IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 06, Issue 1, January 2016, ||V-5|| PP 58-61

Taming Information Stealing Smartphone Application

[1]Mrs. M. Behima, [2]Mr. S. Niresh Kumar, [3]Dr. Robert Masillamani
[1]*Annai Veilankanni's College of Engineering*
[2]*Dhanalakshmi Srinivasan College of Engineering and Technology*
[3]*Dhanalakshmi Srinivasan College of Engineering and Technology*

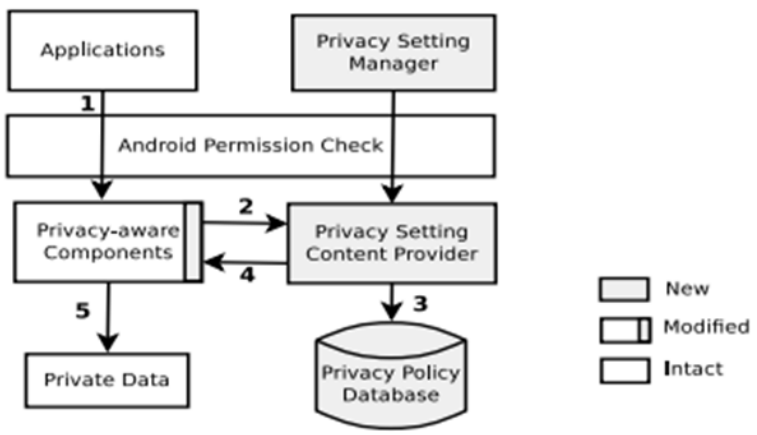
ABSTRACT: *Smart phones are becoming ubiquitous and mobile users are increasingly counting on them to store and handle personal information. The limitations here is the disturbing fact that users' personal information is put at risk by (rogue) smart phone applications. Existing solutions exhibit limitations in their capabilities in taming these privacy-violating smart phone applications. In this paper, we argue for the need of a new privacy mode in smart phones. The privacy mode can empower users to flexibly control during a fine-grained manner what sorts of personal information are going to be accessible to an application. Also, the access to data can be dynamically adjusted at runtime in a fine-grained manner to better suit a user's needs in various scenarios (e.g., in a different time or location). We have proposed a system called TISSA that implements such a secured mode on Android. The evaluation with quite a dozen of information-leaking Android applications demonstrates its effectiveness and practicality. Furthermore, our evaluation shows that TISSA introduces negligible performance overhead.*

KEYWORDS: *TISSA, Android*

I. INTRODUCTION

Mobile phones are increasingly ubiquitous. According to a recent Gartner report [2], in the third quarter of 2010, worldwide mobile phone sales to end users totaled 417 million units, a 35 percent increase from the third quarter of 2009. Among the variety of phones, smartphones in particular received incredible adoption. This trend is further propelled with the wide availability of featurerich applications that can be downloaded and run on smartphones. For example, Google provides Android Market [1] that contains a large collection of Android applications (or apps for short). It is important to note that these app stores or marketplaces contain not only vendor-provided programs, but also third-party apps. For example, Android Market had an increase from about 15,000 thirdparty apps in November 2009 to about 150, 000 in November 2010. Given the increased sophistication, features, and convenience of these smartphones, users are increasingly relying on them to store and process personal

As a demonstration, we have implemented a system called TISSA that implements such a privacy mode in Android. Our development experience indicates that though the privacy mode support requires modifying the Android framework, the modification however is minor with changes in less than 1K lines of code (LOC). We also have evaluated TISSA with more than a dozen of Android apps that are known to leak a variety of private information. Our results show that TISSA can effectively mediate their accesses and protect private information from being divulged. Also, the privacy setting for each app is re-adjustable at runtime without affecting its functionality.



