International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 09 , September 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Counter RIG Attack System

Atul Chandrakant Jadhav[1], Sunil P. Khachane[2]

[1]Student,[2]Professor, Department of Computer Engineering,

Rajiv Gandhi Institute of Technology, Mumbai (India)

ABSTRACT

Stealing of intense data from apps is at all times measured to be one of the most dangerous threats to Android reserve. This can happen to the apps without obvious implementation weaknesses, through abusing some design flaws of the mobile operating system, e.g., mutual communication channels a malicious app needs to run abreast with the target app (such as messaging, phone book, dialer, IoT interface, Bluetooth control service, Internet browser, etc.) to collect its runtime information. To arrange for defence against this new sort of attacks, here is a research of a proficient & inventive system which does not require any alteration of prevailing systems such as on operating system level as well as on applications level. This system will be capable of safeguarding any app from any category. This new line of security, called Counter RIG Attack System, spoils a malicious app's runtime monitoring challenge by suspending (stopping & putting on hold) all apprehensive background processes when the target app is running in the forefront, and reinstating the state of all apprehensive background processes from the state where they had paused after the target app ends execution completely and its runtime environment is sanitised.

Keywords: Android Security, Counter RIG Attack, Information Security, Mobile Security, Runtime Information Gathering.

I. INTRODUCTION

Android devices are extremely popular because of millions of paid & free applications called as apps. Apps are reason for mammoth dispersion of android based mobile systems and nowadays become unavoidable part of it with rapid development. Every day & night android market space is flooded with newer apps from different categories with purpose to provide services such as media, Education, Finance, Banking, Medical, Entertainment, Security, etc. All of these apps needs to access, process and transmit subtle information such as personal, financial, business related activities (e.g., bank account details, diseases and medicines history, investment secret, etc.) that needs to be secure from criminal or malicious programs installed and running on same device at simultaneous time. Android OS provides a security mechanism for preventive apps from accessing each other's data by providing sandboxing environment for apps execution and providing unique process Id's. This protection however is not sufficient to counter Runtime information gathering through shared communication channels (e.g., audio, Bluetooth) or public resources (e.g., memory, CPU usage). Vital data could still be exposed to the malicious app that continuously monitors the prey app's activities and gathers its runtime information from those shared sources. Runtime information gathering pose austere threat to even newest version of android devices and confidentiality of its users[1].

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Runtime information gathering is any activities that involve in maliciously collecting the data processed, produced, transferred or received by another app that does not mutually agrees to share its data during the execution, in an effort to directly steal or indirectly conclude sensitive user information. Such an attack can happen by abusing the permission the malicious app attained from the user, e.g., a non-messaging app reading all incoming and outgoing messages without user's consent, recording phone conversation by a non-dialer app [2], [3], which was granted RECORD_AUDIO permission at the time of install and extracting sensitive data such as transaction PIN or CVV no. [4]. Also, a Bluetooth operated medical device's genuine app can give away vital information [5]. A big concern here is that even zero permission can still gain highly profound data from a variety of side channels, signifying the important weakness of mobile devices in extrication an app's operations from its data. Examples include Internet browsers can give away web content detected through memory foot prints; phone's accelerometer coordinate values stolen from shared channel can reveal key strokes logged [6]. Case in point comprises recording cell phone talks from the phone app, gathering medical history data to conclude the ailment condition the user, etc. This runtime information gathering (RIG) menace is convincing and severe, as validated by earlier study and new discoveries, these malicious apps can read entire message threads, can control Bluetooth data transmission, can read, alter and even spoof contact in the device (which can lead to a Social Engineering attack through which device user can fall prey to the name displayed but with attackers contact no. beneath, and can misinterpret or overlook it as a trusted source), can steal passwords and other credentials no matter if they are encrypted or plain format, can control IoT enabled devices which are connected by means of shared communication channel to handset, can disclose user location, can collect phone call recordings and decipher pin from voice sample, etc.

In Android operating system applications are divided in foreground and background instances where any app can be stopped and resume as per need of operating system. Counter RIG Attack System takes advantage of this treatment given to applications in android. Counter RIG Attack System will also be carefully designed to select the true instants to start and end the shield process, and effectively safeguard itself against malevolent apps. The experimental studies show that this new Counter RIG Attack System independent of OS version works well with small influences on the efficacy of legitimate apps and the performance of the OS. Most essentially, the inkling underlying this approach, includes providing protection at the level of application, defence at the level of side channel with no compromise to performance of the device.

