Hash Based Dynamic Password Authentication Mechanism For Kerberos Environment

Vijendrasinh P. Thakur[1],
[1] M Tech Student,
Department of CSE, BCCE Nagpur
RTMNU Nagpur, India

Prof. K. N. Hande[2]
[2] Assistant Professor,
Department of CSE, BCCE Nagpur
RTMNU Nagpur, India

Abstract

`

Kerberos plays an important role in Authentication of Clients & Servers in a distributed system. Many works have analyzed its security, identifying flaws and often suggesting fixes, thus promoting the protocol's evolution. Several recent results present successful, formal methods-based verifications of a significant portion of the current version. Traditional kerberos is vulnerable to password guessing attack. To remove this weakness, there is a proposed scheme on dynamic password based authentication. In proposed protocol, use of AES Encryption & SHA1 algorithm for hash improves the security of kerberos environment.

1. Introduction

Nowadays, distributed systems are very popular and widely used throughout the globe. Most companies uses some flavor of distributed system to connect their various branches located in different geographic locations. In computer science, distributed Systems studies the coordinated use of physically distributed computers. While it is a desirable concept to use distributed systems widely, some concerns remain to be addressed. One of the main concerns that users face when using distributed system is 'security, authentication, integrity, confidentiality, and authorization'. Password-based authentication is not suitable for use on computer networks [14][15]. Eavesdroppers can intercept passwords sent across the network and use it to impersonate a legitimate user. Massachusetts Institute of Technology MIT began a project research named Athena, during which they analyze the way users authenticate themselves to a network and also what happen at other nodes in the network. The study reveals many flaws in the system and consequently, Kerberos was developed to answer a very complicated question-how do you authenticate a user to the server without worrying about impersonation and replay attack from attackers. The purpose of Kerberos was to provide secure authentication between client and server and to prevent interception by eavesdroppers; however, many studies reveals that Kerberos is not altogether one hundred percent immune against some security attacks.

The process of verifying the user's identity is called authentication. Authentication is a service related to identification. It is a fundamental building block for a secure networked environment. If a server knows the identity of a client, it can decide whether to provide the service, whether the user should be given special privileges, and so forth [2].

Roger Needham and Michael Schroeder of the Xerox Palo Alto Research Center published a paper in December of 1978 describing their framework for designing a secure network authentication system. The paper, entitled "Using Encryption for Authentication in Large Networks of Computers," described two different protocols that could be implemented to provide a reliable, secure authentication service for a distributed network of computers. The first protocol described in the paper uses private key encryption, and it is this protocol that forms the basis of the Kerberos network authentication protocol.[18]

The Kerberos model is based in part on Needham and Schroeder's trusted third-party authentication protocol and on modifications suggested by Denning and Sacco. Kerberos provides a means of verifying the identities of principals, (e.g., a workstation user or a network server) on an open (unprotected) network. This is accomplished without relying on authentication by the host operating system, without basing trust on host addresses, without requiring physical security of all the hosts on the network, and under the assumption that packets traveling along the network can be read, modified, and inserted at will. Kerberos performs authentication under these conditions as a trusted third party authentication service by using conventional cryptography, i.e., shared secret key.

The authentication process proceeds as follows: A client sends a request to the authentication server (AS) requesting "credentials" for a given server. The AS responds with these credentials, encrypted in the client's key.

The credentials consist of :

1) a "ticket" for the server and

2) a temporary encryption key (often called a "session key").

The client transmits the ticket (which contains the client's identity and a copy of the session key, all encrypted in the server's key) to the server.

It may also be used to encrypt further communication between the two parties or to exchange a separate sub-session key to be used to encrypt further communication. The implementation consists of one or more authentication servers running on physically secure hosts. The authentication servers maintain a database of principals (i.e., users and servers) and their secret keys. In order to add authentication to its transactions, a typical network application adds one or two calls to the Kerberos library, which results in the transmission of the necessary messages to achieve authentication.

There are two methods by which a client can ask a Kerberos server for credentials. Usually client requests for a ticket-granting ticket (TGT) which can be used with the ticket-granting server (TGS).

2. RELATED WORK

Kerberos is also introduced to be used in IPv6 networks. S. Sakane, N. Okabey, K. Kamadaz, and H. Esakix describe a method to establish secure communication using Kerberos in IPv6 networks [2].

