International Journal of Research
Available at https://pen2print.org/index.php/ijr/
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 20
September 2018

Reliable Static Analysis Technique to Detect Mobile Malicious Webpages

T V Divya & S G Nawaz Hod

[1] *M.Tech Student, Dept of CSE, SKD Engineering College, Affiliated to JNTUA, AP, India*

[2] *Associate Professor & HOD, Dept of CSE, SKD Engineering College, Affiliated to JNTUA, AP, India*

Abstract:
Web browsers to mobile platforms may lead to new vulnerabilities whose solutions require careful balancing between usability and security and might not always be equivalent to those in desktop browsers. In existing system DNS based mechanisms do not provide deeper understanding of the specific activity. In this paper, kAYO a fast and reliable static analysis technique to detect malicious mobile web-pages is proposed which uses static features of mobile web pages derived from their HTML and JavaScript content, URL and advanced mobile specific capabilities. Finally, a browser extension is build using kAYO to protect users from malicious mobile websites in real-time.

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

I. INTRODUCTION

Mobile browsers are increasingly being relied upon to perform security sensitive operations. Like their desktop counterparts, these applications can enable SSL/TLS to provide strong security guarantees for communications over the web. However, the drastic reduction in screen size and the accompanying reorganization of screen real estate significantly changes the use and consistency of the security indicators and certificate information that alert users of site identity and the presence of strong cryptographic algorithms. Internet attacks that use Web servers to exploit browser vulnerabilities to install malware programs are on the rise. Several recent reports suggested that some companies may actually be building a business model around such attacks. Expensive, manual analyses for individually discovered malicious Web sites have recently emerged. 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.

II. LITERATURE SURVEY

1) C. Amrutkar, P. Traynor, and P. C. van Oorschot. Measuring SSL indicators on mobile browsers: Extended life, or end of the road

Mobile browsers are increasingly being relied upon to perform security sensitive operations. Like their desktop counterparts, these applications can enable SSL/TLS to provide strong security guarantees for communications over the web. However, the drastic reduction in screen size and the accompanying reorganization of screen real estate significantly changes the use and consistency of the security indicators and certificate information that alert users of site identity and the presence of strong cryptographic algorithms. In this paper, we perform the first measurement of the state of critical security indicators in mobile browsers. We evaluate ten mobile and two tablet browsers, representing over 90% of the market share, using the recommended guidelines for web user interface to convey security set forth by the World Wide Web Consortium (W3C). While desktop browsers follow the

majority of guidelines, our analysis shows that mobile browsers fall significantly short. We also observe notable inconsistencies across mobile browsers when such mechanisms actually are implemented. Finally, we use this evidence to argue that the combination of reduced screen space and an independent selection of security indicators not only make it difficult for experts to determine the security standing of mobile browsers, but actually make mobile browsing more dangerous for average users as they provide a false sense of security.

2) W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri. A study of Android application security.
The fluidity of application markets complicate smart phone security. Although recent efforts have shed light on particular security issues, there remains little insight into broader security characteristics of smart phone applications. This paper seeks to better understand smart phone application security by studying 1,100 popular free Android applications. We introduce the ded decompiler, which recovers Android application source code directly from its installation image. We design and execute a horizontal study of smart phone applications based on static analysis of 21 million lines of recovered code. Our analysis uncovered pervasive use/misuse of personal/ phone identifiers, and deep penetration of advertising and analytics networks. However, we did not find evidence of malware or exploitable vulnerabilities in the studied applications. We conclude by considering the implications of these preliminary findings and offer directions for future analysis.

3) 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.
Internet attacks that use Web servers to exploit browser vulnerabilities to install malware programs are on the rise. Several recent reports suggested that some companies may actually be building a business model around such attacks. Expensive, manual analyses for individually discovered malicious Web sites have recently emerged. In this paper, we introduce the concept of Automated Web Patrol, which aims at significantly reducing the cost for monitoring malicious Web sites to protect Internet users. We describe the design and implementation of the Strider HoneyMonkey Exploit Detection System, which consists of a network of monkey programs running on virtual machines with different patch levels and constantly patrolling the Web to hunt for Web sites that exploit browser vulnerabilities. Within the first month of utilizing this new system, we identified 752 unique URLs that are operated by 287 Web sites and that can successfully exploit unpatched WinXP machines. The system automatically constructs topology graphs that capture the connections between the exploit sites based on traffic redirection, which leads to the identification of several major players who are responsible for a large number of exploit pages. For more information on the Strider Honeymonkey research project, please visit
http://research.microsoft.com/honeymonkey.

III. Existing method:

Existing tools such as Google Safe Browsing are not enabled on the mobile versions of browsers, thereby precluding mobile users. 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.

DISADVANTAGES OF EXISTING SYSTEM:

- 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 web pages written in languages other than English.

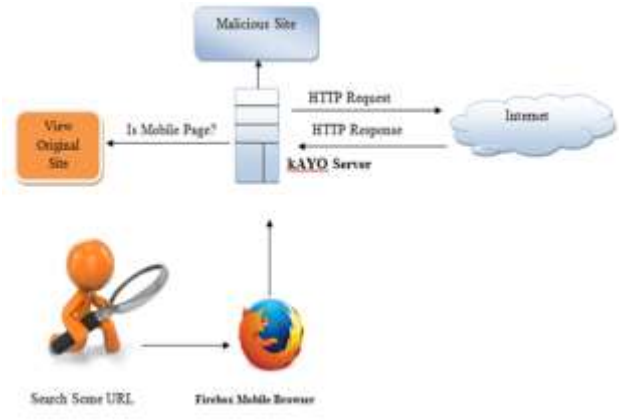
IV. Proposed method:

- 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 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 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.

Advantages of proposed method:

- The results of our experiments demonstrate the need for mobile specific techniques for detecting malicious web pages.
- To the best of our knowledge kAYO is the first technique that detects mobile specific malicious web pages by static analysis.
- 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.

System Architecture



V. MODULES

❖ Admin

In this module, admin server has to login with valid username and password. After login successful he can do some operations such as -- View all users and authorize and Add Topics with Topic name,URL,Desc(enc),Uses,URL Author, Launched year, attach Topic image, List all topics urls with ranking order by desc and rating order by desc, Set Limit to access malicious WebPages and view, List all Malicious WebPages(if admin name is null,publicher name is Hacker) with attacker names with date and time and IP Address, List all Malicious WebPages accessed user details with date and time and IP Address, Block Malicious WebPages accessed user if they cross access limit and view the same, View all recommended WebPages by other users ,View all Web pages viewed users details with date and time and IP Address, View Topic ranks in chart, view NO.of time accessed specified Malicious web page by particular user in the chart, View No.Of blocked and Un blocked users in the chart

❖ User

In this module, User should register before searching the Website contents. After registration successful the user can login by using valid user name and password. After Login successful the user will do some operations --- View profile, Search WebPages by content keyword - Display only topic name order by description and WebPages and then click on topic name to view all details (increase rank), and recommend to other users, click on web url to display webpage, View all other user recommended Web pages, View Top

k web pages ulrs and view the details(increase rank)

VI. CONCLUSION

In this paper, kAYO a fast and reliable static analysis technique to detect malicious mobile web-pages is proposed which uses static features of mobile web pages derived from their HTML and JavaScript content, URL and advanced mobile specific capabilities. Finally, a browser extension is build using kAYO to protect users from malicious mobile websites in real-time. 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. The results of our experiments demonstrate the need for mobile specific techniques for detecting malicious web pages.

REFERENCES

[1] S. Chakradeo, B. Reaves, P. Traynor, and W. 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.

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

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

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

[5] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.