II. SYSTEM ANALYSIS

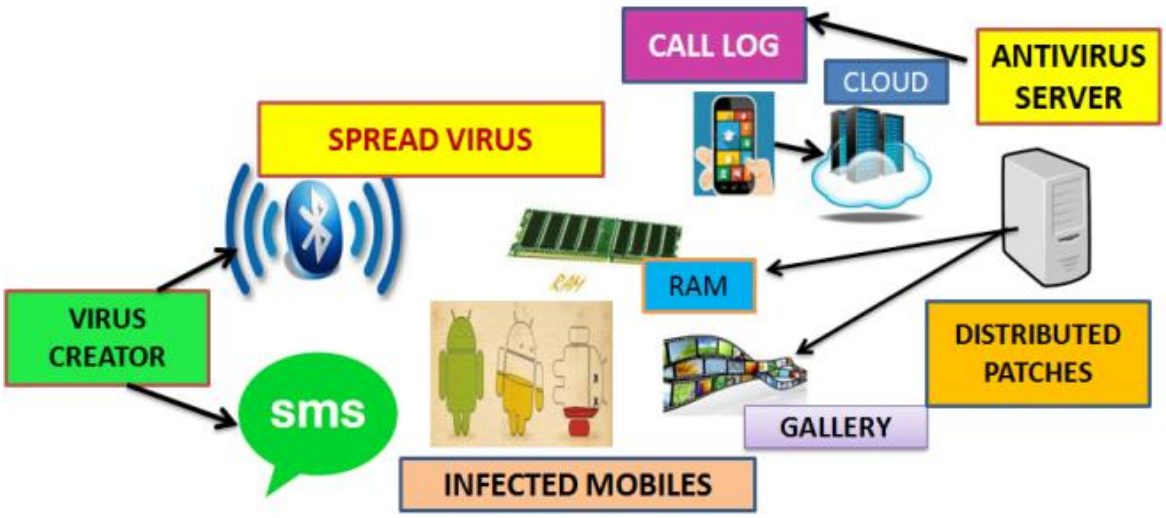
2.1 DRAWBACKS IN EXISTING SYSTEM

The privacy mode can empower users to flexibly control in a fine-grained manner what kinds of personal information will be accessible to an application. Also, the granted access can be dynamically adjusted at runtime in a fine-grained manner to better suit a user's needs in various scenarios (e.g., in a different time or location).

2.2 PROPOSED SYSTEM

The proposed system explains about different scenario of attacks which can steal our contacts, login credentials, text messages, or maliciously subscribing the user to costly premium services.

III. SYSTEM ARCHITECTURE



IV. MODULES

The system after carefully analysis has identified to be present with the following modules.

- Modeling virus
- Monitoring call logs& crashing gallery
- Virus propagation
- Distribution of patches

V. MODULES DESCRIPTION

Modeling Virus

In this Module we will create the Mobile Virus which is malicious code that will perform malicious activities in the User's Mobile Phones. In this Project we are creating a New Folder Virus which will create a Folder inside the Folder virus by developing malicious codes. So that we can generate the Mobile Virus. Once the attackers created the Virus, they will spread the Virus via Bluetooth or SMS technique, So that the virus will be spread to other Users Mobile Phones. While sending via Bluetooth technique, the User's has to be present within the communication range. The Attacker can send the virus file via Mobile Application that was installed in their Mobile Phones.

Monitoring Call Logs & Crashing Gallery

In this module we will create the mobile virus and spread the mobiles, call logs are stored in cloud server like missed call and dialled calls. Also spread the gallery so all data's are changed in encrypted format. So does not view the gallery.