Nitin et. al present an image based authentication system using the Kerberos protocol at 2008 [2][7]. That paper is a comprehensive study on the subject of using images as a password and the implementation of Jaypee University of Information Technology (JUIT) Image Based Authentication (IBA) system called as JUIT-IBA using Kerberos protocol.

In 2007, MIT formed the Kerberos Consortium along with some of the major vendors and users of Kerberos such as Sun Microsystems, Apple, Google, Microsoft, etc., to foster continued development. The MIT Kerberos Consortium was created to establish Kerberos as the universal authentication platform for the world's computer networks [2].

Dr. S. Santhosh Baboo, K. Gokulraj (2010), Authentication is one of the essential security features in network communication. Authentication process ascertains the legitimacy of the communicating partners in communication. The authors introduced a new authentication scheme based on dynamicity which is relatively a different approach to ensure and enhance the smart card based remote authentication and security. This method discusses about the authentication for smart card based network systems. This method introduces a dynamic authentication scheme which includes number of factors, among them the password, password index, and date of modification are important factors which decides the dynamicity [8].

K. Aruna et. al (2010), The aim of this paper is to establish a collaborative trust enhanced security model for distributed system in which a node either local or remote is trustworthy. They have also proposed a solution with trust policies as authorization semantics. Kerberos, a network authentication protocol is also used to ensure the security aspect when a client requests for certain services. In the proposed solution, they have also considered the issue of performance bottlenecks [8].

Dr.Mohammad N. Abdullah & May T. Abdul-Hadi (2009), they try to establish a secure communication between the clients and mobile-bank application server in which they can use their mobile phone to securely access their bank accounts, make and receive payments, and check their balances [8][9].

Kerberos has grown to become the most widely deployed system for authentication and authorization in modern computer networks. Kerberos is currently shipped with all major computer operating systems and is uniquely positioned to become a universal solution to the distributed authentication and authorization problem of communicating parties [2][3][4].

Wen Lei, Hai Cao, Xingjian Liang,Hong Zhang(2010)

Authors from School of Computer Science,Sichuan University of Science & Engineering,Zigong, Sichuan, China implemented a dynamic password & one time password based modified kerberos protocol.Their work is as below.

By studying the Kerberos authentication scheme, an improved authentication scheme is raised, which is based on Dynamic Password Method. In the improved scheme, user's password can be effectively protected, and the authentication is double between users and severs. Also, the scheme can resist jacking connection attack. The improved scheme is more secure and more practical than the original one.[25]

3. KERBEROS

3.1 Kerberos Message Exchange

In an unprotected network environment, any client can apply to any server for service. The obvious security risk is that of impersonation. An opponent can pretend to be another client and obtain unauthorized privileges on server machines. To counter this threat, servers must be able to confirm the identities of clients who request service. Each server can be required to undertake this task for each client/server interaction, but in an open environment, this places a substantial burden on each server.

The Kerberos Key Distribution Center, or KDC for short, is an integral part of the Kerberos system. The KDC consists of three logical components: a database of all principals and their associated encryption keys, the Authentication Server, and the Ticket Granting Server. While each of these components are logically separate, they are usually implemented in a single program and run together in a single process space.

