Novel Framework for Malicious Ads Hidden Attacks Detection in Mobile-Web App Interface

[1]***Prof. Srinivasa Murthy H***, *Associate Professor, SJCIT, Chikkaballapur.*
[2]***Nagendrababu NC***, *UG Student, SJCIT, Chikkaballapur.*
[3]***Balaji AV***, *UG Student, SJCIT, Chikkaballapur.*
[4]***Mayank Anand***, *UG Student, SJCIT, Chikkaballapur.*
[5]***Raju V***, *UG Student, SJCIT, Chikkaballapur*

ABSTRACT: *Mobile users are increasingly becoming targets of malware infections and scams. In order to curb such attacks it is important to know how these attacks originate. We take a previously unexplored step in this direction. Numerous in-app advertisements work at this interface: when the user taps on the advertisement, she is led to a web page which may further redirect until the user reaches the final destination. Our static analysis system identified 242 different ad libraries and dynamic analysis system was deployed for a two-month period and analyzed over 600,000 applications while triggering a total of about 1.5 million links in applications to the Web. We gain a general understanding of attacks through the app-web interface and make several interesting findings including a rogue antivirus scam, free iPad scams, and advertisements propagating SMS Trojans*

INTRODUCTION

Android is the predominant mobile operating system with about 80% worldwide market share [1]. At the same time, Android also tops among mobile operating system in terms of malware infections [2]. Part of the reason for this is the open nature of the Android ecosystem, which permits users to install applications for unverified sources. This means that users can install applications from third-party app stores that go through no manual review or integrity violation. This leads to easy propagation of malware. In addition, industry researchers are reporting [3] that some scams which traditionally target desktop users, such as ransomware and phishing, are also gaining ground on mobile devices. In order to curb Android malware and scams, it is important to understand how attackers reach users. While a significant amount of research effort has been spent analyzing the malicious applications themselves, an important, yet unexplored vector of malware propagation is benign, legitimate applications that lead users to websites hosting malicious applications. We call this the app-web interface. In some cases this occurs through web links embedded directly in applications, but in other cases the malicious links are visited via the landing pages of advertisements coming from ad networks. A solution directed towards analyzing and understanding this malware propagation vector will have three components: triggering (or exploring) the application UI and following any reachable web links; detection of malicious content; and collecting provenance information, i.e., how malicious content was reached. There has been some related research in the context of Web to study so-called malvertising or malicious advertising [4], [5]. The context of the problem here is broader and the problem itself requires different solutions to triggering and detection to deal with aspects specific to mobile platforms (such as complicated UI and trojans being the primary kinds of malware).

In order to better analyze and understand attacks through app-web interfaces, we have developed an analysis framework to explore web links reachable from an application and detect any malicious activity. We dynamically analyze applications by exercising their UI automatically and visiting and recording any web links that are triggered. We have used this framework to analyze 600,000 applications, gathering about 1.5 million URLs, which we then further analyzed using established URL blacklists and anti-virus systems to identify malicious websites and applications that are downloadable from such websites. We need to mention that we could not trigger ads or links in about 5/6th of the applications. Note that many applications do not have any ad libraries (we can statically check for this) but still have to be run as there may be other kinds of links present. To give an example, for a run of 200K applications, we obtained 400K chains with 770K URLs. However, there are only 30K unique applications and 180K unique URLs. The other applications either do not have any ads or links or, in some cases, we may not have been able to trigger those ads or links. Our

methodology enables us to explore the Web that is reachable from within mobile applications, something that is not possible for traditional search engines and website blacklist systems such as Google Safe browsing. We are not aware of any previous work that enables this.

LITERATURE REVIEW

Paper 1: A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, "The dark alleys of madison avenue: Understanding malicious advertisements," in Proceedings of the 2014 Conference on Internet Measurement Conference.

Online advertising drives the economy of the World Wide Web. Modern websites of any size and popularity include advertisements to monetize visits from their users. To this end, they assign an area of their web page to an advertising company (so called ad exchange) that will use it to display promotional content.

