:-

Versatile explicit website pages contrast fundamentally from their work area partners in substance, format and usefulness. As needs be, existing systems to distinguish vindictive sites are probably not going to work for such website pages. In this paper, we structure and actualize kAYO,a system that recognizes malignant and kind versatile site pages. kAYO makes this assurance dependent on static highlights of a website page going from the quantity of iframes to the nearness of realized fake telephone numbers. In the first place, we tentatively exhibit the requirement for versatile explicit procedures and afterward recognize a scope of new static highlights that profoundly relate with portable malignant website pages. We at that point apply kAYO to a dataset of more than 350,000 known favorable and noxious versatile website pages and exhibit 90% precision in characterization. Besides, we find, portray and report various website pages missed by Google Safe Browsing and Virus Total, however identified by kAYO. At long last, we construct a program augmentation utilizing kAYO to shield clients from malignant portable sites progressively. In doing as such, we give the primary static investigation method to identify pernicious versatile site pages.

Keywords:- Mobile security, web pages, 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 web pages. 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 web pages 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 web pages 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 web pages vary dramatically. We then collect over 350,000 mobile benign and malicious web pages 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 web pages not precisely detected by existing techniques such as Virus Total and Google Safe Browsing. Finally, we discuss the limitations of existing tools to detect mobile malicious web pages and build a browser extension based on kAYO that provides real time feedback to mobile browser users. We make the following contributions:

Experimentally demonstrate the differences in the "security features" of desktop and mobile web pages:

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 web pages. 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 web pages.

Design and implement a classifier for malicious and benign mobile web pages:

We collect over 350,000 benign and malicious mobile web pages. We then identify new static features from these web pages that distinguish between mobile benign and 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 web pages 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 web pages 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 web pages 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 web pages. 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.LITERATURE SURVEY:-

1) TVDc: Managing Security in the Trusted Virtual Datacenter

AUTHORS: S. Berger et al

Virtualization technology is becoming increasingly common in datacenters, since it allows for collocation of multiple workloads, consisting of operating systems, middleware and applications, in different virtual machines (VMs) on shared physical hardware platforms. However, when coupled with the ease of VM migration, this trend increases the potential surface for security attacks. Further, the simplified management of VMs, including creation, cloning and migration, makes it imperative to monitor and guarantee the integrity of software components running within VMs.This paper presents the IBM Trusted Virtual Datacenter (TVDc) technology developed to address the need for strong isolation and integrity guarantees, thus significantly enhancing security and systems management capabilities, in virtualized environments. It signifies the first effort to incorporate trusted computing technologies directly into virtualization and systems management software. We present and discuss various components that constitute TVDc: the Trusted Platform Module (TPM), the virtual TPM, the IBM hypervisor security architecture (sHype) and the associated systems management software.

2. A Framework for Building Privacy-Conscious Composite Web Services

AUTHORS: W. Xu, V.N. Venkatakrishnan, R. Sekar, and I.V. Ramakrishnan,

The rapid growth of web applications has prompted increasing interest in the area of composite web services that involve several service providers. The potential for such composite web services can be realized only if consumer privacy concerns are satisfactorily addressed. In this paper, we propose a framework that addresses consumer privacy concerns in the context of highly customizable composite web services. Our approach involves service producers exchanging their terms-of-use with consumers in the form of "models". Our framework provides automated techniques for checking these models at the consumer site for compliance of consumer privacy policies. In the event of a policy violation, our framework supports automatic generation of "obligations" that the consumer generates for the composite service. These obligations are automatically enforced through a dynamic program analysis approach on the web service composition code. We illustrate our approach with the implementation of two example services.

3. Towards Standardized Web Services Privacy Technologies

AUTHORS: P.C.K. Hung, E. Ferrari, and B. Carminati

A Web service is defined as an autonomous unit of application logic that provides either some business functionality or information to other applications through an Internet connection. Web services are based on a set of XML standards such as universal description, discovery and integration (UDDI), Web services description language (WSDL), and simple object access protocol (SOAP). Recently there are increasing demands and discussions about Web services privacy technologies in the industry and research community. In general, privacy policies describe an organization's data practices what information they collect from individuals (e.g., consumers) and what (e.g., purposes) they do with it. To enable privacy protection for Web service consumers across multiple domains and services, the World Wide Web Consortium (W3C) published a document called "Web services architecture (WSA) requirements" that defines some specific privacy requirements for Web services as a future research topic. At this moment, there is still no standardized Web services privacy technology. This paper briefly overviews the research issues of Web services privacy technologies.

