International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Information Leakage Detection and Risk Assessment on Privacy

Miss. S. Gayathri[1], *Mr. R. Ambikapathy*[2]

[1](M.C.A), Department of MCA, Krishnasamy College of Engineering and Technology
[2]M.C.A., M.Phil., Assistant Professor Department of MCA, Krishnasamy College of Engineering and Technology

ABSTRACT

This study focused on designing a web-based information security system to secure transaction details using the AES (Advanced Encryption Standard) algorithm to encrypt and decrypt text and image files. As the success of a financial fraud usually requires possessing a victim's private information, new types of personal identity theft and private information acquirement attack are developed and deployed along with various Apps in order to steal personal private information from mobile device users. This project presents the security and compression for the data with the advance encryption standard (AES). Advanced Encryption Standard is one on the most regular and mostly symmetric block cipher algorithm. This algorithm has a specific form to encrypt and decrypt subtle data and is put in all hardware and software. It is highly tough to hackers to get the actual data when encrypting by AES. Till date there is no proof to crack this algorithm. AES has the capacity to deal with 3 dissimilar key sizes such as AES 128, 192 and 256 bits. Each of its code has 128 bits.

Key Terms: AES Algorithm, Information Security, Encryption, Decryption

I. INTRODUCTION

WITH the increasing universality of smartphones, a lot of mobile applications (a.k.a. Apps) installed on smartphones or other forms of mobile devices have been developed to provide value-added services for individual mobile users and enterprise employees. As mobile users usually use their mobile devices when they are moving, Apps often require Internet access to provide their full functionalities, and data transmission between an App and its corresponding backend server(s) is inevitable most of time. To support Apps running smoothly and effectively, it is necessary to design suitable operating systems for intelligent mobile devices such as smartphones and tablets. Among mobile operating system technologies, Web is one of the most popular platforms and has successfully stimulated the generation of numerous Web-based online App markets all over the world. Recently, a critical challenge for the Web platform has been to resolve the concern from mobile users regarding the leakage of user-privacy-related information while executing Apps.

With more than 50% of smartphone market share , Web-based mobile phone vendors and Internet service providers have to face the new challenge on user privacy management. Since a Web platform is built on handheld mobile devices in general, resource-limited hardware components of mobile devices and the power saving requirement for device runtime have largely restricted the possibility for a Web-based mobile device to run full-fledged resource consuming security solutions constantly as the backend processes of Web. As a result, computation efficiency needs to be considered when designing user privacy solutions on Web-based devices. Apps often interact directly with personal or system sensitive data retrieved from local sensors embedded in mobile devices such as a GPS signal receiver, an embedded digital camera, a hidden microphone, and an accelerometer. However, mobile users (and enterprise employees) do not always know if these data acquired by Apps have been shared and used properly. Without appropriate management or monitoring mechanisms, utilization and transmission of such information may be insecure and vulnerable to malicious attacks from the Internet.

With an increasing number of mobile services and online transactions that utilize sensitive personal information of mobile users, the risk of encountering identity theft attack and other misuse of personal private data has been also rising up quickly in recent years. It is therefore highly risky for mobile users when personal sensitive data are transmitted by running Apps under a public network environment without monitoring by any security checking scheme. Traditional data protection mechanisms utilize encryption algorithms and signature certificates to ensure data secrecy and integrity.

An entertainment App may be developed to unnecessarily request permissions to access the Internet and get the user's current position from the GPS receiver in his mobile device. Often, the user blindly accepts a permission request like this one without knowing how his personal data will be used by the running App. In another example, an instant message App can be exploited by adversaries to launch social engineering attacks. A malicious adversary may try to deceive specific users for their credit card information using spear phishing or click-jacking techniques. To manage user privacy for mobile users, we propose the border patrol concept.

The idea is to prevent valuable user-related information or data from being transmitted by running Apps through various communication channels, such as the Internet, Bluetooth connection, and near field communication connection, without notifying the user. We design a user privacy analysis framework called Leakage Detection to realize the proposed concept. There are five modules defined in the Leakage Detection framework. Two modules are designed to monitor user input operations and message transmission operations from running Apps. Three modules are used to support different levels of user privacy management based on a privacy analysis model developed by us. To evaluate the proposed framework, we have implemented a system prototype called the Leakage Detection App running as a backend service process on a Web smartphone. Two general App usage scenarios are conducted with Line App to verify the effectiveness of our Leakage Detection App.

