Journal of Theoretical and Applied Information Technology
20th November 2014. Vol. 69 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCED KERBEROS AUTHENTICATION FOR DISTRIBUTED ENVIRONMENT

A. JESUDOSS #1, N.P. SUBRAMANIAM*2

[1] Research Scholar, Faculty of Computer Science and Engineering,
Sathyabama University, Chennai, Tamilnadu, India

[2] Asst. Professor, Department of EEE,
Pondicherry Engineering College, Puducherry, India

E-mail: # jesudossas@gmail.com, jesudoss.mca@sathyabamauniversity.ac.in

* npsubbu@yahoo.com

ABSTRACT

This paper aims to provide a unique and enhanced authentication model based on Kerberos environment. With this, it provides a hack-proof authentication system and protects the Kerberos environment from password-guessing attack and replay attack. Traditionally, the banking and financial institutions sends OTP to the client mobile. When it is hacked, the identity can be compromised. Using the proposed authentication model, even when the nonce value or the OTP is hacked, the identity cannot be compromised. This paper insists the need for an additional Session Key and a nonce to be used between the Authentication Server (AS) and Client i.e. Alice. This helps to make the security stronger. Kerberos environment is prone for replay attack and password-guessing attack and hence this security model helps Kerberos environment to prevent such attacks.

Keywords: *Authentication, Kerberos, KDC, Secret key, Session Key, Replay Attack, Password-Guessing Attack.*

1. INTRODUCTION

Many software applications are available in the market to secure our applications. The software that provide high-level of security leads to more execution time and usability of the software become very low. Security is like a double-edged sword because when security increases, usability and performance decreases. Kerberos is the authentication protocol that provides high-level of security and more usability. Kerberos is a network authentication protocol used for proving one's identity in an open network. Kerberos is based on security tokens or tickets. Tickets are used for enabling single sign-on feature and every ticket has expiration time to avoid replay attacks [1]. Kerberos provides mutual authentication which means both the client and server are mutually authenticated.

Kerberos is based on Symmetric-key Cryptography. KDC is used for sharing the secret key. Users must first register with KDC. The passwords chosen by the user at the time of registration is used to generate the secret key for the user and is stored in KDC. KDC is a centralized trusted third party which has repository of secret keys for providing security to applications in distributed network. It enables the users to trust one centralized KDC rather than trusting various workstations.

Kerberos provides single sign-on feature [2]. With single sign-on user needs to login only once to access different sites. There are two types of trusted intermediaries on the network. First is KDC (key distribution centre) that acts as a trusted intermediary to share the secret key between two parties and is used in symmetric-key cryptography. Another one is CA (Certification Authority) that issues certificates signed by his private key and is used in public-key cryptography. Kerberos protocol is most popular protocol used in various operating systems such as Red Hat, Windows Server. A Kerberos Realm is a collection of managed clients that uses the same Kerberos database. The realm usually consists of Kerberos server, registered clients and application servers that are registered with KDC and share the secret key. Kerberos works on the assumption that trusted hosts are being used on an open network or hostile environment. It provides mutual authentication between two parties [3]. It never sends the password across the network in any form [3]. It is used as an authentication mechanism in many operating systems such as Windows, Linux, UNIX, etc.

The Kerberos 4 had become obsolete as it uses DES encryption algorithm [4]. It is possible to impersonate and KDC can be compromised when the encryption fails. If the KDC can be compromised then the entire system can be compromised. KDC is the single point of contact for many clients and hence it affects the performance of the system. If the client is compromised, then it is possible to steal the password. The malicious user on the network can change the network address of the victim's data packet. Thus, it is vulnerable to replay attacks [5] and password-guessing attacks. Security Requirements satisfied by Kerberos are Authentication, Authorization, Confidentiality and Integrity.