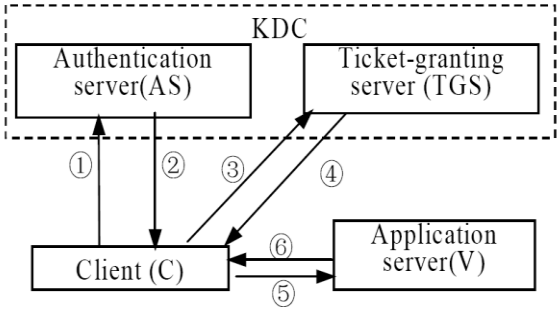


Fig 1: The process of Kerberos authentication

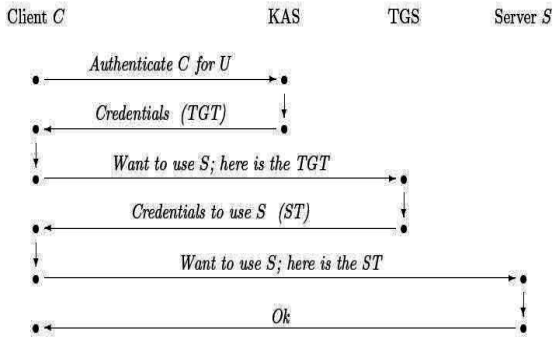


Fig 2: Overview of the Kerberos Authentication

The client's goal is to be able to authenticate him to various application servers (e.g., email, file, print, web servers). This is done by first obtaining credentials, called the "ticket-granting ticket" (TGT), from a "Kerberos Authentication Server" (KAS) and then by presenting these credentials to a "Ticket-Granting Server" (TGS) in order to obtain a "service ticket" (ST), which is the credentials that the client finally presents to the application servers in order to authenticate himself. A TGT might be valid for a day, and may be used to obtain several STs for many different application servers from the TGS, while a single ST might be valid for a few minutes (although it, too, may be used repeatedly while it is still valid) and is used for a single application server. The KAS and the TGS are altogether known as the "Key Distribution Center" (KDC). The client's interactions with the KAS, TGS, and application servers are called the Authentication Service (AS), Ticket-Granting (TG), and Client-Server (CS) exchanges, respectively.

3.2 Kerberos Drawbacks

i. "Password guessing" attacks: Password guessing attacks are not solved by Kerberos. If a user chooses a poor password, it is possible for an attacker to successfully mount an offline dictionary attack by repeatedly attempting to decrypt messages obtained which are encrypted under a key derived from the user's password.

The aim is on designing a user authentication protocol that is not susceptible to password guessing attacks. The main goal is to remove this password guessing attack [1][2][5][6][14].

ii. Principals must keep their secret keys secret. If an intruder somehow steals a principal's key, it will be able to masquerade as that principal or impersonate any server to the legitimate principal [5][6][14].

iii. KDC spoofing: This refers to an attack which relies basically on the ability to spoof KDC responses. Having in mind the Kerberos protocol description, spoofing KDC response should not be a security concern. Indeed, Kerberos has been design to bear an untrusted network. IP spoofing is something that happens on untrusted networks. Kerberos protocol performs mutual authentication. End user's and server's identities need to be proven. This ensures protection against Man-in-the-Middle attacks. Yet circumstances still exit under which this might represent a real risk [14].

iv. "Denial of service" attacks: Denial of service attacks are not solved with Kerberos. There are places in these protocols where an intruder can prevent an application from participating in the proper authentication steps. Detection and solution of such attacks (some of which can appear to be not-uncommon "normal" failure modes for the system) is usually best left to the human administrators and users.

An attacker can mount a DoS attack by flooding the KDC with authentication requests, which may result in poor response time to legitimate requests and in worst cases might even crash the KDC. To prevent a denial of service attacks, one solution could be to place the KDC behind a firewall and/or place redundant KDC slaves to service the requests and balance load [14][15].

v. Compromise of the KDC Server: KDCs maintain an encrypted database of all principals/verifiers (i.e., users and servers) and their secret keys. If the security of the KDC is compromised, the security of the entire network is compromised even though the principal keys are stored in an encrypted form using the master key; the master key itself is stored in the KDC. An attacker can gain control of the entire network, can create or modify any principal's credentials. To prevent such attack, ensure the security of the KDC and

limited the access to KDC to very limited personnel. Even then, it does not protect against an insider attack, which is to an attack by an internal user who misuses his/her privileges i.e. an administrator.

vi. Compromise of a verifier/server: If the security of the server is compromised, all the services on that server is compromised. The attacker will be able to impersonate any service running on the server and/or decrypt any communication between the service and a client/principal. The security of the services running on a server is dependent upon the security of the server. Security measures of servers should be proportional to value of the services and resources stored on that server.

4. 4 PROPOSED RESEARCH WORK

4.1 Proposed Authentication Architecture

Firstly Client or user has to authenticate itself to the authentication server. We will add a mechanism by which dynamicity in passwords can be added according to figure 3 & 4.

Dynamic Password (DP) is also called One Time Password (OTP), which is used to solve the traditional problems which appear when the static Password authentication cannot cope with eavesdropping and replaying, forging, guessing, etc. By means of DP, uncertainties will be considered in authentication information during the process of lodging to make authentication information different every time, which can improve the security of information in the process of lodging. This technology can effectively avoid replay attack, and solve the problems that the static password is likely to be stolen in transmission and database.[25]

So the focus of the research is on providing one time password OTP as well as Dynamic password based Authentication for kerberos environment.