4. Managing and Securing Web Services with VPNs

AUTHORS: L. Alchaal, V. Roca, and M. Habert

Web Services constitute a set of technologies that many believe will change the Web communication landscape within the next few years. They offer standardized and easy communications for distributed systems over the Internet. However their dynamic and distributed nature requires a well-managed system, and pending security issues prevent their widespread adoption. Meanwhile there is a big rage toward the use of Virtual Private Networks (VPNs) to secure communications in a cost-effective environment like the Internet. In this paper we explain how to merge these two technologies in a new powerful hybrid model that: (1) enables an easy management of Web services, (2) provides Web services security thanks to the use of dynamic and programmable VPNs, and (3) remains simple and fully integrated.

5. Software Integrity Protection Using Timed Executable Agents

AUTHORS: J. Garay and L. Huelsbergen

We present a software scheme for protecting the integrity of computing platforms using Timed Executable Agent Systems (TEAS). A trusted challenger issues an authenticated challenge to a perhaps corrupt responder. New is that the issued challenge is an executable program that can potentially compute any function on the responder. The responder must compute not only the correct value implied by the agent, but also must complete this computation within time bounds prescribed by the challenger. Software-based attestation schemes have been proposed before new capabilities introduced in TEAS provide means to mitigate the existing shortcomings of such proposed techniques. TEAS are general and can be adapted to many applications for which system integrity is to be tested.

III.RELATED WORK:-

Content-based and in-depth inspection techniques to detect malicious websites: Dynamic approaches using virtual machines [45], [51] and honey client systems [32],

[42], [46] 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 [28], [30], [35], [39]. However, URL-based techniques usually suffer from high false positive rates. Using HTML and JavaScript features extracted from a webpage in addition to URL classification helps address this drawback and provides better results [20], [41],[55], [59]. Static approaches avoid performance penalty of dynamic approaches. Additionally, using fast and reliable static approaches to detect benign web pages can avoid expensive in-depth analysis of all web pages.

Differences between mobile and desktop websites: 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 [13], [14]. Although differences in mobile and desktop websites have been observed before [19], it is

unclear how these differences impact security. Furthermore, the threats on mobile and desktop websites are somewhat different [26]. Static analysis techniques using

features of desktop web pages have been primarily studied for drive-by-downloads on desktop websites [20],[51], whereas, the biggest threat on the mobile web at present is believed to be phishing [18]. Efforts in mitigating phishing attacks on desktop websites include isolating browser applications of different trust level [29],

email filtering [28], using content-based features [55], [59] and blacklists [38]. The best-known non-proprietary content-based approach to detect phishing web pages is Cantina [59]. Cantina suffers from performance problems due to the time lag involved in querying the Google search engine. Moreover, Cantina does not work well on web pages 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.

Mobile application security: Significant work has been done in the past few years on the security of mobile applications. Static feature extraction, especially with respect to permissions, has been one of the most important early areas of research [21], [23], [25], [27]. Such techniques have led to dramatically more rapid detection of malicious applications across a range of marketplaces.

IV. CONCLUSION

Mobile web pages are significantly different than their desktop counterparts in content, functionality and layout. Therefore, existing techniques using static features of desktop web pages to detect malicious behavior do not work well for mobile specific pages. We designed and developed a fast and reliable static analysis technique called kAYO that detects mobile malicious web pages. kAYO makes these detections by measuring 44 mobile relevant features from web pages, out of which 11 are newly identified mobile specific features. kAYO provides 90% accuracy in classification, and detects a number of malicious mobile web pages in the wild that are not detected by existing techniques such as Google Safe Browsing and Virus Total. 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.

V.BIBLIOGRAPHY

References Made From:-

[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.
[13] C. Amrutkar, K. Singh, A. Verma, and P. Traynor. VulnerableMe: Measuring systemic weaknesses in mobile browser security. In Proceedings of the International Conference on Information Systems Security (ICISS), 2012.
[14] C. Amrutkar, P. Traynor, and P. C. van Oorschot. Measuring SSL indicators on mobile browsers: Extended life, or end of the road? In Proceedings of the Information Security Conference (ISC), 2012.
[15] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for DNS. In Proceedings of the 19th USENIX Conference on Security (SECURITY), 2010.
[16] V. A. Balasubramaniyan, A. Poonawalla, M. Ahamad, M. T. Hunter, and P. Traynor. Pindr0p: using single-ended audio features to determine call provenance. In Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS), 2010.
[17] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE : Finding malicious domains using passive DNS analysis. In Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS), 2011.
[18] 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.

[19] M. Butkiewicz, Z. Wu, S. Li, P. Murali, V. Hristidis, H. V. Madhyastha, and V. Sekar. Enabling the transition to the mobile web with websieve. In Proceedings of the 14thWorkshop on Mobile Computing Systems and Applications (HotMobile), 2013.

[20] D. Canali, M. Cova, G. Vigna, and C. Kruegel. Prophiler: a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th International Conference on World Wide Web (WWW), 2011.

