PROTECTION AGAINST INFORMATION LEAKAGE AND MALWARE WITH RISK FREE PUBLIC WIFI USAGE

K. Ananda[1], P. Ujwala[2], Himanshu Kemwal[3]

[1]*Assistant Professor, Department of ISE, Sri Krishna Institute of Technology, Bengaluru, India*
[2]*Student of Bachelor of Engineering, Department of ISE, Sri Krishna Institute of Technology, Bengaluru, India*
[3]*Student of Bachelor of Engineering, Department of ISE, Sri Krishna Institute of Technology, Bengaluru, India*

Abstract

As the number of Android devices and Android users are increasing day by day, the risk of Android devices being attacked is also increasing. The attackers tries to gain some knowledge about the potential victims such as their IP address and MAC address and then uses those knowledge to access the victim's data, deploy malwares, take backup of the user's data or to modify the victim's data. To avoid this we propose a way using which we can avoid information leakage and can detect malware. Using this, we can also use the free WiFi at public places and avail Bluetooth promotions offers without any fear of being attacked. For this we make use of HTTP packets and signatures for user authentication, to ensure that the data is received from authentic person.

Keywords: *— Android, Authentication, Free WIFi, HTTP packets, Information leakage, Malware.*

----------***----------

1. INTRODUCTION

Many Android devices are exposed to highly vulnerable environment when they connect to any free WiFi connections and Bluetooth promotions that are present in the public places such as malls and metro stations. Any person who is having a little bit of knowledge about networks can easily access the network to gain knowledge about the people connected to the public WiFi. Using such information, anyone can easily spoof anyone's Android device and can use it to send malicious messages to the victim to maybe access his data by sending a malware, or a API call or system call coded within a message to take a backup of user's data or if not that then attacker can try to generate junk data to fill victim's storage space.

Android is an open source software and it is based on linux, so it is very difficult to make malwares to access an Android device. Unless and until the malware designed for particular kernel build of Android is used in same kernel build of Android that malware is not work. So, Android environment is highly secure.

Android also uses java. Java is a high level language which can be characterized by the buzzwords like simple,architecture neutral, object oriented, portable, distributed, high performance, interpreted, multithreaded, robust, dynamic, secure.With most programming languages, you either compile or interpret a program so that you can run it on your computer. The java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into a intermediate language called java bytecodes- the platform independent codes interpreted by the interpreter on the java platform. The interpreter parses and run each java byte code instruction on computer. Compilation happens just once; interpretation occurs each time a program is executed.

We propose a method to cut-off the sources of information leakage such as messaging apps, public WiFi sources, Bluetooth promotions in the malls, and the third party apps.Since 99.99% of the times the attacker will try to hide his identity before attacking an Android user. So, first step in our method is to authenticate all the Android device users who try to send and receive the data. For this, we take data to be sent and that is broken into the clusters, to which the hashing is done and signature is attached and that message is sent to the receiving Android device which verifies the signature to authenticate the sender. Second step is to ensure that the data sent is not modified in midway of the transmission and the data is same as the data sent by the sender.

Multicast is an efficient way of transmitting the data from the sender to the receiver or group of receiver which works efficiently for the real-time applications such as games, messaging apps, video chat, video calls etc. But multicasting requires a way to properly authenticate the users of each device. The advantages of using multicast are that it provides data integrity, data origin authentication and non-repudation. Data integrity refers to the ability to assure that the received packets have not been modified during transmission. Data origin authentication means that the packets received are from the source that it claims to be from. Non-repudation means that the sender of a packet should not be able to deny sending the packet to receivers in case there is a dispute between the sender and receivers. All the three services can be supported by an asymmetric key technique called signature. The existing malware detection techniques detect malwares based on the permissions called by the malwares during execution of the application. Or it uses the suspicious API calls or system calls to detect the presence of malicious event. But these techniques may not detect the intercomponent communication between components of Android. So we try to remove data which might contain the malware before data is stored in the receiver.

2. RELATED WORK

Signature Generation: There are various signature generating techniques proposed. In "Review of Signature for private Information Leakage in Android in Android Application" by D.N. Rewadhkar [1], they focus on the issue that arises for the Android users which comes in the free Android apps as advertisements which usually collects the user sensitive data and broadcasts it across the network. Users must accept this business model to make use of free applications but in most of the cases information is broadcasted without the knowledge of the user.In this, for the signature generation they use combination of signature set created from hierarchical cluster results. In "MABS in network security" by Shrikanth Bethu and Asrar Ahmed [2], they propose a technique to propose a technique for Multicast Authentication by making use of the batch signatures by maintaining efficiency. Kyo-Hui Yeh and NAI-Wei Lo proposed a framework called AppLeak [3] for information loss evaluation, privacy leakage detection, and privacy risk assessment on Android applications to cope up with the variety of new user-privacy frauds.