There are three fields which are sent to authentication server or KDC. Those are Principal ID, Principal Password & current system timestamp of client's machine. Principal password & timestamp are hashed first & then sent.

Client Side Authentication:

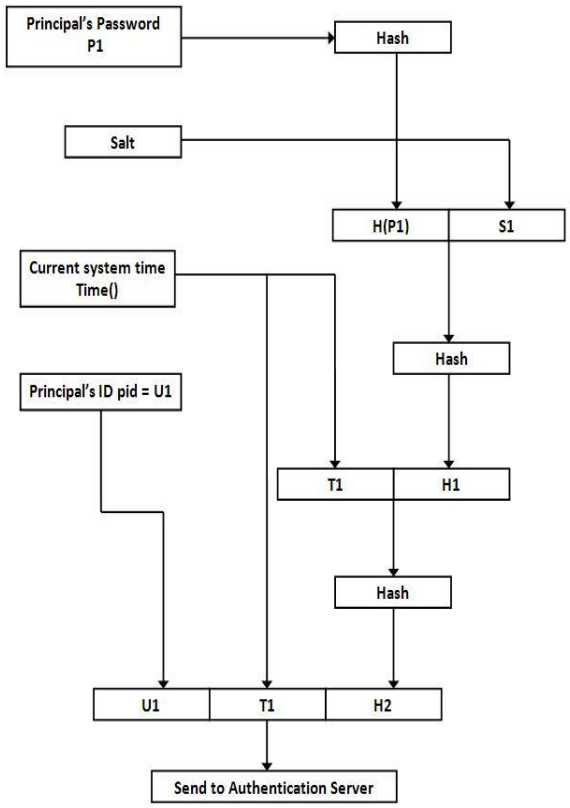


Fig 3 Design of New Authentication Client Side

Server Side Authentication:

Now at server side, server checks to see that user is the right one or not who it pretends to be. Server has its database of valid Principal ID & Principal Password pairs(fig 4) . Server firstly checks for replay attack by comparing the timestamps. Then server checks to see a right password is supplied or not by comparing hash values of received & server generated values. Now next step is generation of secret key used to encrypt the ticket. It is shown in fig 5. Principal's profile is appended with system timestamp & it is first hashed with SHA1 hashing algorithm & then encrypted by AES algorithm to generate a secret key to encrypt a ticket.

