[1,*]Juanjuan Gong
[2]Tao Men
[3]Sha Feng
[4]Dongming Yu

An Analysis and Improvement Scheme for the Weakness of Kerberos V5 Authentication

Abstract: - The security protocol of network system can ensure the reliability of network communication. Kerberos protocol is a kind of security verification mechanism based on shared key, which has been adopted for a long time on the Internet. This text introduces the authentication process of Kerberos v5 protocol. To the defect that the authentication server don't authenticate the user's identity but it hold safe in building on the basis whether the user can correctly decrypt the ciphertext message in the protocol, we put forward a scheme that adding in the Challenge/Response mechanism to improve its security in Symmetric Cipher Mechanism. Passing the improved scheme's analysis we can perceive the method that adding in Challenge/Response identity authentication technology will greatly resist Offline Password Conjecture Attack.

Keywords: Secure Protocol, Kerberos V5, Shared Key, Challenge/Response, Identity Authentication.

I. INTRODUCTION

With the rapid development of open network systems, the security and reliability of network systems are facing severe challenges. The security protocol uses encryption technology to achieve cipher text transmission, identity authentication and other functions on the open Internet, which is the basic guarantee for the realization of secure e-commerce. In order to meet the authentication requirements of open systems, the security protocol Kerberos [1] came into being.

Kerberos is a security authentication mechanism based on shared key [2,3], which is developed as the authentication service of Athena program of MIT (Massachusetts Institute of Technology) and has been used on the Internet for a long time. The authentication model is based on Needham-Schroeder and is based on encryption. It provides automatic authentication, data security and integrity services, and key management services for users to access remote servers. The overall design scheme of Kerberos is based on the trusted third-party authentication service of the protocol. Both the client and the server trust the authentication of the Kerberos authentication server. The Kerberos protocol defines a series of communication processes between clients / Key Distribution Center (KDC) / servers to obtain and use Kerberos tickets [4]. Kerberos also has the advantage of enhancing interoperability. In a hybrid environment of multiple operating systems, the Kerberos protocol provides the ability to perform user verification for various computing tasks through a unified user database. Even users who have been verified by KDC on different operating system platforms can obtain seamless network service access through the trust relationship between KDC domains.

At present, there are two versions of Kerberos [5]. Version 4 [MILL88, STEI88] is widely used, and version 5 [KOHL94] improves the security of version 4 and becomes the Internet standard draft (RFC1510). Although Kerberos V5 has made improvements in clock synchronization, ticket authorization service, etc., there are still problems such as offline password guessing attacks. Many literatures have improved the defects of the Kerberos protocol. The literature [6] increases the random number after the client enters the user password to resist the dictionary attack, but does not consider the server-side key storage problem. Reference [7] replaces the server's own key with the session key between KDC and Server to enhance the security of the server, but ignores the insecurity of the client key. Reference [8] introduced the public key system into the Kerberos protocol, so as to remove the personal key and completely avoid the password guessing attack, but it also increased the complexity of the system and the overhead of the system.

In this paper, the original Kerberos V5 protocol is improved in the authentication process. The original protocol has the defect that the authentication server does not perform user authentication but establishes the security based on whether the user can correctly decrypt the ciphertext message. The improved protocol adopts the Challenge /

[1] Leshan Normal University, Leshan, Sichuan, China
[2] Leshan Normal University, Leshan, Sichuan, China
[3] Leshan Normal University, Leshan, Sichuan, China
[4] Leshan Normal University, Leshan, Sichuan, China
*Corresponding author: Juanjuan Gong

Response mechanism in the symmetric cryptosystem to improve its security, so that the protocol can well resist offline password guessing attacks.

II. KERBEROS V5 PROTOCOL IDENTIFICATION PROCESS

Before discussing the working process of the protocol, some symbols and concepts need to be defined [9].