Mobile Malwre Detection: For detecting the static malwares many malware detection techniques analyze the behaviour of the app codes for malicious code without running the apps. For instance, Kirin [4] detects malwares depanding on violation of the permissions and Stowaway [5] detects over-priviledges in Android apps by mapping API calss to permissions. DroidMiner [6] and DroidAPIMiner [7] detects sensitive API callsfor detecting presence of malware. ICCdetector [8] tries to detect malwares which tries to bypass the detection by inter-communication between components of Android system.

In "Context-based Access Control Systems for Mobile Devices" by Bilal Shebaro, Oyindamola Oluwatimi, Alisa Bertino[9],in this they propose a context-based access control mechanism by which privileges can be dynamically granted or revoked to applications based on the specific context of the user by modifying the Android operating systemso that the context based restrictions can be formed and enforced.

3. SYSTEM MODEL AND ANALYSIS

The proposed method consists of mainly sender, receiver, and verifier/server. In this, the server is responsible for selecting the data to be sent, its fragmentation, hasing and adding up of the signature to it. Fragmentation is done to ensure that when receiver recwiver the data which contains malware, it should not work in the Android system, malware will not work unlesss all piece of malware code is put together.Reciever needs to receive the data sent by the sender and its dehashing, which is then sent to verifier to verify its content and owner. Verified data is stored in the receiver and unverified data is discarded. This method can be applied on the servers of third party apps provider or in the system of public WiFi providers.Using the proposed method they can easily maintain security for the users.

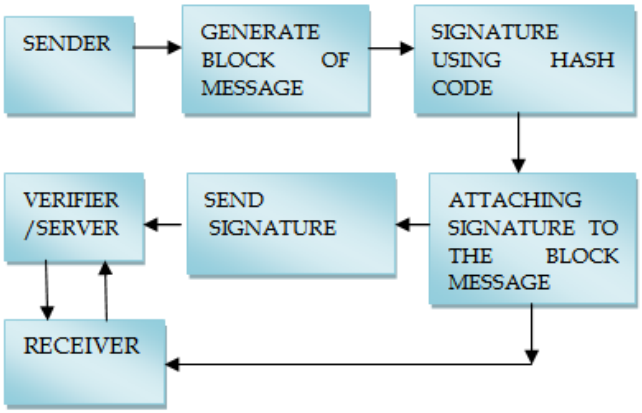

Fig. 1. The proposed system architecture

3.1 Sender

The sender initiates the data communication. Sender need to select a file/message to be sent. The selected data is divided into byte of data to form a cluster. For each cluster hashing is done and the signature is generated which is attached to the data of cluster and the data is combined together in a single HTTP packet. And the HTTP packet is sent to the receiver.Along with HTTP packets sender sends the cluster ID to the receiver and Cluster ID and signature is sent to the verifier. This hashing and adding signature later helps us in the authentiaction of the authentic person which is very important for the protection from information leakage.

3.2 Receiver

The receiver receives the cluster ID and data coming from the sender but doesn't store it in the device. Instead receiver performs dehashing on the clusters received using which receiver gets a signature. Receiver sends the cluster ID and the signature of the received data to the verifier/server where the verification of the data and the sender is done.

3.3 Verifier

Verifier first receives the Cluster ID and signature from the sender which it stores for later. Later verifier receives the Cluster ID and signature sent by the receiver. Cluster ID is used for identification of clusters sent and the signatures are compared. If both the signatures are matched then it means that the data is received from authentic source and data integrity is maintained and data is then stored in the receiver else the data is discarded. It also assures that the data was not modified while transmission and so it may not have malware in it.

4. FLOW CHART

The following flowchart tells us the flow of our method.