The privacy risk assessment status for every targeted App appointed by the user can be displayed in the Leakage Detection App. In our scenario 2, the Line App gets the high-risk warning signal on user privacy assessment, as shown in. The status is assigned based on the privacy measure model of privacy risk assessment and the information from App execution data flow measure. In this case, the user has revealed his birthday and credit card number to a self-claimed customer service clerk via Internet connection. Based on the privacy risk assessment measure model defined in Section V, the value range of P Rγ for the Line App is between 0.394 and 0.238. Assume we set the threshold value P Rmedium as 0.290 and the threshold value P Rhigh as 0.342 by dividing the value range of P Rγ into three equal parts.

As the P Rγ value for the Line App is 0.367 in this case (scenario 2), which is larger than the threshold value P Rhigh, therefore, the privacy risk assessment status for the Line App is set to high, as shown in Fig. 4(c). A comparison among Leakage Detection and recently published user privacy protection systems for the Web platform is shown in Table II. In general, most frameworks/systems adopt a dynamic analysis approach; some of them use a static analysis approach at the same time. Only WebLeaks [4] utilizes a static analysis approach solely. Only Leakage Detection introduces the user perception concept onto a determination mechanism for user privacy disclosure in which individual users can actively control the semantic definition of his own user privacy. In addition, Leakage Detection provides three different levels of user privacy management functionalities to reduce user management workload and support flexibility on module extension and addition at each level.

II. LITERATURE SURVEY

Numerous studies have attempted to identify malicious Apps and the leakage of private information through these Apps. Most of these studies have been carried out from the perspective of a Web-systematic scheme or a data- flow-based approach. On the other hand, recent research has been conducted using data mining concepts in order to increase efficacy of locating malicious patterns in mobile Apps that may cause privacy leakage. Here, we introduce the studies that are the most relevant to Leakage Detection. In 2010, Enck et al. proposed a static taint analysis method to detect potential paths that lead to privacy leakage.

The authors then evaluated 30 popular Web Apps involving location, camera, and microphone data. They found that half of them sent sensitive information to remote servers for advertisement purposes, although none of the Apps that were found to have sent sensitive information explained this use of information in advance to the App user. In 2012, Gibler et al. developed a program analysis framework for Java source and byte code. The authors created a call graph of the execution flow of the specific App's code and performed a reachability analysis to examine if privacy information could be sent out over the network.

To secure privacy information, their proposal mainly adopted two techniques:

1) a set of mappings between Web API methods and the required permissions as the sources and sinks of private data for dataflow analysis; and

2) Web Leaks, a static analysis framework for finding potential leaks of private information in Web Apps. In a test of 24 350 Web Apps, they found potential privacy leaks involving identical phone information, location data, WIFI data, and recorded audio. In the same year, Frank et al. Proposed a machine learning technique that would allow for the examination of a Web Apps' permissions pattern. They surveyed almost 188 380 Apps available from the official Web Market in 2011. Using the Boolean matrix factorization, the authors found commonality in permission request patterns for the higher-rated Apps.

III. PROPOSED SYSTEM

To protect the data from attackers, AES algorithm is used. AES encryption technique is the process of converting the data to stop it from attackers to read the original data clearly. Encryption involves conversion of plain text to unreadable format. It is known as cipher text. The user cannot read the above format. Hence, the next process that is carried out by the user is Decryption. In order to secure data AES encryption technique is used in this project. Advanced Encryption Standard is a block cipher with a block length of 128 bits. It permits three different key lengths: 256, 192, 128 or bits. Encryption algorithm is the process of scrambling data so that it cannot be read by doing various substitutions and transformations in plaintext (original message) and converting them into ciphertext (random messages).

System Modules

1. Register and login

2. Purchase Item

3. Attacker

- Secure Payment
- Unsecure Payment

Module Description

REGISTER LOGIN MODULE: -

In login module the customer and merchants can login to the application if they already created their account and signed in.

PURCHASE ITEMS:

User purchase some items through online with their online payment. When we use online payment, we have entered our card details also.

UNSECURE PAYMEN

Both encrypting and decrypting cookies is a difficult process since it requires additional coding. Due to the time involved in the encoding process, the application's performance will be affected. Cookies are not restricted based on internet usage. Whenever a user surfs the web, more and more cookies will be accumulated. Unless the user deletes them, these cookies will be a part of the hard drive space. This eventually slows down or lags the browser.

Apart from security, privacy is another concern for users in cookies. Whenever the user browses the internet, the cookie enabled sites will be recording all the online activities. Most users are unaware that such information is stored on their hard drive. As a result, this information can be accessed by any third parties including government agencies and businesses.

