ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (821-826), Month: April - June 2015, Available at: www.researchpublish.com

Detecting and Awarding Sensitive Information: Leakage in Android Application

[1]UJWALA S GANGASHETTI, [2]SANDHYA A, [3]B.R.PRASAD BABU

[1]M.Tech Student, Department of CSE - R&D Centre, SEACET, BANGALORE – 560049, ujwalais056@gmail.com
[2]Lecturer, Department of Computer Science and Engineering, SEACET, BANGALORE – 560049
[3] Prof & Head (PG), Department of Computer Science and Engineering, SEACET, BANGALORE – 560049

***Abstract*: in current years, there has been quick growth in mobile gadgets such as smartphones, and smartphone market has established with plenty of applications. In particular, there are several applications that are "free" to the user, but depend on advertisement services for their expenses. Such applications include an advertisement module - a reference library provided by the advertisement service - that can assemble a user's sensitive information and transfer it across the network. Such data is used for some accepted advertisements, and user performance statistics. Users receive this business model, but in most situations the applications do not require the user's acknowledgment in order to transfer sensitive information. Hence, these applications' behavior becomes an incursion of privacy. In our investigation of 1,189 Android applications' network traffic and permissions, 93.01% of the applications we studied connected to multiple destinations whenever we are using the network.**

***Keywords:* Security, Smartphone, Privacy**

1. INTRODUCTION

The fast evolution of smartphones has led to popularity for mobile gadgets. With the growing popularity of smartphones and tablets, advance for mobile device operating systems has drastically improved, especially the growth of applications for online marketplaces.Applications presented on such marketplaces are characterised as free or paid. The mutability of the markets also offers huge security challenges.

A smartphone holds many kinds of private data, such as location tracking information, the subjects to the user's address book, and the unique device identifier. In order to maintain security, Android offers a framework which needs applications to have specific permissions for retrieving limited resources. Applications or advertising segments with definite permission combinations can send the user's sensitive data to the outside servers using the network. While this data is generally used for directed advertising, it can also be discovered and used by mischievous parties without the inventive user's awareness [1].

Our popularity-focused security analysis for the most regularly used applications. Our results inform the following wide explanations.

- We found inclusive misuse of privacy important information—particularly phone Identifiers and geographical location. Phone identifiers, e.g., IMEI, IMSI, and ICC-ID,were used for tracing to accounts numbers.
- We found no clues of telephony misuse, background recording of audio or video, Abusive connections, or harvesting lists of installed applications.
- Ad and analytic network libraries are combined with 51% of the applications. Numerous applications include more than one ad library.
- Many developers fail to steadily use Android APIs. These failures generally comes under the classification of unsatisfactory protection of privacy sensitive data.

Background:

MULTICAST is an effective method to distribute multimedia content from a sender to a group of receivers and is gaining standard applications such as real-time stock quotes, interactive games, video conference, live video broadcast, or video on request. Authentication is one of the serious topics in securing multicast in an environment attractive to mischievous attacks. Basically, multicast authentication may offer the following security services:

1. **Data integrity**: Each receiver should be capable to reassure that received packets have not been altered during transmissions.

2. Data origin authentication: Each receiver should be capable to reassure that each received packet comes from the real sender as it privileges.

3. Nonrepudiation: The sender of a packet should not be capable to refute sending the packet to receivers in case there is a argument between the sender and receivers.

II. SYSTEM DESIGN

2.1 Block Diagram

Here we are presenting all the events of the signature generation. Initially, the sender chooses the data. Then that data is categorized into Block of Messages with 48bytes each. Later that, a unique signature is generated based on hashing technique for each of these message blocks. These signatures along with the Message Block ID is send to the Verifier and simply the Message blocks (i.e. data). The receiver accepts this message and generates signature. This signature is again sent to the Verifier, if both the signature along with their corresponding block ID are same then the Message is accepted else rejected.