This paper shed light on a little studied, yet important, aspect of advertisement networks, and can help both advertisement networks and website owners in securing their web pages and in keeping their visitors safe.

Paper 2: Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, "Knowing your enemy: Understanding and detecting malicious web advertising," in Proceedings of the 2012 ACM conference on Computer and Communications Security. ACM, 2012, pp. 674–686.
Display ads (and not just malvertisements), are delivered through a network involving publishers, advertisers, and audiences. Publishers display ads on their pages on behalf of advertisers by embedding ad tags in their web pages. These generate requests to an ad network for ad content, which may be dynamically customized according to the user.

PAPER 3: V. Rastogi, R. Shao, Y. Chen, X. Pan, S. Zou, and R. Riley, "Are these ads safe: Detecting hidden attacks through the mobile app-web interfaces

Mobile users are increasingly becoming targets of malware infections and scams. Some platforms, such as Android, are more open than others and are therefore easier to exploit than other platforms. In order to curb such attacks it is important to know how these attacks originate.

They have realized this methodology through various techniques and contributions and have developed a robust, integrated system capable of running continuously without human intervention.

PROBLEM DEFINITION

As part of our triggering app-web interfaces, we developed a novel technique to interact with UI widgets whose internals do not appear in the GUI hierarchy. We develop a computer graphics-based algorithm to find clickable elements inside such widgets.

In order to assist with determining the provenance of the identified malicious links, we conducted a systematic study to associate ad networks with ad library packages in existing applications. We apply the MinHash [6] and LSH [7] techniques to greatly improve the efficiency.

OBJECTIVES OF THE PAPER

- To interact with the application to launch web links, which may be statically embedded in the application code or may be dynamically generated?
- To discriminate between malicious and benign activities that may occur as a result of triggering.
- To identify the origin of a detected malicious activity, and attributing events to specific entities or parties.
- To design algorithm which enable to perform the root cause analysis of malware.

PROPOSED METHODOLOGY

We have developed a framework for analyzing the app-web interfaces in Android applications. We identify three features for a successful methodology: triggering of the app-web interfaces, detection of ma-licious content, and provenance to identify the respon-sible parties. We incorporate appropriate solutions for the above features and have implemented a robust system to automatically analyze app-web interfaces. The system is capable of continuous operation with little human intervention.

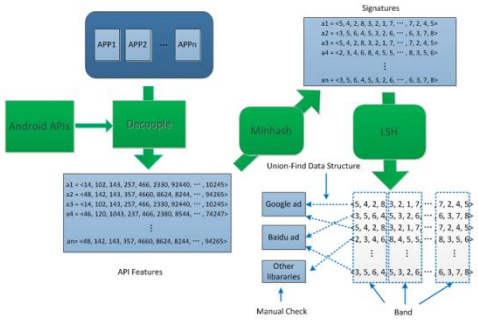


Figure 1 : Overview of the structure of ad library identification

In order to assist with determining the provenance of the identified malicious links, we conducted a systematic study to associate ad networks with ad library packages in existing applications. We apply the MinHash [6] and LSH [7] techniques to greatly improve the efficiency. The system is also incremental, allowing new apps to be analyzed on demand. Our study reveals 242 ad networks and their associated ad library packages. To the best of our knowledge, this is the largest number of ad libraries identified. We also analyze the popularity of the applications to help us understand the distribution of ad libraries in the markets.

Once we have identified components in applications, we can make clusters of similar components over our entire application set. Ad libraries tend to be used by many applications at once and thus bigger clusters are more likely to correspond to ad libraries that smaller clusters. Our clustering should be robust against minor differences in code of components as well as renaming of classes and packages.