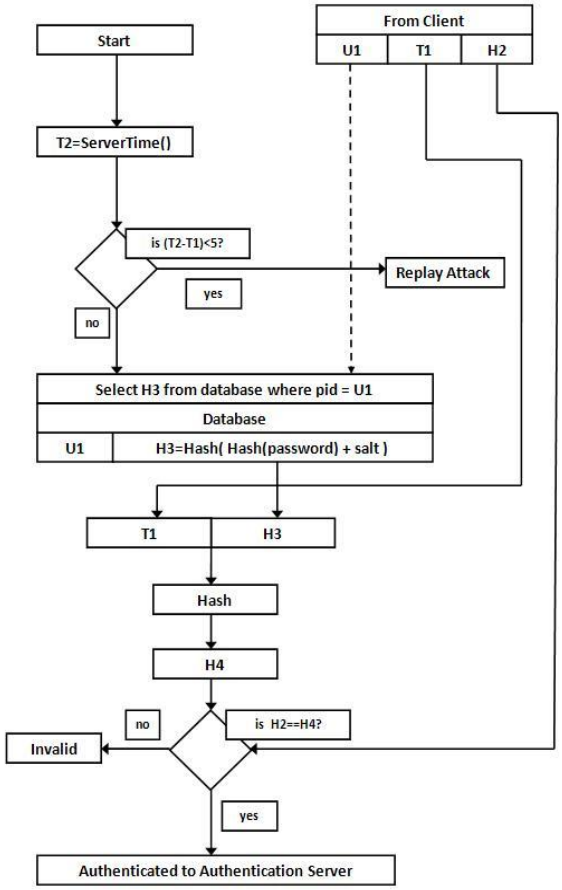


Fig 4 Design of New Authentication Server Side

4.2 Proposed Secret key generation

The proposed method to improve Kerberos towards password guessing attack as the use of strong hashing algorithm on user's profile. It is obvious that Kerberos is vulnerable to password guessing attacks. There is one suggestion on an authentication protocol based on Kerberos with a little modification in the Kerberos database. It will be independent of the user password. Instead, the KDC will save a profile for every principal in the realm that it manages. The contents of the profile may be audio, video, image, or text data. The KDC database may have profiles of mixed data contents (some profiles may be audio, others may be images, and so on). The realm principal may be a client or a server instance that participates in the network communication. Every principle (user or server) has to register with the Kerberos database. The principal will register with the Kerberos server by the principal ID. Then, the KDC will map this ID to the principal profile. The Kerberos server will generate the principal secret key by applying a hashing algorithm to the principal profile. The input to the hashing algorithm will be the principal profile and the output will be encrypted by AES Algorithm [31][32] to generate the principal secret key. This secret key is then used to encrypt the tickets for accessing ticket granting server TGS & Server V The block diagram of Fig. 5 summarizes proposed scheme to generate the principle secret key. It is also suggested to control the lifetime of that secret key. There is a simple idea for that. Since the system clocks of the hosts that are involved in the protocol should be synchronized (this can be maintained manually or assured by using Network Time Protocol daemons), it will append the current system timestamp to the principal profile every certain predefined period (this period is a design parameter; i.e. a site constant). Consequently, the input to the hashing algorithm will change, and thus the secret key will change too [2][3][4].

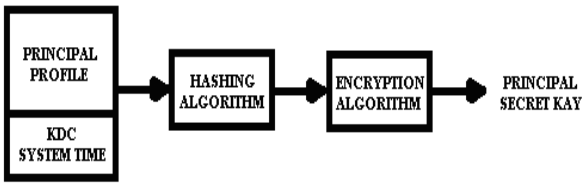


Fig. 5 Proposed Secret key generation Architecture

4.3 Ticket Validation

Similar to the original scheme, there is such a hypothesis: KDC user database is absolutely safe, however, once the database is broken through, all user information might be exposed, it will result in certification system failure. To solve this problem, a token generating algorithm (TGA) may be considered, each time a user is authenticated successfully, the algorithm computes a token and sends it to the user for the next request, but the server doesn't keep this token. When the user asks for authentication again, the KDC does not only to verify his personal information, but also verify his token by calculating; its process is shown in Fig 6. Thus, even an attacker obtains the user information, he cannot pass through the authentication.

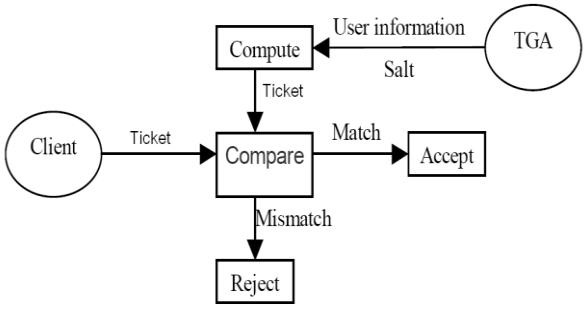


Fig 6. The authentication principle based on token

5 Security analysis & strengths of proposed protocol:

5.1 Strengths:

Table 1. Comparison of proposed system with existing system

Operation	Counts	
	The original	The improved
DES Encryption Algorithm	2	0
AES Encryption Algorithm	0	2
SHA1 Hash Algorithm	0	4
AES Encryption Algorithm for Secret Key Generation	0	1
SHA1 Algorithm for Secret Key Generation	0	1

Password Salt: Use of password salting technique such that password of every principal is stored by respective hash value of passwords. Because hashing is one way function getting the password back from hash value is impossible.

So if security of kerberos database is compromised it does not matter a lot because intruder or attacker cannot do anything with the database.

Dynamicity in Password: As we have added timestamp in password before hashing, hash value will change according to change of time. So as time changes the key used will also change.

It will also avoid reuse of password. So the frequent key renewal under secure condition is achieved.

Removal of weak algorithms: Removal of weak algorithms for encryption as directed by RFC 6649 & notice from National Institute of Standards and Technology(NIST) for Federal Register / Vol. 70, No. 96 / Thursday, May 19, 2005 / Notices [15][21]

Use of strong cryptographic algorithms:

Use of NIST(National Institute of Standards & Technology,)'s Federal Information Processing Standard (FIPS) 197 i.e AES Advanced Encryption Standard for encryption.