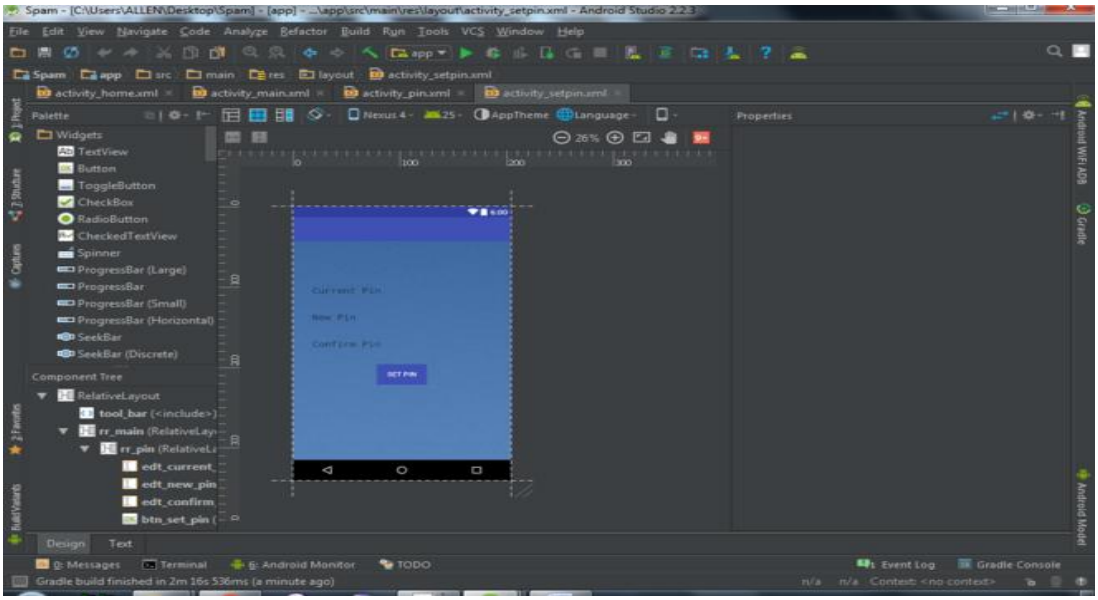
Virus Propagation

Once the attack spread the Virus File to other User's Mobile Phone the content of the message of the file will be analyzed by the Server to detect whether the file contains that Malicious behaviour or not. If the file contains the malicious behaviour, then the Server will detect the file as Virus file. Once the Server detected that the Virus file it will deliver the patches to the User's Mobile Phone and deletes the Virus File.

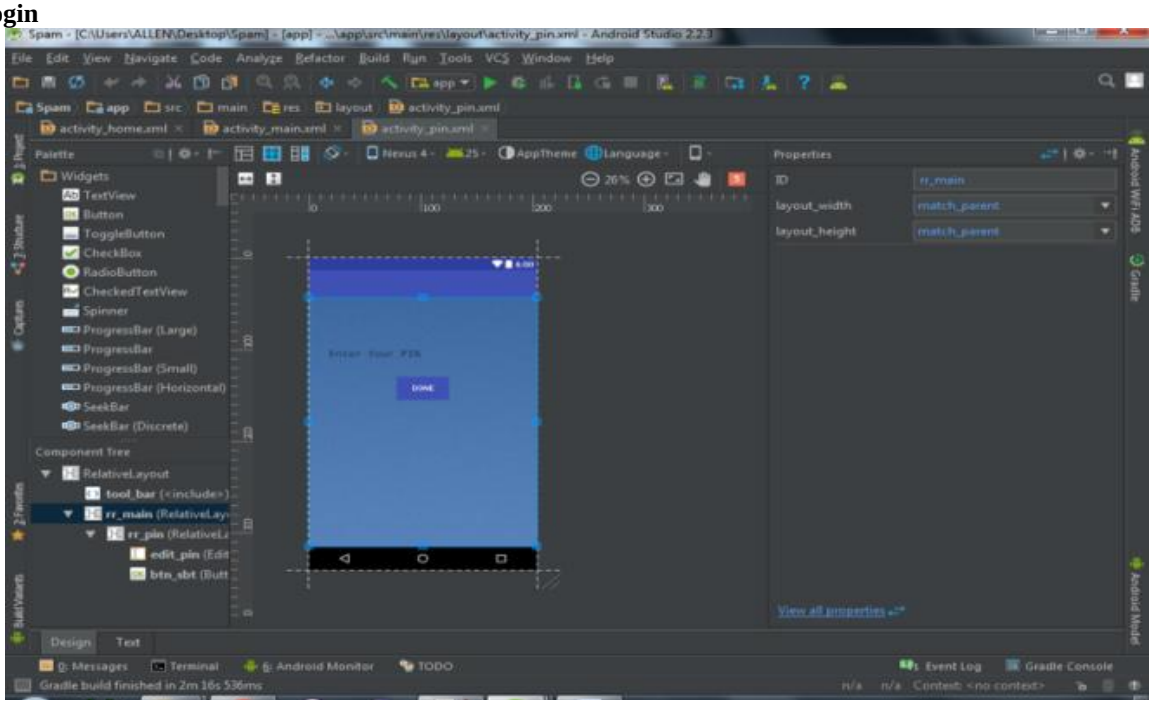
Distribution Of Patches

Once the Server identify virus file was sent to the User's Mobile Phone, the Server will provide the patch files to delete the Virus file. Using an Android Application the patches will be distributed to the User's Mobile phone automatically to clear the Virus.