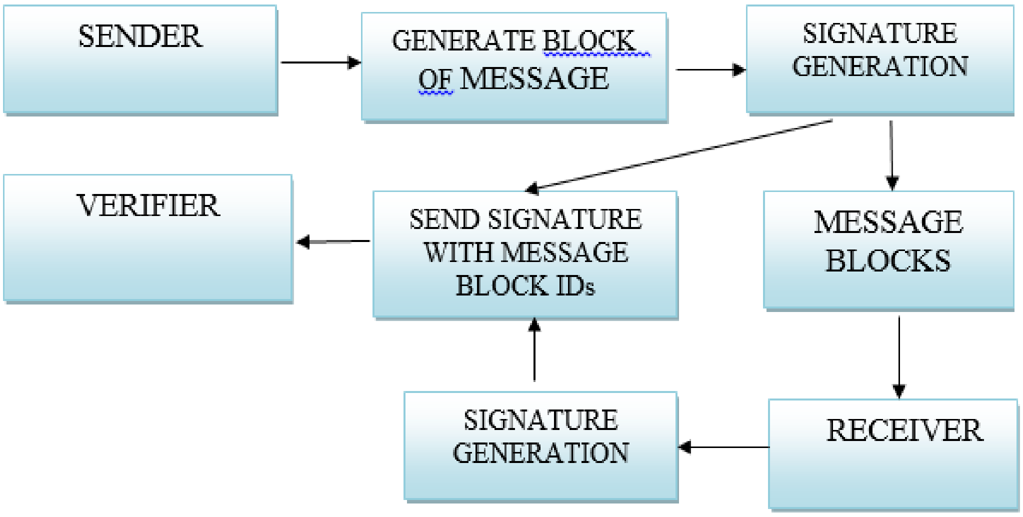

Figure 2.1: System design

2.2 **Architecture of our clustering and signature generation System.**

Below Figure shows our methodology, which involves of two parts. Initially, a separate server (shown in Figure 2.2) gathers application traffic, clustering the data and generating signatures. Second, an information flow control application on the user's device (shown in Fig 2.2) gets signatures from the servers and manages the transmission of other applications' network traffic.

The server produces signatures by the following method. Initially, it generates a payload check, which separates application network traffic into two clusters: one holding packets with thoughtful information and the other not. Second, the server clusters the group containing sensitive information based on packet destination distance and contents distance, and builds a signature using conjunction signatures.

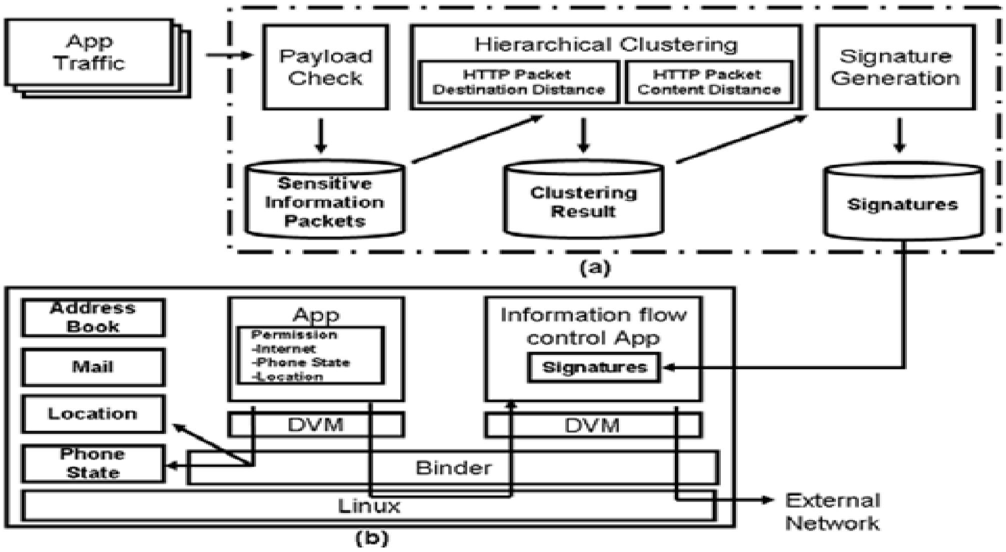

Figure 2.2 The architecture of our clustering and signature generation System (b)The information flow control application that uses the signature generated by(a) Source[1].