2. LITERATURE SURVEY

The paper [5] shows how a replay attack can be used to steal user credentials. ARP spoofing attack can be used to impersonate the network address of the victim [5]. The Kerberos is built on the assumption that the encryption is effective and cannot be compromised [6]. The encryption is stronger or weaker depending upon the nature, type and complexity of the encryption algorithm chosen and the keys used for encryption. The impersonation attack is much more perilous than the confidentiality breach [7]. Kerberos 4 provides a vulnerability which allows the malicious user to impersonate [7]. The Kerberos 5 helps to rectify most of the flaws existed in Kerberos 4 [7]. But still Kerberos 5 suffers from many drawbacks such as impersonation, password-guessing attack, replay attack, etc. Kerberos protocol can be effectively used for IPv6 networks also [8]. The Trojan horse can be used to compromise many authentication mechanisms including Kerberos [9]. The user credentials must be dynamic so as to cheat the Trojan Horses. They must be different for each transaction [9].

The offline password-guessing attack is the most important vulnerability that exists in Kerberos [10]. Thomas Wu [10] explains the format of TGT Request Packet and TGT Return Packet. This data structure can be modified to avoid password-guessing attacks. It can include nonce and final nonce value to avoid such attacks. The paper [11] shows how Kerberos can be effectively used to authenticate using images. This makes the authentication to be user-friendly, easy to remember, hard to crack and so on [11]. The paper [12] shows a modified version of Kerberos that sends three passwords across the network. If weak passwords are chosen by the principal, then system is at risk. It is not advisable to send passwords across the network. It provides an opportunity for the hackers. The limitations of Kerberos [13] help to identify the limitations and weakness that exist in Kerberos and enable us to plan and design an innovative authentication model.

The PrivaKERB [14], a modified version of Kerberos shows how user anonymity helps in preventing an eavesdropper from ascertaining one's identity and user behavioural access patterns. The paper [15] shows how impersonation can be made in public key environment using PKINIT, a Kerberos public-key authentication mechanism.

3. MOTIVATION

Replay attack and password-guessing attack [16] is possible in Kerberos. It is possible to replay the userkey and obtain data from the Authentication Server. i.e. Message A and B sent by authentication server. Password-guessing attack may be performed on the data received from Authentication Server. Since Kerberos suffers from these drawbacks, to avoid these attacks, an enhanced security model should be deployed.

4. PROPOSED WORK

The proposed work is to design an authentication security model that prevents against password guessing attack and replay attack in Kerberos environment.

In traditional Kerberos, passwords are not sent across the network. Instead, user key, a hashed value, is sent to the AS for requesting a service. The user key is verified by the Authentication Server and if it is available on the database, then the Authentication Server sends Message A and B to the client. Message A (TGS Session Key) is encrypted with client's secret key and Message B (TGT) is encrypted with TGS secret key. Hence message B cannot be opened by the client.

The Man-In-The-Middle (MITM) may replay the user key and obtain message A and message B from the Authentication Server and may perform password-guessing attack. To avoid password-guessing attack, some secret must be shared between the Authentication Server and the Client. So when the Authentication Server is requested for a service, it can verify whether the request is from legitimate client or malicious client.

Hence, it is mandatory to share a session key between the AS and the Client. By sharing the session key between the AS and Client, password-guessing attack may be avoided. In existing Kerberos environment, there is no such session key between the AS and the Client and provides a room for password-guessing attack.

To avoid replay attack, the nonce can be used along with the user key. So the user key cannot be tampered so easily as it is combined with nonce value. Mutual authentication is achieved as the server knows the client by the nonce value and by the session key being used. The Client also identifies the server by its session key. Hence replay attack can be avoided.

Even though the session key is used between the Authentication Server and the Client, the data requested by the client may be replayed. The malicious user may not be able to decrypt or find the session key but he may use the entire data packet for replay attack. In the proposed work, a nonce has been used to avoid such replay attack

When the client sends user key encrypted with session key to access a service from the Authentication, the AS verifies whether the user key is available on the database and if available, it generates a random nonce value between 1 to 10 and sends the nonce encrypted with the session key. The client receives the packet and obtains the nonce value. The client must calculate the final nonce value based on the formula that is shared by the Client and the AS.