Use of improved hashing algorithm SHA1 & dual hashing is preferred in the proposed scheme.

5.2 Security analysis / Attacks Prevented:

5.2.1 Password Guessing Attack, Brute Force Attack, Dictionary Attack:

When U1, T1, and H2 were sent to the server, the server would verify T1. If T1 (Client's Time) differed from the server's time more than 5 seconds, it could be concluded that U1+T1+H2 were captured and undertook the Replay Attack. In that case, the browser would be redirected to the login page. If T1 and the server's time differed not over 5 seconds, the server would verify the user's password via SQL command order to verify the user's password in database (DB) under the username's condition that matched with U1.

The security of this model was that the password field (H2) could be changeable because each time the user logged in, the client's time would change so that the Hash (P1+T1) also changed along. In case that the client was sending U1+T1+H2 to the server and the hacker wanted to detect U1+T1+H2, he would make ARP Spoof on the LAN, as well as run the Ethereal or WireShark to detect the data. After that, it would take several minutes to receive the data, stopped detecting, and started searching for the password. This process would take quite a long period to be completed. In general, it would take approximately 1-5 minutes (60-300 seconds). It surely would not be completed within 10 seconds (by manual processing). After the hacker received U1+T1+H2 and undertook the Manual Replay Attack, it still could not attack the system since the model provided the security system against the attack by verifying the difference between the server's time and T1 (the client's time). If the time differed more than 5 seconds, it was admitted to be the replay attack. So brute force attack, Dictionary attack is not possible since it requires time to make attack.

5.2.2 Sniffing Attack:

Network sniffer can silently observing network traffic.In such a case if an attacker captures the ticket & uses it on his own machine then such type of attack is prevented by our system because at ticket validation we are checking for correct ip address of client & server. A ticket generated for client alice & for client ip 172.16.55.156 can not be used from any other mahine having different ip address.

5.2.3 Insider Attack : ip spoofing attack:

Suppose there are two legitimate users alice & bob. If alice has got his ticket but somehow can not keep it secret. If anyhow bob gets ticket intended for alice & uses it then system will not give service to such attack. I have checked client id in ticket for that.

Ticket generated for client having id alice can not be used by another user which is legitimate but having different id than in the ticket. This is type of masquerade attack,

5.2.4 Ticket Tampering:

If ticket is tampered for unauthorized use of it, then system detects the attack while ticket validation process at TGS Server.

5.2.5 Replay Attack , Timing Attack:

In the proposed authentication model (fig 3 &4) if difference between the timestamps T2 & T1 exceeds a certain delay configured e.g. 5 ms, then a replay attack is detected & server does not replies to such requests as they are replay or timing attacked packets.

5.2.5 Session Hijacking Attack: Man in the middle Attack:

Fig 7. Session Hijacking Example

An attacker seating at different location hijacks the session between two hosts located at Chicago & London. Two hosts A & B cannot predict the presence of attacker who is listening their communication in passive attack. In active attack Attacker changes the contents of messages exchanged between host A & B. It can be more dangerous if the communication is private. Such types of attacks are prevented in secure authentication to kerberos environment.

5.2.6 Eaves Dropping and Passive Monitoring:

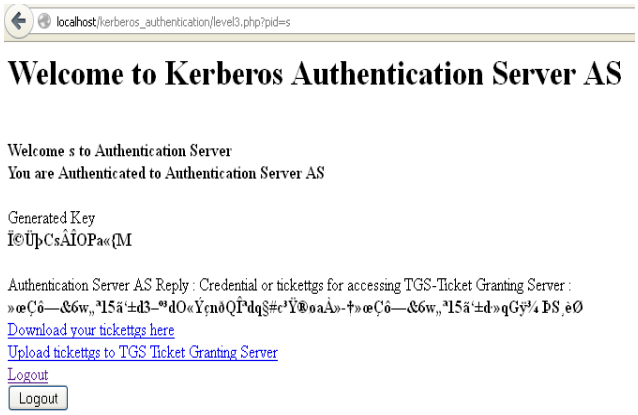
The intercepting and reading of messages and information by unintended recipients. If the messages are not protected by cryptographic mechanisms, then definitely the adversary can capture the message.