III. LITERATURE SURVEY

Signature generation technique permits users to resistor the transmission of their private information. This signature support to preserve security, in which it needs application to have particular permission for accessing limited resources. This method results in a high percentage of true positives, and a low percentage false positives. The author advises effective and secure algorithms to prove multiple digital signatures based on the discrete logarithm. Instead of verifying each individual signature individually, it is proposed to validate multiple signatures simultaneously. The proposed Clusters verification algorithm can maintain a constant verification time as to verify a single signature.

3.1 Methods of Signature Generation:

The signature techniques are suitable for real-time information purifying on personal communication system (PCS). Signature technique may be commonly applied to several types of information media and this technique is mainly good for sensitive information filtering condition. Signatures are very easy to create and search, thus they are suitable for personal communication where real time searching with restricted buffer space is required. There are three types' signatures

1. Simple signature.

2. Integrated signature.

3. Multi-level signature

Signatures are created from the information frames and spread together with the information frames. The signature may be broadcasted as a group before the information [2].

3.1.1 Simple Signature:

The signature is generated with information frames is to build a frame signature for a piece information frames. The signature frame is advertised before the corresponding information frames.

3.1.2 Integrated Signature:

A overview of the simple signature scheme is to produce a signature, called an integrated signature, for a group of one or more information frames, called a frame group. The integrated signature is broadcast before the frame group. In this scheme, a signature is indexed by any number of information frames.

3.1.3 Multi-level Signature:

The multi-level system is a grouping of the simple signature and integrated signature schemes. It contains of various levels of signature. Signature at the upper levels are integrated signature and those at the lowest are simple signature. The signature is produced for each level of the signatures. After tuning into the broadcast channel, the corresponding signature are used to match with different levels of signature. If a signature flops in the comparison, the PCS switches to doze mode until a signature at the same or upper levels arrives. Otherwise, the PCS stays in active mode and continues the filtering process. If match found, the corresponding information frame is received; if not, the PCS may go into doze mode till the next signature arrives.

3.2 Generating Signature for Worm:

James Newsome et al.. Present Polygraph, a signature generation system that successfully generates signatures that match polymorphic worms. They adds a definition of the polymorphic signature generation problem; suggest classes of signature suited for identical polymorphic worm payloads; and present algorithms for automatic generation of signatures in these classes. A network administrator organizes an intrusion detection system (IDS) at the gateway between his edge network and the Internet, or on an distinct end host. The IDS searches inbound traffic for known designs, or signature. When such spiteful traffic is found, the IDS may raise an alarm; block upcoming traffic from the source address; or even block the remainder of the flows traffic [2].

3.3 Disadvantage of Existing System:

Packet loss is unavoidable. In the Internet, congestion at routers is a main reason affecting packet loss. An overloaded router drops buffered packets according to its predetermined control policy.

The insecurity of wireless channel can cause packet loss very frequently. Moreover, the smaller data rate of wireless channel increases the congestion possibility.

IV. PROPOSED SYSTEM

We represent a method of clustering technique which takes certain HTTP packets to generate signatures that can correctly identify new HTTP packets containing sensitive data. Our major concern is not malware, but free applications which risk leaking sensitive information. In many cases, malware is identified by the anti-virus software. Conversely, free software that agrees sensitive information leakage is not malware but still presents a threat to the user’s confidentiality. In our calculation we observed the number of HTTP packets that involved sensitive information and sent it to external servers. We then applied clustering to a model of the developed data to generate signatures, and re-applied these signatures to the whole dataset. This method resulted in a high percentage of true positives, and a low percentage false positives. Thus, we determine that our generated signatures have enough accuracy for identifying sensitive information transmissions.

Our technique essentially consider:

- Clustering method using HTTP packet distance that finds the likeness between the two of Android application network packets.
- A system, using the clustering method followed by signature generation, which can discover sensitive information leakage without modifying the Android framework.

V. IMPLEMENTATION

5.1 Generation of Block Messages

In this component we splinted the selected data into small packages. The next step generate a block of messages has been taken from the each packet. The block generation depends on the receiver.

We consider a receiver-oriented approach by taking into explanation the heterogeneity of the receivers. As receiving devices have different computation and communication competences, some could be powerful desktop computers, while the others could be low-cost handsets with restricted buffers and low-end CPUs. Mixed with several channel loss rates, this heterogeneity pretenses a demand on the capability of altering the buffer size and authenticating buffered packets at all time when the high layer application involves at each receiver.