C: Client;

AS: Authentication server;

TGS: Bill Distributor;

V: Application server;

||: connector;

ID_x: the identity of entity X;

AD_x: the network address of entity X;

$Ticket_x$: Client access to X's ticket;

K_x: Shared key between X and AS;

$K_{x, y}$: the shared key of X and Y;

$Authenticator_c$: the authentication code of client C, client C uses Authenticatorc to prove that it is the legal holder of the bill;

Realm: identify the domain to which the user belongs;

Options: Used to request the setting of the Flags identifier in the returned ticket;

Times: for client requests to set the time in the ticket;

Nonce: A temporary interaction number that is reused in messages (2) and (4), a random number used to ensure that the response is refreshed and not used by the attacker.

Flags: Kerberos V5 introduces Flags to support functional expansion, with INITIAL logo, indicating that the certificate is issued by AS, not TGS; the PRE-AUTHEN mark indicates that it is pre-certification, that is, AS is required to give a certificate.

TS: time synchronization allowed identification;

Subkey: the client selects a subkey to protect a specific application session, and if this item is ignored, the session key $K_{c, v}$ in the ticket is used;

Sequence (serial number): Optional to indicate the serial number of messages sent by the server to the client during this session. Sorting messages prevents replay attacks.

The authentication process of Kerberos V5 protocol is divided into three stages and six processes. The whole authentication process model is shown in Figure 1 [9]:

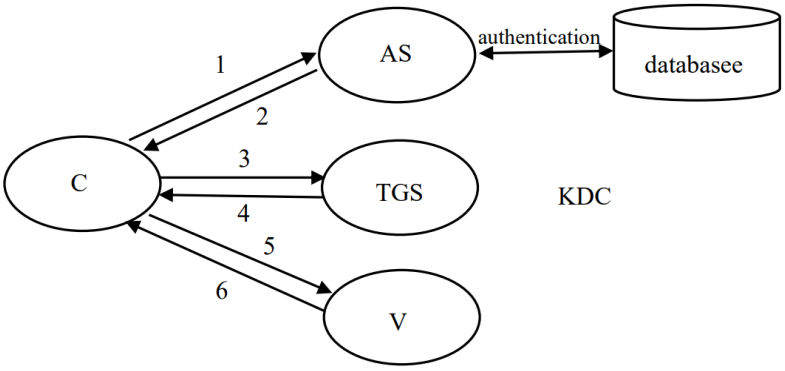

Figure 1: Kerberos Protocol Authentication Process

Stage I: Authentication service exchange, and client C obtains a licensed ticket to communicate with the ticket distributor TGS.

(1) Request for ticket Licensing ticket

C→AS: Options | | ID_c | | $Realm_c$ | | ID_{tgs} | | Times | | $Nonce_1$

The client sends a message to the authentication server requesting a ticket license ticket, including the identity of the user and the TGS, as well as new elements. The definition of these elements is referred to the above instructions. After the authentication server AS receives the message sent by the client, it searches the user's identity in the database. If the user is found, a ticket license ticket is generated.

(2) Ticket licensing ticket

AS→C: $Realm_c$ | | ID_c | | $Ticket_{tgs}$ | | E_{Kc} [$K_{c, tgs}$ | | Times | | $Nonce_1$ | | $Realm_{tgs}$ | | ID_{tgs}]

$Ticket_{tgs} = E_{Ktgs}$ [Flags | | $K_{c, tgs}$ | | $Realm_c$ | | ID_c | | AD_c | | Times]

Message (2) returns the ticket authorization ticket, identifies the client's information and a key-encrypted data block formed by the user's password. The data block contains the session key used between the client and the TGS, the time and temporary interaction number set in message (1), and the identification information of the TGS. The ticket itself contains the session key, the identification information of the client, the set time value, the identification and options that affect the status of the ticket, and is encrypted with the key between AS and TGS.

After receiving the message from AS, the client decrypts the encrypted data block with the key E_{Kc} generated by its own secret password, and obtains the session key $K_{c, tgs}$ for the client to communicate with TGS, which is saved with the ticket licensing ticket $Ticket_{tgs}$.