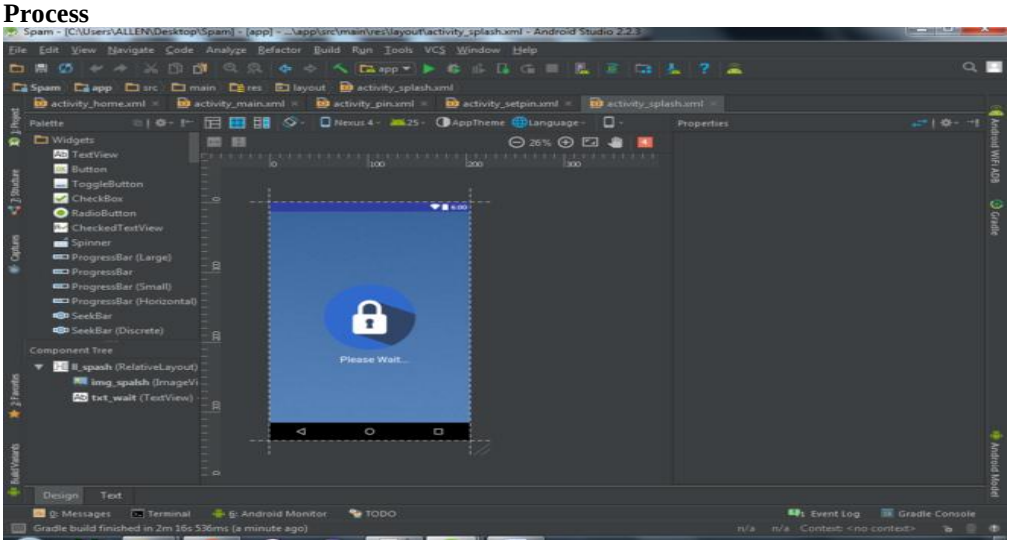
Registration



Login



Loading Process



VI. CONCLUSION

Starting from the end of 2011, attackers have increased their efforts toward Android smart phones and tablets, producing and distributing hundreds of thousand of malicious apps. These apps threaten the user data privacy, money and device integrity, and are difficult to detect since they apparently behave as genuine apps bringing no harm. This paper proposes MADAM, a multi-level host-based malware detector for Android devices. By analyzing and correlating several features at four different Android levels, MADAM is able to detect misbehaviours from malware behavioural classes that consider 125 existing malware families, which encompass most of the known malware. To the best of our knowledge, MADAM is the first system which aims at detecting and stopping at run-time any kind of malware, without focusing on a specific security threat, using a behaviour-based and multi-level approach. Not only the accuracy of the runtime detection of MADAM is very high, but it also achieves low performance (1.4%) and energy overhead (4%).

FUTURE ENHANCEMENT

Android smartphones and accurately detect and delete the virus of the content before enter into the mobile. Future work can be enhanced the virus content of the data's enter into the smartphone through Bluetooth and SMS . It automatically filter the virus and data separately and delete the virus but not the data

REFERENCES

[1]. M. Backes, S. Gerling, C. Hammer, M. Maffei, and P. von Styp-Rekowsky, Appguard 2014 fine-grained policy enforcement for untrusted android applications,' in Data Privacy Management and Autonomous Spontaneous Security, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

[2]. S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A. Sadeghi, and B. Shastry 2012 'Towards taming privilege-escalation attacks on android,' in 19th Annual Network and Distributed System SecuritySymposium, NDSS 2012, San Diego, California, USA.

[3]. M. G. Christian Funk 2013, 'Kaspersky security bullettin'.

[4]. W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth 2010, 'Taintdroid: An information-flow trackingsystem for realtime privacy monitoring on smartphones,'in Proceedings of the 9th USENIX Conference on OperatingSystems Design and Implementation, ser. OSDI'10. Berkeley, CA, USA: USENIX Association.

[5]. A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner 2012, 'Android permissions: user attention, comprehension, and behavior,' in Symposium On Usable Privacy and Security, SOUPS'12, Washington, DC, USA .

[6]. K. S. Labs 2014, 'Kindsight security labs malware report '.