Improved Mobile Application Security Mechanism based on Kerberos

Jiabin Sun, Zhao Gao

International School, Beijing University of Posts and Telecommunications, Beijing, 100876, China

Keywords: Kerberos, security mechanism, keys.

Abstract: This paper focuses on the improvement of mobile application security mechanism. Security problem of mobile application is a great challenge nowadays. This paper proposes a security model of mobile application which is based on Kerberos authentication. In this improved security model, every request from the client will be authenticated by the Kerberos server, but the authentication will be different between users with different access. And the classification of the users can be implemented by using different kinds of keys in the ticket-granting server.

1. Introduction

With the popularity and serious security challenges for the mobile application, it is necessary to discuss how to improve the security mechanism and prevent potential attacks. The paper discuss the issue from the following aspect. In the first part, the mobile application security model will be discussed. This includes the main threats and the currently used security for mobile application. And in the second part the paper will have a briefly introduction about the Kerberos authentication system, on which the improved model based. And in the third part, the paper will introduce how the mechanism of Kerberos can be introduced in mobile application and the advantage in doing that.

2. Mobile application security model

2.1 Main threats

The Main threats of mobile application usually lies in the following three aspects:

2.1.1 Vulnerabilities of the application

The problems of the mobile application itself , which mainly refers to the risk of mobile application in configuration, code, compiler process.Vulnerable Applications are apps that allow an attacker to exploit sensitive information, perform actions against user's command, stop a service from proper functioning, or download apps forcibly to your device without your permission[1] [2].

2.1.2 Malicious body

The data communication security problem, which mainly happened in the communication process between mobile application and server, such as clear transmission, the certificate without checking. The malicious body can be categorized into the following types: malicious users [3], malicious third-party applications[3], and malicious access to back-end system.

(1) Malicious users

The main threats from the malicious users is the unauthenticated access. Mobile applications that don't properly manage sessions or that provide local mechanisms for remembering user IDs and passwords are easily compromised. For example, sessions are often left open on mobile applications for long periods of time so mobile users can seamlessly pick up where they left off when bringing an application to the foreground. Not closing open sessions on a regular basis increases the likelihood that a malicious user can gain unauthorized access to critical data and applications.

(2) Malicious third-party applications

Applications that can be downloaded from pirated software can initiate different types of security

DOI: 10.25236/iwmecs.2019.017

issues for mobile devices. “Malicious apps“ may pretend to look authorized on a download site, but their intensions are only to commit fraud. Even some verified software can be misused for fraudulent purposes. These threats are:

Malware[3] is software that performs fraudulent activities while installed on a mobile device. Malware can make charges to phone bill without the knowledge of end user, send unprompted or spontaneous messages to contact list, or give efficacious control over specific device.

Spyware[3] is a program that uses private data without taking permission of owner. Spyware usually attacks on target's phone call history, contact list, text messages, user's current location, photographs, browser history and email. Attackers can use this stolen information for cyber crimes.

Privacy Threats may be caused by software applications that are not exactly program codes that can be used to modify or erasing mobile data. They can misuse sensitive information stored in mobile device for some malicious purpose.

(3) Malicious access to back-end systems

A common attack in Web applications is to circumvent the front end and attempt to gain authorized access to a back-end system. Mobile applications are susceptible to these same types of attacks, but they often provide attackers with additional back-end system information, helping them breach security[3].

2.2 Security mechanism

In view of the security threats of these mobile fields, the research on mobile security is mainly carried out in the following aspects:

Mobile operating system security testing. Different from the traditional operating system safety, the database of mobile system vulnerabilities has not been established currently, so the security evaluation based on vulnerability is difficult to achieve in the mobile terminal. Therefore, security function testing, combined with security risk assessment method is suiTable for the security testing of mobile applications[4]. The key point of the security function test is to test whether the system to be tested has some specific security functions under a security mechanism in the evaluation criteria. While based on the results of the security function test, the assessment results of entire operating system or system security mechanisms could be given. Figure 2 [4] shows the security assessment process of applications.