The Client calculates the final nonce value as given below. Let X be the nonce value. The secret values used in the formula be S1 and S2 and Y be the computed final nonce value. Therefore, the formula is given below:

$$X * S1 + S2 = Y,$$

where $y \geq (X + S2) \;\&\&\; \leq (S1 * 10) + 100$

Since this formula is shared by both the Client and the AS, the AS can verify the identity of the Client by its final nonce value. The Client has to once again request the AS for TGT by sending userkey along with final nonce value.

When the AS receives a request for TGT, it must verify whether the request has got final nonce value, if it is there then it can understand that the request is from the legitimate Client and not a replay attack. Because the nonce value sent by the AS is like OTP. After sending the final nonce value, it is marked by the AS against the userkey in the KDC database. After verifying successfully, the nonce will be removed from the database. So the final nonce value cannot be reused.

On receiving TGT request, the AS sends the nonce value encrypted with session key directly to the Client's IP address. So, the malicious user may not be aware of the nonce value. Suppose, if the malicious user initiates the TGT request and also captures the nonce sent by the AS, then he may not know what to do with nonce value received as he is not aware of the formula. Hence, replay attack is not possible. A Timer will be maintained by the AS for receiving the TGT request. Hence, the request for TGT must reach the AS before the timer expires. Similarly, after the first request for TGT with final nonce value, the nonce value maintained by the AS gets expired.

5. IMPLEMENTATION OF PROPOSED WORK

The proposed work has been implemented using Java. NetBeans 7.0.4 IDE has been used for developing the application. The client and server are configured on the same machine.

Traditionally, the banks and financial institutions sends only OTP to the client's mobile to verify the identity of the request. If the SIM card is hacked, the secret is revealed and the identity can be compromised. In our proposed work, a new and unique idea to use computed final nonce value has been implemented and hence it cannot be hacked and the identity cannot be compromised. Because when AS sends the nonce value, if it is hacked also. It will not be useful to the hacker. Whereas in the previous case, if the information sent by the bank is hacked, then the identity is lost.

Though, session key can be used to identify the identity of the Client, the final nonce value provides not only additional verification of the identity but also helps to avoid replay attack. The final nonce value must be in multiples of 5 within the range of 105 to 150. The TGT request must include the final nonce value that is sent from the Client to the AS.

6. RESULTS

The client application of the proposed authentication model generates the output as given below in Figure 2.

It shows how the client makes the initial request and on receiving OTP or nonce value, it sends TGT request. When the TGT request is successful, it receives TGT from the server.

The above figure 3 shows how the server is monitoring the client request and how it responds to the client. It sends OTP to the client on successful verification of the user and on successful OTP verification, it sends TGT to the client. This TGT can be used by the client to obtain the service ticket from the TGS server.

7. COMPARISON BETWEEN KERBEROS AND PROPOSED WORK

The Table given below shows the comparison between Kerberos and proposed work. The table clearly demonstrates the advantages of the proposed work over the traditional Kerberos.

8. CONCLUSION AND FUTURE WORK

The proposed work focuses mainly on replay attack and password-guessing attack. It can also prevent from keylogger attack and screenshot attack when the communication between the client and the server is implicit. It provides mutual authentication between the client and the server and thus, OTP cannot be trapped. This concept can be implemented in banks and financial institution where financial transactions are frequently made. Even though the malicious user possesses a duplicate SIM card, by using this concept, the OTP cannot be used to steal one's identity. The paper does not insist that Kerberos has many flaws. In fact, there is no doubt that Kerberos is the most popular and highly efficient network authentication protocol. The proposed work seeks to focus on how it can be made hack-proof. Like the proverb "a fruitful tree is often stoned" says, it aims to strengthen the Kerberos protocol and not to stone it.

Though the proposed work has many advantages, the replay attack is possible if the malicious user captures the first TGT request. The nonce value marked against the user database becomes invalid only after first TGT request. So if the first TGT request itself is replayed, it may not be possible to identify the malicious user. Our future work will focus on this limitation and enhance the authentication security model.

REFERENCES

[1] Giampaolo Bella, Elvinia Riccobene, "Formal Analysis of the Kerberos Authentication System", *Journal of Universal Computer Science*, vol. 3, 1997, pp. 1337-1381.