[21] S. Chakradeo, B. Reaves, P. Traynor, andW. Enck. MAST: Triage for Marketscale Mobile Malware Analysis. In Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), 2013.

[22] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2006.

[23] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri. A study of Android application security. In Proceedings of the 20th USENIX Security Symposium, 2011.

[24] B. Feinstein and D. Peck. Caffeine monkey: Automated collection, detection and analysis of malicious javascript. In Proceedings of the Black Hat Security Conference, 2007.

[25] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security, 2011.

[26] A. P. Felt and D. Wagner. Phishing on mobile devices. In Web 2.0 Security and Privacy (W2SP), 2011.

[27] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin. Permission re-delegation: attacks and defenses. In Proceedings of the 20th USENIX conference on Security, 2011.

[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.

[29] S. Gajek, A.-R. Sadeghi, C. St ¨ uble, and M. Winandy. Compartmented security for browsers or how to thwart a phisher with trusted computing. In Second International Conference on Availability, Reliability and Security (ARES), 2007.

[30] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In Proceedings of the ACM workshop on recurring malcode, 2007.

[31] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and detecting fast-flux service networks. In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2008.

[32] A. Ikinci, T. Holz, and F. Freiling. Monkey-spider: Detecting malicious websites with low-interaction honeyclients. In Proceedings of Sicherheit, Schutz und Zuverlassigkeit, 2008.

[33] L. Invernizzi, S. Benvenuti, M. Cova, P. M. Comparetti, C. Kruegel, and G. Vigna. Evilseed: A guided approach to finding malicious web pages. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, 2012.

[34] P. Kolari, T. Finin, and A. Joshi. Svms for the blogosphere: Blog identification and splog detection. In Proceedings of AAAI Spring Symposium on

Computational Approaches to Analysing Weblogs, 2006.

[35] A. Le, A. Markopoulou, and M. Faloutsos. Phishdef: Url names say it all. In Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2011.

[36] C. Lever, M. Antonakakis, B. Reaves, P. Traynor, and W. Lee. The core of the matter: Analyzing malicious traffic in cellular carriers. In Proceedings of Network and Distributed System Security Symposium (NDSS), 2013.

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

[38] C. Ludl, S. Mcallister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. In Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2007.

[39] 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.

[40] D. K. McGrath and M. Gupta. Behind phishing: an examination of phisher modi operandi. In Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008.

[41] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In Proceedings of International Conference on Security and Privacy in Communication Netowrks (SecureComm), 2008.

[42] Y. min Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In Proceedings of the Networking and Distributed Systems Security (NDSS), 2006.

[43] A. Mohaisen and O. Alrawi. AV-Meter: An Evaluation of Antivirus Scans and Labels. In Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2014.

[44] T. Moore, R. Clayton, and H. Stern. Temporal correlations between spam and phishing websites. In Proceedings of the 2nd USENIX conference on Largescale exploits and emergent threats: botnets, spyware, worms, and more (LEET), 2009.

[45] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy. A crawler-based study of spyware on the web. In Proceedings of Network and Distributed System Security Symposium (NDSS), 2006.

[46] J. Nazario. Phoneyc: a virtual client honeypot. In Proceedings of the 2nd USENIX conference on Large-scale Exploits and Emergent Threats: botnets, spyware, worms, and more (LEET), 2009.

[47] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In Proceedings of 3rd International Conference on Malicious and Unwanted Software (MALWARE), 2008.

[48] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. Fluxor: Detecting and monitoring fast-flux service networks. In Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2008.

[49] R. Perdisci, I. Corona, D. Dagon, and W. Lee. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In Proceedings of Annual Computer Security Applications Conference (ACSAC), 2009.

[50] G. Podjarny. Mobile web performance optimization. http://www.slideshare.net/blazeio/ mobile-web-performance-optimization-tips-and-tricks.

[51] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iframes point to us. In Proceedings of the 17th USENIX conference on Security (SECURITY), 2008.

[52] L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. The American Statistician, 1988.

[53] C. Seifert, I. Welch, and P. Komisarczuk. Identification of malicious web pages with static heuristics. In Telecommunication Networks and Applications Conference, 2008.

[54] F. Weimer. Passive DNS replication. 2005.

[55] C. Whittaker, B. Ryner, and M. Nazif. Large-scale automatic classification of phishing pages. In Proceedings of the Networking and Distributed Systems Security (NDSS), 2010.

[56] G. Xiang, J. Hong, C. P. Rose, and L. Cranor. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security (TISSEC), 14(2), Sept. 2011.

[57] C. Yue and H. Wang. Characterizing insecure javascript practices on the web. In Proceedings of the 18th international conference on World Wide Web (WWW), 2009.

[58] B. Zdrnja, N. Brownlee, and D. Wessels. Passive monitoring of DNS anomalies. In Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2007.

[59] 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.