5.2 Cluster Signature Generation

In cryptography, the **Boneh–Lynn–Shacham** signature scheme permits a user to authenticate that a signer is authentic. This scheme uses a bilinear pairing for verification and signatures are set of elements in some elliptic curve. Working in an elliptic curve affords protection against index calculus attacks against permitting shorter signatures than FDH signatures. Signatures are often mentioned to as short signatures, BLS short signatures, or simply BLS signatures. The signature scheme is provably secure assuming both the actuality of random oracles and the uncontrollability of the computational Diffie–Hellman problem.

5.2.1 The Signature Generation Scheme:

A signature scheme contains of three tasks that are generate, sign and verify.

Key generation:

The key generation algorithm chooses a random integer x in the interval [0, n− 1]. The private key is x. The owner of the private key distributes the public key g^x.

Signing:

Given the private key x, and some message m, by using hashing the bit string m we compute the signature, as h=H(m).

We produced the signature sigma=h^x.

Verification:

Using a signature sigma and a public key issued by the private key holder, we verify that e (sigma,g)=e(H(m),g^x), where 'e' is a pairing function of BLS scheme.

5.3 Transmission

After producing the signature it will be attached to the block of messages. Formerly the block of messages will be transferred to the destination by using the receiver's IP address as input.

At the destination side message will be saved such that receiver can validate the signature in the meantime.

If the result is true then message will be stored at the receiver otherwise the message will rejected.

5.4 Verification

In the ClusterVerify () algorithm should varify the following properties:

- Given a Cluster of packets that have been signed by the sender, ClusterVerify() results True.
- Given a Cluster of packets containing some unauthentic packets, the possibility that ClusterVerify () results True is very low.

The computation complexity of ClusterVerify () is comparable to that of verifying one signature and is increased only steadily when the Cluster size n is increased.

VI. CONCLUSION

To moderate the signature verification outflows in the protected multimedia multicasting, We proposed block-based authentication systems. Unfortunately, most previous schemes have various problems such as susceptibility to packet loss and lack of rigidity to denial of service (DoS) attack. To overcome these problems, we presents a novel authentication. We have validated that our authentication is perfectly resilient to packet loss due to the rejection of the correlation among packets and can efficiently deal with DoS attack. Moreover, we also display that the usage of Clusters signature can attain the efficiency less than or comparable with the conventional schemes.

Numerous Android applications involve permissions for sensitive information access and network features, and that among applications that are connect to several outside servers without the user's acknowledgment. Applications network performance contains a large amount of sensitive information, mainly, UDIDs and Android ID which are immutable identifiers. We have introduced a novel clustering method using HTTP packet distances which contain both the distance between HTTP packet destinations and between HTTP packet contents. Using that clustering method in combination with signature generation in our dataset succeed 94% accurate detection of packets having sensitive data with only 3% false positives.

REFERENCES

[1] Hiroki Kuzuno," Signature Generation for Sensitive Information Leakage in Android Applications", Intelligent Systems Laboratory, SECOM, Tokyo, Japan 2013

[2] J. Newsome, B. Karp, and D. Song, "Polygraph: Automatically generating signatures for polymorphic worms," in IEEE Security and Privacy (S&P 2005), May 2005.

[3] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A study of android application security," in 20th USENIX Security Symposium 2011, Aug.2011.

[4] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droid You're looking for: Retrofitting android to protect data from imperious applications," in 18th ACM Conference on Computer and Communications Security (CCS 2011),

[5] M. Grace, W. Zhow, X. Jiang, and A.-R. Sadeghi, "Unsafe exposure analysis of mobil in-app advertisements," in 5th ACM Conference on Security and Privacy in Wireless Mobile Networks (WiSec 2012),Apr. 2012.

[6] Chaeon, "An Efficient Id-Based Signature Scheme with Clusters Verifications"

[7] Yoon, " Secure Id-Based Signature Scheme with Clusters Verifications"

[8] J. Jeon, K. K. Micinski, J. A. Vaughan, N. Reddy, Y. Zhu, J. S. Foster, and T. Millstein, "Dr. android and mr. hide: Fine-grained security policies on unmodified android," in CSTR-5006, Dec. 2011