[2] Frederick Butler, Iliano Cervesato, Aaron D. Jaggard, Andre Scedrov, "A Formal Analysis of Some Properties of Kerberos 5 Using MSR", in proc. CSFW '02, 2002, p. 175.

[3] Asad Amir Pirzada, Chris McDonald "Kerberos assisted authentication in mobile ad-hoc networks", in proc. ACSC '04, 2004, vol. 26, p. 41-46.

[4] Alexandra Boldyreva and Virendra Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos", in proc. SP '07, 2007, pp. 92-100.

[5] Kimmo Kasslin, Antti Tikkanen and Teemupekka Virtanen, "Kerberos vs. Replay Attacks", in proc. AIWSC '03, 2003.

[6] Giampaolo Bella, Lawrence C. Paulson, "Kerberos Version IV: Inductive Analysis of the Secrecy Goals", *Lecture Notes in Computer Science*, vol. 1485, 1998, pp. 361-375,

[7] Tom Yu, Sam Hartman and Kenneth Raeburn, "The Perils of Unauthenticated Encryption: Kerberos Version 4", in proc. NDSS '04, 2004.

[8] Sakane, S., Okabe, N., Kamada, K., Esaki, H., "Applying Kerberos to the communication environment for information appliances", in proc. SAINTW '03, 2003, pp. 214-217.

[9] Neuman, B.C., Ts'o, T., "An authentication service for computer networks", IEEE Communications Magazine, vol. 32, issue 9, 1994. pp. 33-38.

[10] Thomas Wu, "A real-world analysis of Kerberos password security", in proc. 1999, NDSS '99.

[11] Nitin, Vivek Kumar Sehgal, Durg Singh Chauhan, "Image Based Authentication System with Sign-In Seal", in proc. 2008, WCECS '08.

[12] Gagan Dua, Nitin Gautam, Dharmendar Sharma, Ankit Arora, "Replay Attack Prevention In Kerberos Authentication Protocol Using Triple Password", International Journal of Computer Networks & Communications, vol. 5, no.2, March 2013, pp. 59-70

[13] Steven M. Bellovin, Michael Merritt, "Limitations of the KerberosAuthentication System", in proc. USENIX Conference '91, 1991, p. 253-267.

[14] F. Perenigueza, R. Marin-Lopeza, G. Kambourakisb, S. Gritzalisb, A. F. Gomeza, "A User Privacy Framework for Kerberos", Computers and Security, vol. 30, issue 6-7, pp. 446-463, Sept. 2011.

[15] Minkyu Kim (2009) prof. Raj Jain's Home Page Available: http://www.cse.wustl.edu/~jain/cse571-09/ftp/kerb5/index.html

[16] Eman El-Emam, Magdy Koutb, Hamdy Kelash, and Osama S. Faragallah, "An Authentication Protocol Based on Kerberos 5", International Journal of Network Security, vol.12, May 2011.

Table 1. Comparison of Kerberos and Proposed Work

Types of Attacks	Kerberos	Reason for loophole	Proposed Work	Justification
Replay Attack	Possible	Does not prove identity & Tokens reusable	Not Possible	Nonce Mechanism
Password-Guessing Attack	Possible	Initial handshake is not protected	Not Possible	Session Key
Keylogger Attack	Possible	Password is entered via keyboard	Not Possible	Password is not entered via keyboard
Screenshot Attack	Not possible	Internal Communication	Not Possible	Internal Communication

.