Stage II: Service authorization ticket exchange, client C obtains a license ticket to communicate with the application server V.

(3) Request server ticket

C→TGS: Option | | ID_v | | Times | | $Nonce_2$ | | $Ticket_{tgs}$ | | $Authenticator_c$

$Ticket_{tgs} = E_{Ktgs}$ [Flags | | $K_{c, tgs}$ | | $Realm_c$ | | ID_c | | AD_c | | Times]

$Authenticator_c = E_{Kc, tgs}$ [ID_c | | $Realm_c$ | | TS_1]

Message (3) contains the client's authentication code, ticket and request service identification, and also includes the time, option and a temporary interaction number of the service authorization ticket request. Authenticatorc is an authentication message sent by the client. This authentication message is encrypted by the session key between the client and the TGS, including the client's identification information and timestamp. This authentication message can only be used once and has a very short lifetime.

(4) Server ticket

TGS→C: $Realm_c$ | | ID_c | | $Ticket_v$ | | $E_{Kc, tgs}$ [$K_{c, v}$ | | Times | | $Nonce_2$ | | $Realm_v$ | | ID_v].

$Ticket_v = E_{Kv}$ [Flags | | $K_{c, v}$ | | $Realm_c$ | | ID_c | | AD_c | | Times].

Message (4) returns the service authorization ticket, identifies the client's information and a data block encrypted with a session key. The data block contains the session key used between the client and the application server, the time and temporary interaction number set in message (3), and the identification information of the application server.

Stage III: client / application server authentication exchange, client C from V to get online services.

(5) Request application services

C→V: Option | | $Ticket_v$ | | $Authenticator_c$

$Ticket_v = E_{Kv}$ [Flags | | $K_{c, v}$ | | $Realm_c$ | | ID_c | | AD_c | | Times].

$Authenticator_c = E_{Kc, v}$ [ID_c | | $Realm_c$ | | TS_2 | | Subbkey | | Seq]

In message (5), the client submits service authorization tickets, authentication codes and options to the application server.

(6) Application server authentication

V→C: $E_{Kc, v}$ [TS_2 | | Subbkey | | Seq]

The client verifies the identity of the application server. The message returned by the server to the client contains a negotiated session key that can be used to encrypt (decrypt) messages between the client and the application server.

III. KERBEROS V5 SECURITY ANALYSIS OF AUTHENTICATION CLIENT

In step (2) of the kerberos V5 protocol authentication process, the authentication server AS does not verify the identity of the user when the user applies for a ticket license ticket, but encrypts the information sent back to the user based on the key K_c. Only those who know the key K_c can decrypt it, but in the kerberos protocol, the user's secret key K_c is converted by the user's secret password through some algorithm. In this way, the kerberos system is vulnerable to offline password guessing attacks due to the existence of obvious plaintext-ciphertext pairs and the small password space of users [10].

The following will analyze how the active attacker implements the offline password guessing attack on the kerberos protocol. In the stage I of the kerberos protocol authentication process, the interaction message between the user and the AS during the authentication service exchange is as follows (RFC1510):

(1) C→AS: Options||ID_c||$Realm_c$||ID_{tgs}||Times||$Nonce_1$

(2) AS→C: $Realm_c$||ID_c||$Ticket_{tgs}$||E_{Kc}[$K_{c,tgs}$||Times||$Nonce_1$||$Realm_{tgs}$||ID_{tgs}]

$Ticket_{tgs}$=E_{Ktgs}[Flags||$K_{c,tgs}$||$Realm_c$||ID_c||AD_c||Times]

Message (1) is initiated by the user. The user uses the plaintext message (1) to communicate with the authentication server AS and requests to return the ticket to access the TGS. The request contains the ticket's lifetime Times and freshness identifier Nonce1 (selected random number).