II. EXISTING SYSTEM

Existing solutions for countering Runtime Information Gathering attack needs a modification of either the Android Operating System or the exposed applications which are at risk. Refining the access control mechanism in android system is also possible to evade the danger of Runtime information gathering but this undesirably affects the system's usability. Android provides security to each application by sandboxing its space which treats each app uniquely thus providing process identification and file system access control. Each app is treated as a user thus assigned a user ID (UID), in order to separate them from each other. Thus keeping shared resources out of the scope of this security mechanism causing easy resource sharing such as Bluetooth, Internet connection, audio, camera captures, etc. among multiple apps. Each app at the time of its installation must be granted a permission to access these shared resources. There are different protection levels [7] assigned to each

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

permission, such as some permission are automatically granted to the apps when prompted, some risky permissions need user's consent, and critical system permissions for system apps. Apps can access these shared resources only with a proper permission granted.

There are some serious flaws in this security mechanism. Such as once a permission is granted then there is no control of user or OS over how and when that grant is utilised, For example, a non-messaging app with MESSAGE_READ permission can access all messages as per its will and wish, an app with CONTACT_READ and INTERNET_CONNECT permission can transmit sensitive contacts from phonebook, an app with AUDIO_RECORD permission can catch all phone conversations. Added to this no protection provided to runtime information flow between apps which can lead to a RIG attack. Malicious apps running in the background can cause severe damage to privacy of user. When any such RIG attack happens and detected then Google & sometimes manufacturer of mobile device come up with a security patch which again possesses a huge scope of success of other RIG attacks. Scope of application developers revisiting to develop patch for vulnerable app is too much time consuming and technically impossible because no app can override the permission grants given to any other app.

III. PROBLEM DEFINITION

Android is ingenuous to endure the RIG menace. The operational restrictions, such as public channels and shared resources, expose it to abundant practices of runtime information gathering, which resulting into the disclosure of intense user information. This vulnerability is genuine, persistent and severe, thus only new methodologies or techniques can possibly tackle and provide actual solution with suitable installation over millions of android devices across globe.

Thus there is immense need to develop a security system capable to counter known RIG attacks and provide secured environment for execution of target app and defend user sensitive data from leaking through side & shared communication channels on the application level, without touching the OS or the target apps under protection, this system will be called Counter RIG Attack System, that can be accessed from Google Play store and installed on any Android device to acquire immediate protection of user's target apps by taking advantage of side channel information and detection of malicious apps with suspending these malicious apps and avoid permission exploitation granted at the time of install.

IV. CONTRIBUTION

In the offensive world of information security, one must believe offence Is the best defence and applying the philosophy "to catch a thief think like a thief" so in this scenario of Runtime information gathering where side channels are exploited by misusing the permissions granted, proposing here an offensive defence system called Counter RIG Attack system where this system will detect malicious apps by reading their shared channel data and other logs representing their behaviour which is then stored and used for detection of these malicious apps. That means trying Runtime information gathering (RIG Attack) on malicious apps to see if they are involved in Runtime information gathering.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 09 , September 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Counter RIG Attack system works in two different modes as workflow described and shown in Fig.1 below. Counter RIG Attack system can be started from any mode thus providing detection and protection against unnecessary message reads by third party apps or malicious apps.