Countermeasure by our proposed protocol:

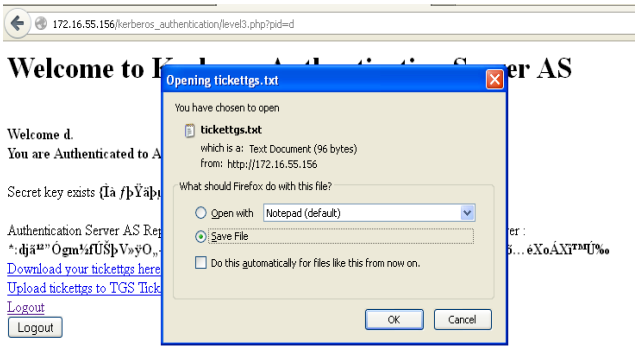
Every message that transmitted between AS, TGS, Application Server and Cilent is encrypted formats, Hence no scope for Eaves dropping and passive monitoring.

6 EXPERIMENTAL RESULTS

Screenshots

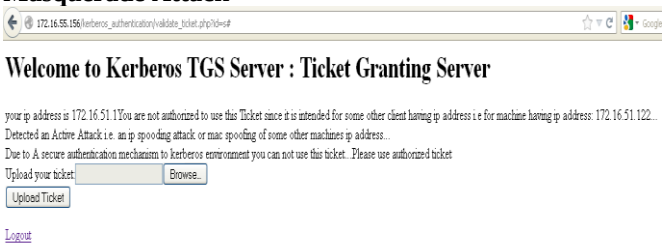


Snapshot 1 Authentication Phase

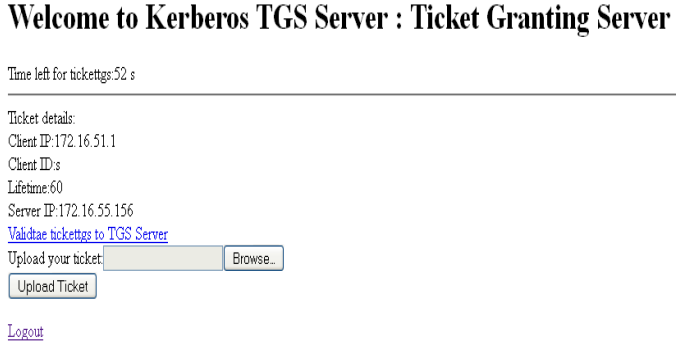


Snapshot 2 Generation of Encrypted Ticket tickettgs to access TGS Ticket Granting Server

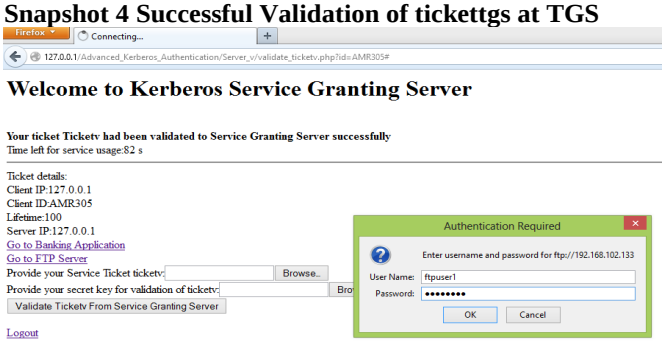
Masquerade Attack



Snapshot 3 Validation phase of tickettgs at TGS, ip Spoofing Attack of client.



Snapshot 4 Successful Validation of tickettgs at TGS



Snapshot 5 Access of FTP service.

7 CONCLUSION

Our scheme is based on the reliable Kerberos protocol. We have introduced certain changes to the original protocol for improved security against password guessing attack; the main drawback of kerberos. Frequent key renewal under secure condition provides dynamic passwords. This scheme based on dynamicity in password is more secure & not vulnerable to various types of attacks as described. We have used an AES standard for improving security as stated in NIST,s FIPS Standard no 197. Also hashing algorithm SHA1 is used for making one way hashes. Used an advantage of one way function as it is an irreversible process & attacker cannot get back the password from hash. Every ticket is encrypted by a secret key which is dynamic in nature. At validation phase on servers we have applied detection mechanism of different attacks. So Proposed scheme is not vulnerable to different network attacks.

REFERENCES

[1] Emir Accilien CMPT 585001 "Security issues in Distributed Systems:Is Kerberos the Answer?"