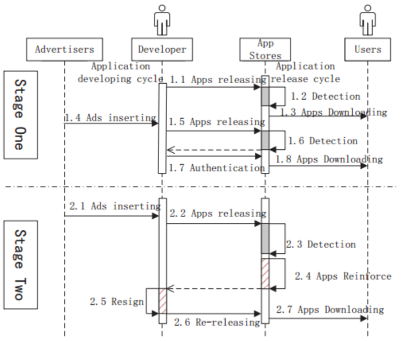


Figure 1. Security model of the application

Access control and privacy protection. The key lies in the monitor and authority control of user access to privacy data. Representative research includes the TISSA access control scheme proposed by the North Carolina State University researchers and the SEAndroid project

3. Kerberos authentication service

In this paper Kerberos 4 is mainly discussed, since the two version of Kerberos: Kerberos 4 and Kerberos 5 share the same authentication process. And this authentication service mainly focus on the solving the problem of user authentication. The process of the traditional Kerberos authentication is shown in figure 1[6]. First, the definitions of abbreviation used in the figure 1 are made as follows: KDC, key distribution center; AS, authentication server; TGS, ticket granting server. Figure 1 shows the general meaning of 6 steps: (1) request the authentication ticket; (2) issue the authentication ticket and TGS session key; (3) request server ticket; (4) issue server tickets and server session keys; (5) request services; (6) reverse authentication[5].

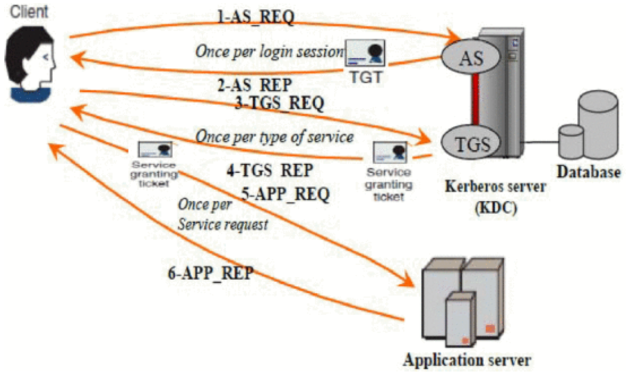
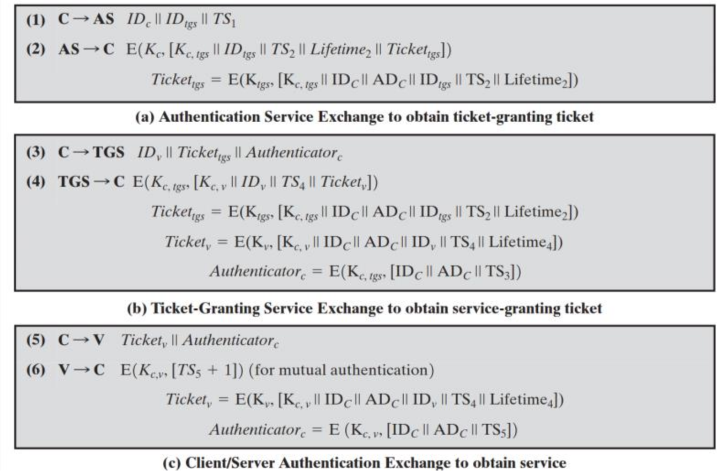


Figure 2. Kerberos authentication process

Second a process for detailed analysis with relevant symbol would be described, but firstly symbols definition should be made as follows: C, client user; AS, authentication server; TGS, ticket authorization server; S, application server; IDc,IDtgs,IDs , the corresponding entities pin; TS_1, TS_2, TS_3, TS_4, corresponding time stamp of data packets, used to prevent replay attack; Kc,Kc,tgs,Ks,Kc,s , corresponding key; Lifetime $_1$, Lifetime-corresponding tickets validity; Ticket tgs , Ticket, corresponding authorization ticket; ADc , the client address; Authenticator sc authentication information, used to authenticate identity who shows tickets and user identity[5].



(1) **C → AS** $ID_c \| ID_{tgs} \| TS_1$

(2) **AS → C** $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) **C → TGS** $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) **TGS → C** $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) **C → V** $Ticket_v \| Authenticator_c$

(6) **V → C** $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E\,(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

Figure 3.

The first stage happens once per user logon[6], and it includes the steps (1) and (2) : C sends request to AS, asking authorization ticket to get access to TGS, and then AS sends back the information encrypted by Kc include Ticket tgs , ID tgs , time stamp TS_2 and so on. Kc is the registered user password key, and here it is the one-way Hash function encryption results of the user password. The encryption key of Ticket tgs is the public key of TGS [5].

The second stage happens once per type of service [4] and it includes the steps (3) and (4) : C gets Kc by calculating their passwords and this calculating process is transparent to end user, and uses Kc to decrypt the related information returned from AS to get Ticket tgs,Kc,tgs and so on. And then C shows TGS its Ticket tgs, IDs and Authenticator c, at the other hand TGS sends C the server session keys Kc,s, and server authorization ticket Ticket s after verifying user information successfully, and all these information returned from TGS is encrypted by Kc,tgs [5].