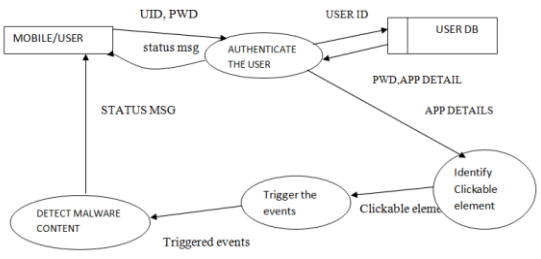


Figure 2: Data Flow diagram

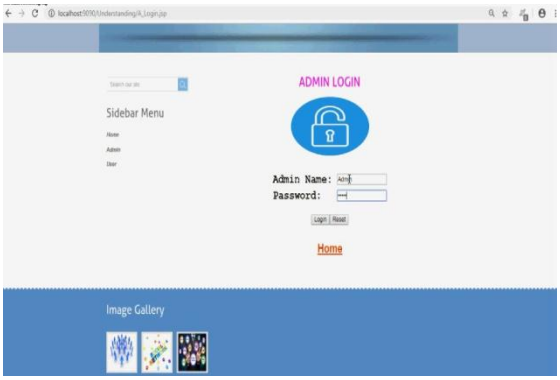
IMPLEMENTATION

Triggering: In order to trigger web links from within the application, we run the applications in a custom dynamic analysis environment. To enable scalability and continuous operation, running applications on real devices is not a feasible option. We deployed our system using multiple AVDs (20 in our test) in parallel for large-scale testing. If we use multiple real phones to run apps, it will increase the costs.

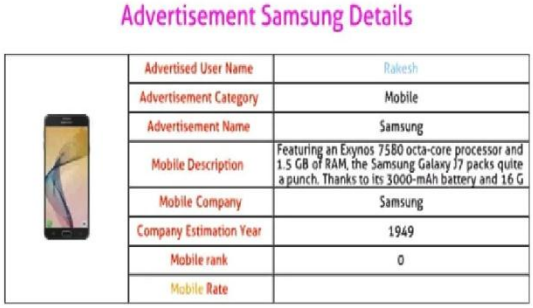
Detection: This includes the various processes to discriminate between malicious and benign activities that may occur as a result of triggering. As the links are triggered, they may be saved for further analysis and detection of malicious activity such as spreading malware or scam. We would like to capture the links, their redirection chains, and their landing pages.

Provenance: This is about understanding the cause or origin of a detected malicious activity, and attributing events to specific entities or parties. Once a malicious activity is detected, this component provides the information required in order to hold the responsible parties accountable. Once a malicious event is detected, it is necessary to make the right attributions to the parties involved so that these parties can be held responsible and proper action may be taken.