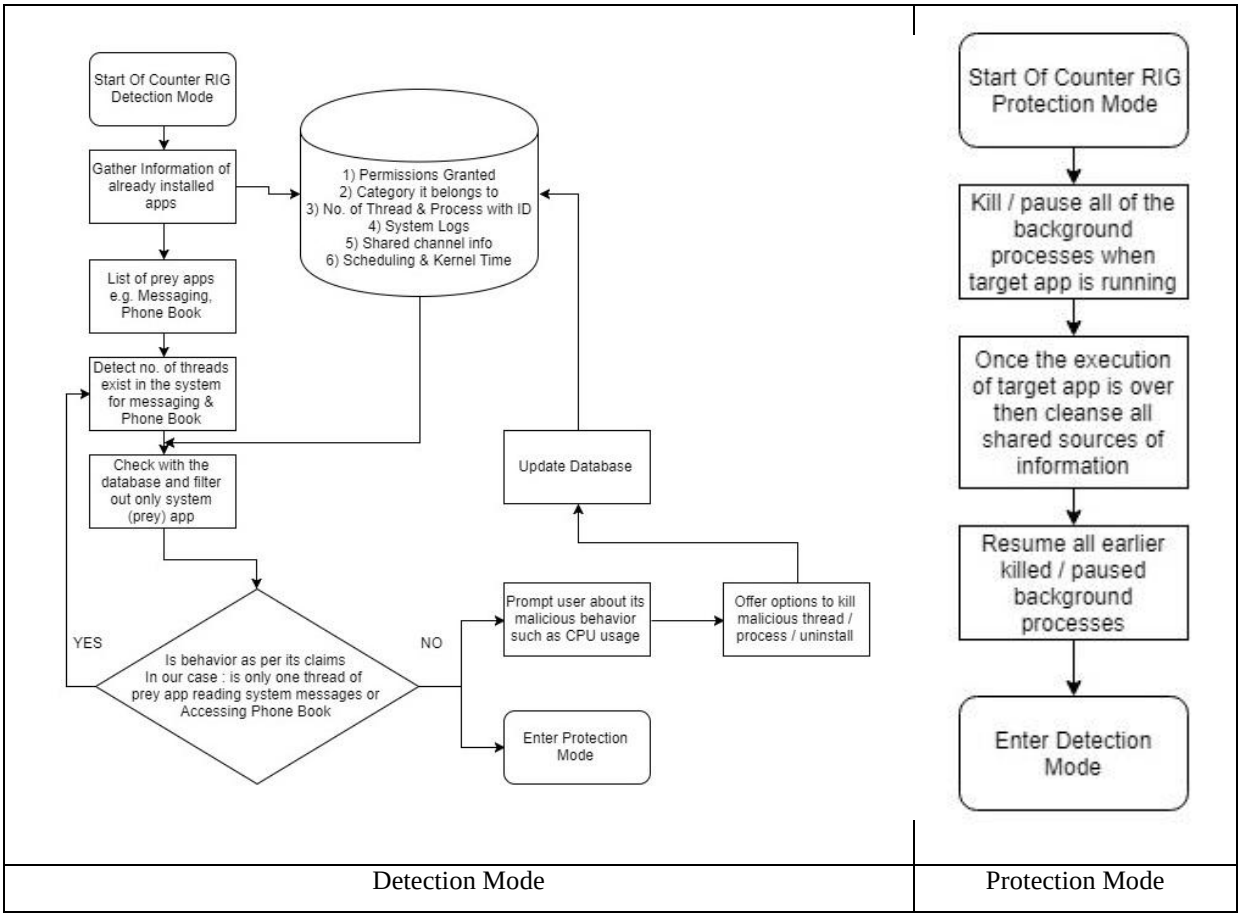

Fig.1 : Functions for Counter RIG Attack System.

First attack vector is on messaging service where situation is worst and even neglected by users & developers because READ_SMS permission is granted for more than 70% (i.e. more than 2 million) of apps on google play store. Whereas 95% out of these apps need not to read more than 1% of messages out of message book during its entire lifecycle of existence (i.e. between install till uninstall of that app) and that too for reading verification code sent over message for 2-way authentication. But with this READ_SMS permission all of these apps can read complete message book from user's device, where every user have one system message management app who reads 100% of messages all the time. Thus absolutely no need of granting READ_SMS permissions for any other apps.

Performance measurement is done against applications which are not system's default messaging apps but still aquired READ_SMS permission at the time of install and tries to expose user information recived or sent in the messaging service. The performance of Counter RIG Attack system is examined by sampling 100 to 200 attempts for each of the application listed in Fig.2, and results of killing or pausing non sysyetm messaging app is given below. Apps with more than 8 score were found to be killed or paused for more that 85% on android device running more than 10 different applications simultaneously.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 09 , September 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

App	Kill / Pause	oom_score_adj	Effective
Facebook	87%	9	Yes
Google play	72%	6	Yes
Imo	85%	8	Yes
Olacabs	91%	9	Yes
Quickr	93%	9	Yes
Truecaller	88%	8	Yes
Twitter	95%	8	Yes
Uber	94%	9	Yes
Kotak bank	69%	5	Yes
Amazon	89%	9	Yes

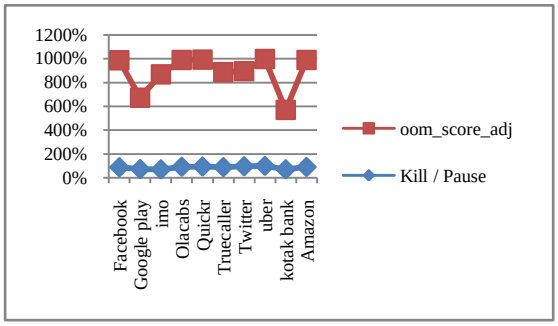

Fig.2 : Performance Measurement against READ_SMS.