The third stage happens once per service request [6] and it includes the steps (5) and (6): C uses Kc,tgs to decrypt the related information returned from TGS to get the Kc,s, and Ticket s s, and then sends Ticket s and Authenticator c to S. And S opens a conversation with C ecrypted by Kc,s, and has the reverse authentication after verifying user information successfully[5].

4. An improved mobile security model

This improved model has basically the same authentication process of 6 steps, and both the Authentication Server(AS) and Ticket-Granting-Server are required. The reason the model introduce Kerberos into mobile application security mechanism is that AS provides the access authentication and TGS guarantees the secure access. The problem is in reality, users may have different access authority, and not all App users are registered users. As a result, the authentication should be divided according to different access authority. In the traditional Kerberos, when the client sends the request to AS in the first step, if AS cannot find the user ID in the database then AS would not send the Ticket-Granting-Ticket to the client, but this is not practical in reality since there are non-registered mobile application users.

To solve this problem, there should be different kinds of Ticket-Granting-Ticket for different access authorities. And the server needs to set restriction on files to confine visits from non-registered users. For the users who only use the application for once and not registered, for the first time they logon, AS will send then a key for non-registered visitor. And as TGS received the request from the users, it can recognize the access authority of the user. After the connection has been established, the user interface can only display the content which matches the visitor's access authority.

The advantage of this authentication process is obvious, using Kerberos, the attacks of impersonation and "workstation impersonation“ will be prevented, since the secret keys cannot be copied, and the users with lower access level will be restricted to certain service of the application. The drawback of this authentication includes the following aspects: 1. replay attacks 2. Dictionary attacks 3. Key storage [7].

First, since every request contains a timestamp, so that the trade-off of the life time is a problem. Second, the message sent from AS to C is encrypted by users private key which is produced with user password, and the users private key is dealt with user password by using one-way Hash function. Therefore, the attackers can collect a lot of data which is from AS back to C, and make dictionary attacks. When the user password is not strong, it cannot effectively prevent the dictionary attacks. Once the user password is obtained, the user identity can be pretend to be random one. Third, Traditional Kerberos uses symmetric cryptosystem, so the shared keys between customers and KDC, between application server and KDC, between KDC and distant KDC have to be established and maintained. And in the expanding Internet using Kerberos protocol, keys management and maintenance become the most difficult problem to solve.

Table 1. The advantages and disadvantages of the Kerberos-based mechanism

Advantages	1. Authenticated users 2. Proprieties separated 3. Lower implementation costs[7]
Disadvantages	1. Trade-off 2. Key storage 3. Dictionary attack

5. Conclusion

The security threat of mobile application is becoming more and more serious, and as a result, a novel security model based on the Kerberos system is designed. As a classic authentication service, Kerberos has obvious advantages in the authentication in distributed system. Different from the traditional Kerberos authentication, this model provides service for access classification by using different kinds Ticket-Granting-Tickets that ensures all accesses are authenticated. But there are also drawbacks in the Kerberos system that needed to be improved, such as the trade-off of the requests' life time, the key storage and dictionary attacks.

References

[1] Suzhen Wang, Jianli Hu, Aizhen Liu and Jiazhen Wang, "Security Frame and Evaluation in Mobile Agent System," 2005 2nd Asia Pacific Conference on Mobile Technology, Applications and Systems, Guangzhou, 2005, pp. 1-6.

[2] S. Vashisht, S. Gupta, D. Singh and A. Mudgal, "Emerging threats in mobile communication system," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Noida, 2016, pp. 41-44.

[3] J. Payne, "Secure Mobile Application Development," in IT Professional, vol. 15, no. 3, pp. 6-9, May-June 2013.

[4] Hang Dong, Chengze Li, Ting Li, Yuejin Du and Guoai Xu, "Research on the security model of mobile application," 2014 Communications Security Conference (CSC 2014), Beijing, 2014, pp. 1-5.

[5] S. T. F. Al-Janabi and M. A. Rasheed, "Public-Key Cryptography Enabled Kerberos Authentication," 2011 Developments in E-systems Engineering, Dubai, 2011, pp. 209-214.

[6] C. Wang and C. Feng, "Security Analysis and Improvement for Kerberos Based on Dynamic Password and Diffie-Hellman Algorithm," 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, 2013, pp. 256-260.CNs-39-CNs-45.

[7] H. Kandil and A. Atwan, "Mobile agents' authentication using a proposed light Kerberos system," 2014 9th International Conference on Informatics and Systems, Cairo, 2014, pp.