In the response, the authentication server AS generates a new session key $K_{c, tgs}$; then the generated session key is encrypted and encapsulated in the ticket $Ticket_{tgs}$, and message (2) is returned to the user.

Part of the message used for TGS in message (2) is $Ticket_{tgs}$, which is encrypted with the long-term shared session key K_{tgs} of AS and TGS. Part of the message for the user is E_{Kc} [$K_{c, tgs}$ | | Times | | $Nonce_1$ | | $Realm_{tgs}$ | | ID_{tgs}], which is encrypted using the user's secret key K_c.

After receiving the message (2), the user decrypts the encrypted message to obtain the ticket session key $K_{c, tgs}$, and continues to complete the subsequent protocol interaction process.

Through observation, it can be seen that there are obvious following plaintext-ciphertext pairs in the above interactive messages:

Cipher text: $E_{Kc}[K_{c,tgs}||Times||Nonce_1||Realm_{tgs}||ID_{tgs}]$

Plain text: ID_{tgs}

And it can be seen that the cipher text does not contain a salt operation similar to the encryption key exchange protocol, that is, the'amplification' operation of the password space, so we can easily implement the offline password guessing attack. In a smaller password space, the offline password guessing attack that exhaustively searches the password space will inevitably crack the correct user password within an acceptable time (polynomial time).

Once the attacker obtains the key Kc through the offline password guessing attack, he can obtain the corresponding tickets and application services by using the original user's workstation or configuring his own workstation as the same network address to impersonate the legitimate user.

From the above analysis process, it can be seen that the offline password guessing attack caused by the failure to identify the user when the client applies for the ticket license ticket is a relatively weak point of the kerberos V5 protocol, which can be improved in the symmetric cryptosystem. The improved method in the symmetric cryptosystem can not only avoid the computational complexity brought by the public key cryptosystem, but also reduce the overhead of the system, so this improved method is also a suitable solution.

IV. Methods for Improving Kerberos V5 in Symmetric Cryptosystems

Challenge / Response (Question-Response) mode [11] can be used to identify the user's identity in the process of applying for a bill license. The query-response method is an implementation mechanism of OTP [9,12] (one-time password). During the authentication based on the query-response, the verifier sends a certain value (query message) to the claimant, which participates in the operation of the authentication information. The generated non-repeated query message is completely determined by the verifier, so that the authentication information transmitted each time is different. When the claimant wants the verifier to authenticate, the claimant sends an authentication request to the verifier, and the verifier sends the claimant a random query message n generated by itself. The claimant input password Psw, Psw and ID (the identity of the claimant) is calculated by the hash function f. The result q and n are calculated by the hash function h to obtain r', r', which is sent to the verifier. The verifier calculates the saved q and the generated n through the hash function h to obtain r, and determines whether the identification is passed by comparing whether r and r' are equal.

Adding a query-response request in the process of kerberos V5 protocol authentication stage I authentication service exchange allows the authentication server AS to authenticate the user's identity before the user requests to obtain the ticket license ticket [13]. If the authentication is successful, it is proved to be a legitimate user, and the user can send a message requesting a ticket license ticket to the authentication server AS.

The following is a detailed description of the message authentication process using Challenge / Response mode in the process of stage I authentication service exchange. Before the message exchange, we assume that the user has been registered on the authentication server AS [14], that is, the user's password P_{sw} and the identity ID_c are stored in the AS's database. The hash value $q = f(P_{sw} | | ID_c)$ is generated by the user's password P_{sw} and the identity ID_c, and the user has his own secret password P_{sw}. When the user requests authentication from the authentication server, the user first logs on to the authentication interface. The process of improved authentication service exchange is as follows:

(1) C→AS:ID_c

(2) AS→C:n

(3) C→AS: $E_{Kc}[ID_c||T_c||R_c]||Options||ID_c||Realm_c||ID_{tgs}||Times||Nonce_1$

(4) AS→C: $Realm_c||ID_c||Ticket_{tgs}||E_{Kc}[K_{c,tgs}||Times||Nonce_1||Realm_{tgs}||ID_{tgs}]$

$Ticket_{tgs}=E_{Ktgs}[Flags||K_{c,tgs}||Realm_c||ID_c||AD_c||Times]$

First, the client makes an authentication request to the authentication server AS, and enters the user identity ID_c. After receiving the request, AS will find the user ID in its own database. If the same identity is found, AS will return a query message n to the user. This query message n is non-repetitive, and its value is completely determined by the authentication server, so that the authentication messages transmitted each time are different.