SCREENSHOTS

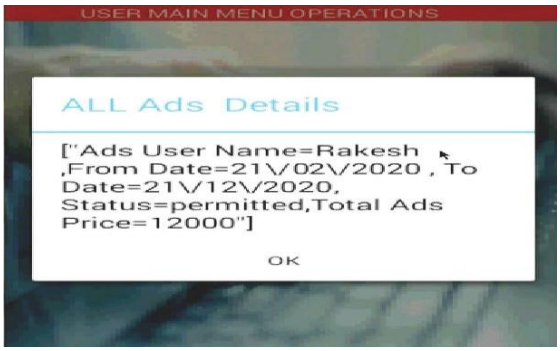


Admin login

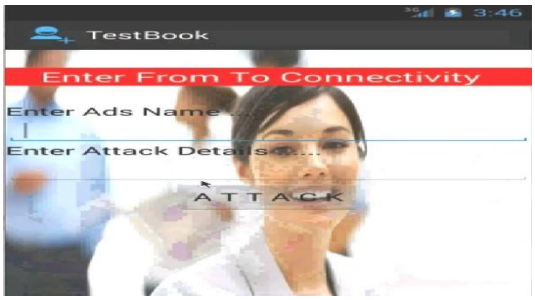


Advertisement details in Webapp

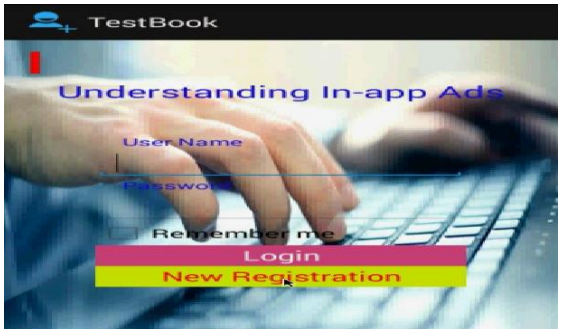
Authorized Users



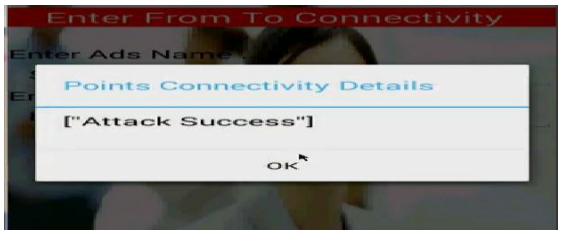
All Ads details



Android Screen for Attack initialization



New registration for android user



Succesfull Attack

Viewing published Ads

REFERENCES

[1] "Smartphone os market share, q1 2015," http://www.idc.com/ prodserv/smartphone-os-market-share.jsp.
[2] "Malware infected as many android devices as windows laptops in 2014," http://bgr.com/2015/02/17/ android-vs-windows-malware-infection/.
[3] "Android phones hit by 'ransomware'," http://bits.blogs.nytimes. com/2014/08/22/android-phones-hit-by-ransomware/? r=0.
[4] A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, "The dark alleys of madison avenue: Understanding malicious advertisements," in Proceedings of the 2014 Conference on Internet Measurement Conference. ACM, 2014, pp. 373–380.
[5] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, "Knowing your enemy: understanding and detecting malicious web advertising," in Proceedings of the 2012 ACM conference on Computer and Communications Security. ACM, 2012, pp. 674–686.
[6] A. Z. Broder, "On the resemblance and containment of documents," in Compression and Complexity of Sequences 1997. Proceedings, Jun 1997, pp. 21–29.
[7] J. Buhler, "Efficient large-scale sequence comparison by localitysensitive hashing," vol. 17, no. 5, pp. 419–428, 2001.
[8] V. Rastogi, R. Shao, Y. Chen, X. Pan, S. Zou, and R. Riley, "Are theseads safe: Detecting hidden attacks through the mobile app-web interfaces," 2016.
[9] W. Zhou, Y. Zhou, M. Grace, X. Jiang, and S. Zou, "Fast, scalable detection of piggybacked mobile applications," in Proceedings of the third ACM conference on Data and application security and privacy. ACM, 2013, pp. 185–196.
[10] V. Rastogi, Y. Chen, and W. Enck, "AppsPlayground: Automatic Security Analysis of Smartphone Applications," in Proceedings of ACM CODASPY, 2013.
[11] "Selendroid: Selenium for android," http://selendroid.io/. [12] V. Rastogi, Y. Chen, and W. Enck, "Appsplayground: automatic security analysis of smartphone applications," in Proceedings of the third

ACM conference on Data and application security and privacy. ACM, 2013, pp. 209–220.
[13] "Celery: Distributed task queue," http://www.celeryproject.org/.
[14] N. Viennot, E. Garcia, and J. Nieh, "A measurement study of google play," in The 2014 ACM international conference on Measurement and modeling of computer systems. ACM, 2014, pp. 221–233.
[15] Symantec, "Airpush begins obfuscating ad modules," November 2012, http://www.symantec.com/connect/blogs/ airpush-begins-obfuscating-ad-modules.
[16] http://forums.makingmoneywithandroid.com/ advertising-networks/1868-tapcontext-shit-breaking-policy-making-loosing-active-users. html#post12949.
[17] http://www.androidauthority.com/armor-for-android-342192/.
[18] "Reputation of amarktflow.com," https://www.mywot.com/en/ scorecard/amarktflow.com.
[19] "Free iPad mini scam spreads via facebook rogue application," https://nakedsecurity.sophos.com/2012/10/ 31/free-ipad-mini-facebook/.
[20] "Apple iPad scam," http: //blog.spamfighter.com /software/ apple-ipad-scam.html.