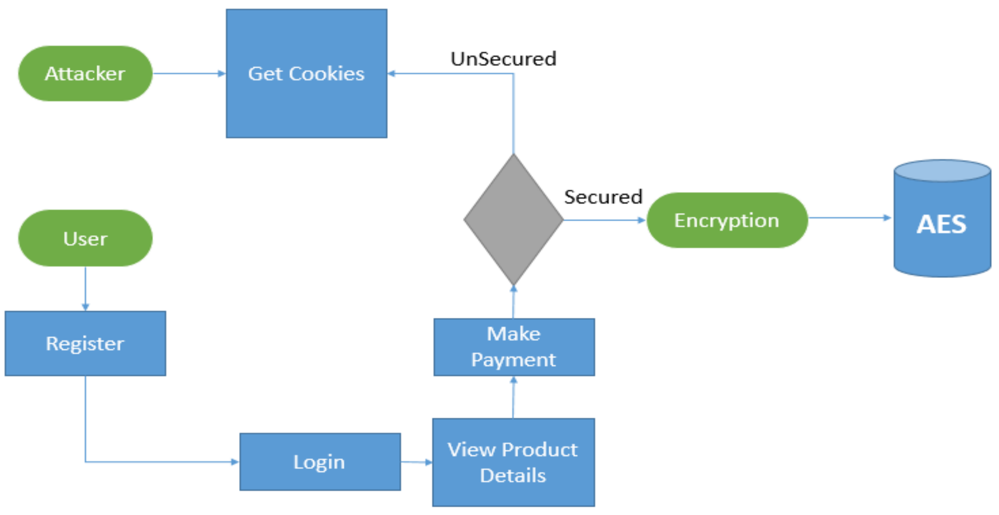
Since cookies are stored in the hard drive as text files, it possesses some serious security risks. Any intruder can easily open these files and view the information. And also, not all the sites that collect information from cookies are legitimate. Some of them can be malicious that uses cookies for the purpose of hacking.

SECURE PAYMENT:

After user purchasing attacker enter the system and view the user transactions details. Like card number, pin number and etc.

After getting validation from application interface from the cardholder, Card details will be sent to merchant bank. When card details will be shared with the merchant bank then the in between process will be secured by AES algorithm at first stage. Card information will be encrypted and will be converted into cipher text so that attackers can't access the data. At acquiring bank end, Key will be decrypted to ensure the validity of transaction which includes ensuring the availability of transaction amount, authenticity of card. The acquiring bank captures all the information and then it directs the transaction to the appropriate card network. Before routing the transaction to the card network, card Information will be encrypted again to ensure the security. Card network will again decrypt the information and it will send the transaction details to the Issuer's bank for approval. Issuer Bank will send one-time 8-digit password to the registered mobile number with bank to the card holder and Transaction will be completed. Card pin will not be used to authenticate the transaction as card pin is not dynamic and it can be stolen easily. After user purchasing attacker enter the system and view the user transactions details. Like card number, pin number and etc.

IV. ARCHITECTURE DIAGRAM



V. CONCLUSION

Personal information leakage and privacy disclosure are critical concerns to mobile users when they utilize different Apps installed on their mobile devices to accomplish various works. In this paper, a user privacy analysis framework called Leakage Detection has been proposed for an Web platform to offer a user privacy management model. In the Leakage Detection framework, we defined required models to achieve user privacy management: App execution data flow, user perception, leakage awareness, information leakage detection, privacy disclosure evaluation, and privacy risk assessment. To support the proposed privacy analysis model, two information capture modules for Leakage Detection were designed to acquire incoming data inputted by a mobile user and outgoing data transmitted from a targeted App. A system prototype based on the Leakage Detection framework was developed to evaluate the feasibility and practicability of Leakage Detection. Two general App usage scenarios were adopted during the usage of Line App to evaluate the effectiveness of Leakage Detection on user privacy disclosure by social engineering attack, user information leakage from normal operations of a running App, and privacy risk assessment of targeted running App. Our experiments have shown that the system prototype of Leakage Detection supports user privacy management effectively on a Web.

VI.REFERENCES

[1] S. E. Whang and H. Garcia-Molina, “A model for quantifying information leakage,” in Proc. Secure Data Manag., vol. 7482, Lecture Notes in Computer Science, 2012, pp. 25–44.

[2] C. D. Manning, P. Raghavan, and H. Schtze, Introduction to Information Retrieval. New York, NY, USA: Cambridge Univ. Press, 2008.

[3] W. Enck et al., “TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones,” in Proc. 9th USENIX Conf. OSDI, Berkeley, CA, USA, 2010, pp. 1–6.

[4] C. Gibler, J. Crussell, J. Erickson, and H. Chen, “WebLeaks: Automatically detecting potential privacy leaks in Web applications on a large scale,” in Proc. 5th Int. Conf. TRUST, Berlin, Germany, 2012, pp. 291–307.

[5] M. Frank, B. Dong, A. Porter-Felt, and D. Song, “Mining permission request patterns from Web and Facebook applications,” in Proc. IEEE ICDM, 2012, pp. 870–875.