After receiving the query message, the user uses his secret password P_{sw} and identity and the query message n to obtain the value r' through a series of hash transformations, which can be used as the secret key K_c, that is, K_c = h [n | | f (P_{sw} | | ID_c)]. Then the user uses the secret key K_c to encrypt [ID_c | | T_c | | R_c] and sends it to the authentication server together with other items, where T_c and R_c are the timestamp and the generated random number of the client, respectively [15,16].

After receiving the message (3) sent by the client, AS finds the saved hash value f (P_{sw} | | ID_c) through the user identity ID_c, and calculates r with the query message n sent to the user through the hash function h. Then AS uses the calculated value r to decrypt the ciphertext E_{Kc} [ID_c | | T_c | | R_c]. If the ID_c obtained after decryption is consistent with the ID_c in message (3), and the timestamp T_c is the current valid time, then the authentication server AS can confirm the user's identity. After the user's identity is verified, the user can continue the subsequent protocol interaction process to obtain application services.

V. Security Analysis of the Improved Protocal

In this paper, the Challenge / Respons mechanism in OTP is used to improve the shortcomings of Kerberos protocol [17]. In the improved scheme, the user needs to request authentication from AS before applying for the ticket license ticket to the authentication server AS. The authentication protocol here is based on the query-response authentication technology. When the user requests authentication, it will send its own identity IDC to the authentication server AS. After receiving the request, AS sends a random query message n to the client, and then the user enters his own secret password P_{sw} [18]. The client will calculate the hash value r' based on the user password, identity and query message, and use this hash value as the user's secret key to encrypt the data [ID_c | | T_c | | R_c] and send it to AS. Since we add a random query message n when the client generates the user key, and use two different hash algorithms to generate the user key, this makes the user key generated each time different, that is, the plaintext-ciphertext pairs that appear in the message sent by the user to the AS are different, which makes the offline password guessing attack difficult [19]. And the timestamp T_c and the random number R_c in the ciphertext E_{Kc} [ID_c | | T_c | | R_c] will change every time, which also makes the offline password guessing attack quite difficult.

After the authentication server AS receives the message (3), the decryption key K_c is generated by the saved hash value f (P_{sw} | | ID_c) and the query message n. After decrypting the data block, AS compares the user identity obtained with the plaintext identity in the message, and checks whether the timestamp T_c is valid. If both are consistent, the authentication server can confirm the user's legal identity

VI. Conclusion

This paper describes the authentication process of Kerberos V5 protocol in detail, and analyzes the security of the authentication client. It is found that the authentication server AS does not authenticate the client and may lead to offline password guessing attack. The attacker can use this attack to obtain the key, and then impersonate the legitimate user to obtain the corresponding service. In view of this defect, this paper proposes a scheme to improve the Kerberos V5 protocol in the symmetric cryptosystem. This method uses the query message and the hash algorithm to make the user key different each time, which makes the offline password guessing attack relatively difficult. Use hash values and query messages to generate decryption keys to confirm the user's legal identity.

Although this paper improves the off-line password guessing attack of Kerberos V5, this improvement can only make the off-line password guessing attack relatively difficult, but it can not eliminate such attacks, and the limitations and other weaknesses of Kerberos V5, such as clock synchronization problem and key storage problem, have not been solved. In the future development, public key cryptosystem can be considered to further improve the security of the protocol.