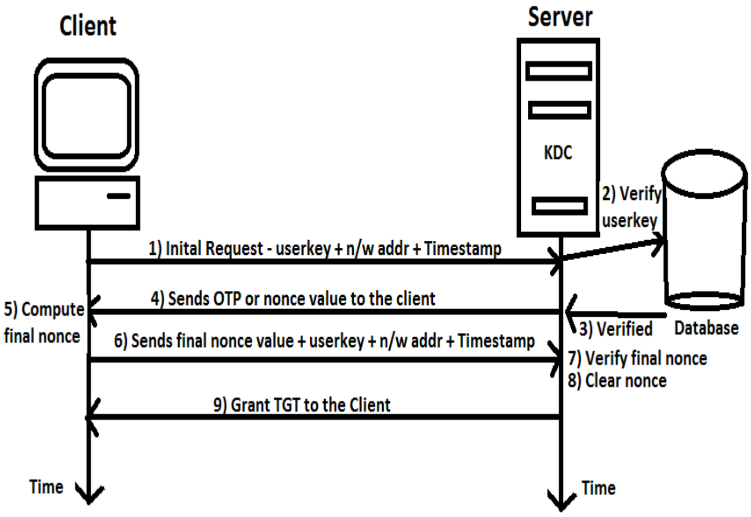
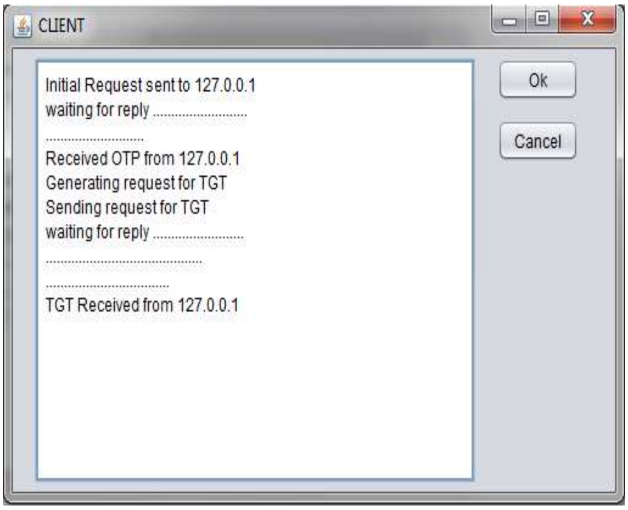
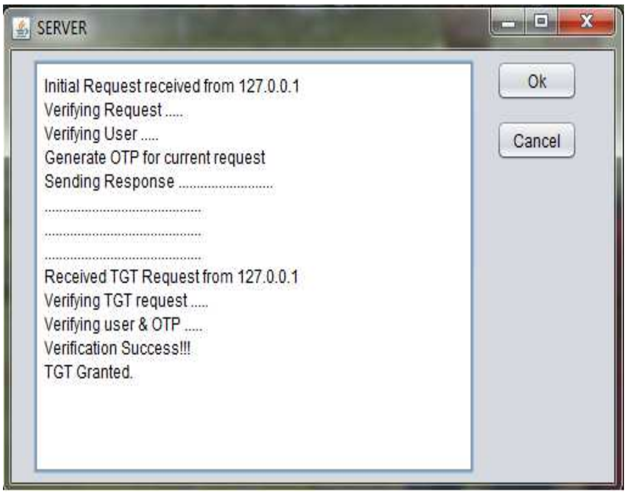

Client
Server
KDC
2) Verify userkey
Database
1) Inital Request - userkey + n/w addr + Timestamp
4) Sends OTP or nonce value to the client
3) Verified
5) Compute final nonce
6) Sends final nonce value + userkey + n/w addr + Timestamp
7) Verify final nonce
8) Clear nonce
9) Grant TGT to the Client
Time
Time

Figure 1. Enhanced Authentication Security Model



```
Initial Request sent to 127.0.0.1
waiting for reply ..........................
...........................
Received OTP from 127.0.0.1
Generating request for TGT
Sending request for TGT
waiting for reply .........................
..........................................
.................................
TGT Received from 127.0.0.1
```

Figure 2. Initial Request and TGT Request by the Client.



```
Initial Request received from 127.0.0.1
Verifying Request .....
Verifying User .....
Generate OTP for current request
Sending Response .........................
..........................................
..........................................
..........................................
Received TGT Request from 127.0.0.1
Verifying TGT request .....
Verifying user & OTP .....
Verification Success!!!
TGT Granted.
```

Figure 3. Authentication Server verifying and providing TGT.