Second attack vector is also tried and tested. In this attack vector another area in mobile systems which deals with the contact details information called as Phone Book. The contact details in the mobile device are very sensitive and should be considered utmost private property of the mobile user. Contact information is second most favourite thing in RIG attacks that are besieged by malicious apps around the globe where malicious app attempts to access, update, transfer & wipe out partial or entire contact list without mobile user's consent in the runtime. Android operating system provides a permission grant control for protection of phone book information such as READ_CONTACT & WRITE_CONTACT but once these permissions are granted then application is allowed to modify and play with phone book information at its will. Because RW_CONTACT permission is granted for more than 82% (i.e. more than 2.4 million) of apps on google play store. Whereas 98% out of these apps need not to read more than 2% of contacts out of phone book during its entire lifecycle of existence (i.e. between install till uninstall of that app) and that too for informing user about any contact in phone book is using this app for social networking or automatic service utilization. But with this RW_CONTACT permission all of these apps can read complete phone book from user's device, where every user have one system (default) phone book management app who need to read 100% of contacts all the time. Thus absolutely no need of granting RW_CONTACT permissions for any other apps.

Performance measurement is done against applications which are not system's default phone book management apps but still aquired RW_CONTACT permission at the time of install and tries to expose or misuse user information stored in phone book service. The performance of Counter RIG Attack system is examined by sampling 100 to 200 attempts for each of the application listed in Fig.3, and results of killing or pausing non default phone book management app is given below. Apps with more than 7 score were found to be killed or paused for more that 80% on android device running more than 10 different applications simultaneously.

App	Kill / Pause	oom_score_adj	Effective
Facebook	85%	8	Yes
Google play	90%	9	Yes
Imo	68%	6	Yes
LinkedIn	83%	7	Yes
CiscoWebex	87%	7	Yes
Truecaller	94%	9	Yes
Twitter	75%	7	Yes
Uber	81%	8	Yes
Kotak bank	62%	4	Yes
Amazon	94%	9	Yes

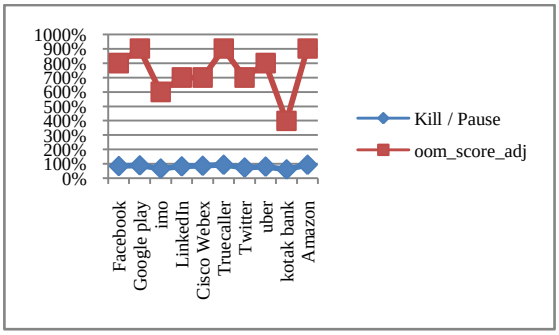

Fig.3 : Performance Measurement against RW_CONTACT.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

V. CONCLUSION

In Counter RIG Attack system for android a function is introduced which exploit the runtime information, system logs, and other related behavioural information (such as no. of threads present, CPU usage when running in background, source of installation, category it belongs, etc.) of malicious apps for killing or pausing them and avoid sensitive user information from malicious access. Concentrating on two vectors type of RIG attack that is illegal message read & exploitable Phone book access, it is examined that this Counter RIG Attack system is found efficient against different category of apps who are not message management apps or default system message apps in case of first attack vector and default phone book management apps in second case of attack vector but still posing threat for stealing information in message system & contact information in phone book of device. This approach can be further extended to study and mitigate other types of RIG attacks such as contact spoofing, phone call recordings, Shared channel information stealing, etc.

REFERENCES

[1] Nan Zhang, Kan Yuan, Muhammad Naveed, Xiaoyong Zhou and XiaoFeng Wang, "Leave Me Alone: App-level Protection Against Runtime Information Gathering on Android" in IEEE Symposium on Security and Privacy, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7163068/

[2] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, "Identity, location, disease and more: Inferring your secrets from android public resources," in Proceedings of 20th ACM Conference on Computer and Communications Security (CCS), Nov. 2013. [Online]. Available: http://www.cs.indiana.edu/~zhou/files/fp045-zhou.pdf

[3] S. Jana and V. Shmatikov, "Memento: Learning secrets from process footprints," in Proceedings of the 2012 IEEE Symposium on Security and Privacy, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 143–157. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.19

[4] R. Schlegel, K. Zhang, X. yong Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A stealthy and context-aware sound trojan for smartphones." in NDSS. The Internet Society, 2011. [Online]. Available: http://dblp.uni-trier.de/db/conf/ndss/ndss2011.html#SchlegelZZIKW11

[5] M. Naveed, X. Zhou, S. Demetriou, X. Wang, and C. A. Gunter, "Inside job: Understanding and mitigating the threat of external device misbonding on android," 2014.

[6] L. Cai and H. Chen, "Touchlogger: inferring keystrokes on touch screen from smartphone motion," in Proceedings of the 6th USENIX conference on Hot topics in security, ser. HotSec'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 9–9. [Online]. Available: http://dl.acm.org/citation.cfm?id=2028040.2028049

[7] "Android permission," http://developer.android.com/guide/topics/ manifest/permission-element.html/, 2014.