References

[1] Steiner J G, Neuman B C, Schiller J I. Kerberos: An Authentication Service for Open Network Systems. Proc.of the Winter 1988 Usenix Conference, 1988.

[2] Li Zhongxian, Zhan Banghua, Yang Yixian. The development of authentication theory and technology. Journal of Electronics, 1999, 27 (1): 99-102.
[3] John T. Kohl, The Evolution of the Kerberos Authentication Service, EurOpen Conference Proceedings, May 1991.
[4] John T. Kohl,C. Neuman. The Kerberos Network Authentication Service (V5), RFC 1510, September 1993.
[5] Written by J. J. Storlins; translated by Meng Qingshu et al. Cryptography and Network Security: Principles and Practices: Fourth Edition. Beijing: Electronic Industry Press, 2006. 11.
[6] SHAO Yeqin, CHEN Xishann, GU Xiang.Improved KerberosSingle Sign-on Protoco.Computer Engineering, 2013, 37 (24): 109-111.
[7] Shenyang, Du Zhongjun. Research and Design of Single Sign-On Based on Kerberos Protocol. Computer Engineering and Design, 2011, 32 (7): 2249-2251.
[8] Liu Kelong, Qing Sihan, Meng Yang. A method for improving Kerberos protocol by using public key system. Journal of Software, 2001, (12): 874-877.
[9] Liu Jiayong, Ren Debin, Hu Yong, Fang Yong. Applied cryptography. Beijing: Tsinghua University Press, 2008. 9.
[10] Harold Weiss, (ed.). How passwords are cracked, The EDP Audit Control and Security Newsletter, 3(3), 1985.
[11] Cheng Xiao-rong, Feng Qi-yuan, Dong Chao, Zhang Ming-quan. Research and Realization of Authentication Technique Based on OTP and Kerberos. IEEE COMPUTER SOCIETY, 2005, 0-7695-2486-9/05.
[12] Haller N. “The S/KEY One-Time Password System”. http://www. faqs.org/rfcs/rfc1760. Html, 1995.
[13] Hasan Md Mehedi;Ariffin Noor Afiza Mohd;Sani Nor Fazlida Mohd, Efficient mutual authentication using Kerberos for resource constraint smart meter in advanced metering infrastructure, Journal of Intelligent Systems. Volume 32, Issue 1. 2023.
[14] Haggag Mohamed; Tantawy Mohsen M.; El-Soudani Magdy M.S, Token-based authentication for Hadoop platform, Ain Shams Engineering Journal. Volume 14, Issue 4. 2023.
[15] A. S. Anakath; S. Ambika; S. Rajakumar; R. Kannadasan; K. S. Sendhil Kumar, Fingerprint Agreement Using Enhanced Kerberos Authentication Protocol on M-Health, COMPUTER SYSTEMS SCIENCE AND ENGINEERING. Volume 43, Issue 2. 2022. PP 833-847.
[16] Jörg Schwenk; Douglas Stebila, A reduction-based proof for authentication and session key security in three-party Kerberos, International Journal of Applied Cryptography. Volume 4 , Issue 2. 2022. PP 61-84.
[17] Albaldawi Wafaa S.; Almuttairi Rafah M,Kerberos Authentication for Big Data Applications on Cloud Environment, Journal of Physics: Conference Series. Volume 1804, Issue 1. 2021. PP 012062.
[18] M.B. Benjula Anbu Malar; Prabhu J, Trust based authentication scheme (tbas) for cloud computing environment with Kerberos protocol using distributed controller and prevention attack, International Journal of Pervasive Computing and Communications. Volume 17, Issue 1. 2020. PP 78-88.
[19] Diaz Motero Carlos; Bermejo Higuera Juan Ramon; Bermejo Higuera Javier; Sicilia Montalvo Juan Antonio; Gamez Gomez Nadia, On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey, IEEE ACCESS. Volume 9, Issue. 2021.