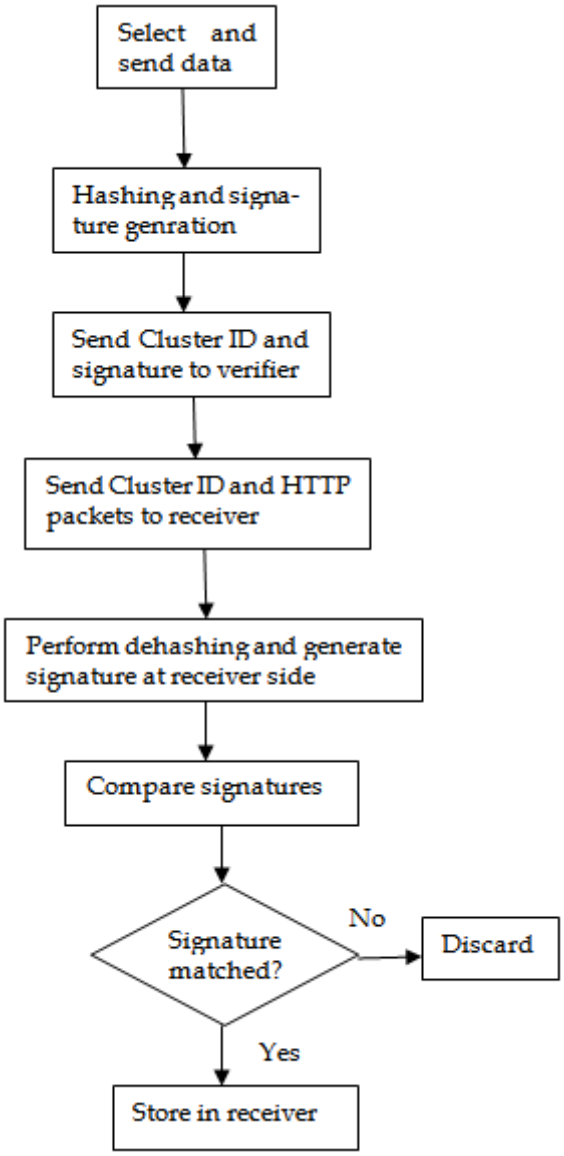

Fig. 2 Flow chart of the system

5. CONCLUSION

Using our proposed method, we can protect our device from the information leakage and it also provides data origin authentication and data integrity. Protection from malwares is also provided.Android users can use Bluetooth promotions and public Wi-Fi without any fear of being attacked. Signatures are generated for every cluster generated so to reduce signature verification overheads in the secure multicasting, block-based authentication schemes have been proposed.But the vulnerability to packet loss and resilience to denial of service(DoS) attack. Clustering in general is useful for pulling together patterns in large amounts of data, but the number of the generated signatures tends to increase with cluster size, and can produce signatures that match most network packets, if the signature generation is applied carelessly. It has been difficult for a signature approach to achieve high detection rates using real dataset.Probablistic signatures might improve detection of information leakage on Android applications.

REFERENCES

[1] Pavitra Mangesh Ratnaparkhi, Prof. D. N. Rewadkar, "*Review of Signature Generation for Private Information Leakage in Android Applications*", International Journal of Advanced Research in Computer Science and Software Engineering 4(8), August - 2014, pp. 1067-1071.

[2] Srikanth Bethu, Asrar Ahmed M.D. , Jesurun Prem Kumar Dasari "*Multicast Authentication Based on Batch Signature (MABS) in Network Security*" ,K.R Venugopal and L.M. Patnaik (Eds.) ICCN 2013, pp. 180-188.

[3] Kuo-Hui Yeh, Nai-Wei Lo, Chuan-Yen Fan, "*An Analysis Framework for Information Loss and Privacy Leakage on Android Applications*", Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference.

[4] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 235–245.

[5] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, 2011, pp. 627–638.

[6] C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras, "DroidMiner: Automated mining and characterization of fine-grained malicious behaviours in Android applications," in *Proc. 19th Eur. Symp. Res. Comput. Secur. (ESORICS)*, Wroclaw, Poland, Sep. 2014, pp. 163–182. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11203-9-10

[7] Y. Aafer, W. Du, and H. Yin, "DroidAPIMiner: Mining API-level features for robust malware detection in Android," in *Proc. 9th Int. ICST Conf. Secur. Privacy Commun. Netw. (SecureComm)*, Sydney, NSW, Australia, Sep. 2013, pp. 86–103. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-04283-1-6

[8] Ke Xu, Yingjiu Li, and Robert H. Deng, "*ICCDetector: ICC-Based Malware Detection on Android*", in IEEE Transactions on Information Forensics and Security Volume: 11, Issue: 6, June 2016 pp. 1252-1264.

[9] Bilal Shebaro, Oyindamola Oluwatimi, Elisa Bertino, "*Context-based Access Control Systems for Mobile Devices*", DOI 10.1109/TDSC.2014.2320731, IEEE Transactions on Dependable and Secure Computing Volume: 12, Issue: 2, March-April 1 2015 pp. 150-163.