[2] Eman El-Emam, Magdy Koutb, Hamdy Kelash, and Osama Farag Allah, "An Authentication Protocol Based on Kerberos 5", IJNS International Journal of Network Security, Vol.12, No.3, PP.159-170, May 2011.

[3] Vijendrasinh Thakur, Prof. K.N.Hande, "Improving Kerberos Security by using a dynamic Password Based Authentication", International Journal of Emerging trends in Engineering and Development Issue 2, Vol.7 (November 2012), Available online : **http://rspublication.com/ijeted/nov12/31.pdf**

[4] Eman El-Emam, Magdy Koutb, Hamdy Kelash , and Osama Farag Allah,"A Network Authentication Protocol Based on Kerberos", IJCSNS International Journal of Computer Science and Network Security, Vol. 9 (8), August 2009.

[5] El-Emam, "An optimized Kerberos authentication protocol", ICCES 2009.International Conference on Computer Engineering & Systems, pp. 508-523, 14-16 Dec. 2009.

[6] [RFC1510] Kohl, J., Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, Sept 1993.

[7] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos network authentication service (V5)". Network Working Group. Request for Comments: 4120. Available at http: //www.ietf.org/rfc/rfc4120.txt, 2005.

[8] Nitin et al. "Security Analysis and Implementation of JUIT—Image Based Authentication System Using Kerberos Protocol". Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science, (ICIS 2008).

[9] Prof R.P. Arora, Garima Verma, "Implementation of highly efficient Authentication and Transaction Security", International Journal of Computer Applications (0975 – 8887) Volume 21– No.3, May 2011.

[10] Dr. Mohammad N. Abdullah, May T. Abdul-Hadi, "A Secure Mobile Banking Using Kerberos Protocol", Eng. & Tech. Journal, Vol. 27, N0. 6, 2009.

[11] William Stallings, "Cryptography and Network Security principles and practices", Fourth edition. Pearson Prentice Hall, (2006). pp. 401-419, pp. 433-435.

[12] A. Boldyreva and V. Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos". IEEE Symposium on Security and Privacy (SP'07). May 2007.

[13] Shital S. Thorat, Prof. H. K. Sawant, Mrs. Sarita S. Gaikwad, Prof. G. T. Chavan, "Comparative study of various PKINIT methods used in Advanced Kerberos", IJCSE International Journal of Computer Science and Engineering, Vol. 2 (7), 2010.

[14] L. Hornquist Astrand, "Deprecate DES, RC4-HMAC-EXP, and other weak cryptographic algorithms in Kerberos" February 27, 2012, Accessed July 31, 2012. Available Online URL: http://tools.ietf.org/pdf/draft-ietf-krb-wg-des-die-die-die-04.pdf

[15] Thomas Wu, "A Real-World Analysis of Kerberos Password Security", Available Online: www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/wu.pdf Accessed: July 23, 2012.

[16] Alexandra Boldyreva, Virendra Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos", IEEE Symposium on Security & Privacy Proceedings, pp. 92–100, 2007 & in IET Information Security Journal, Vol. 5(4), pp. 207–219, 2011.

[17] Rafael Marín-Lópe, Fernando Pereñíguez, Gabriel López, Alejandro Pérez-Méndez, "Providing EAP-based Kerberos pre-authentication and advanced authorization for network federations", Computer Standards & Interfaces Volume 33, Issue 5, September 2011, Pages 494–504 ScienceDirect.com, Accepted 22 February 2011.

[18] Jason Garman , "Kerberos the Definitive Guide", Publisher : O'Reilly, Pub Date : August 2003, ISBN : 0-596-00403-6, Pages : 272. Accessed: October 9 2012.

[19] Needham, R., and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Communications of the ACM, Vol. 21 (12), pp. 993-999, December 1978.

[20] Wen Lei, Hai Cao, Xingjian Liang,Hong Zhang, "An Improved Kerberos Scheme Based on Dynamic Password", Available Online: December 2010 in MECS (http://www.mecs-press.org/) I.J. Information Technology and Computer Science, 2010, 2, 33-39

[21] Shital S. Thorat, Prof. H. K. Sawant, Mrs. Sarita S. Gaikwad, Prof. G. T. Chavan, "Comparative study of various PKINIT methods used in Advanced Kerberos", IJCSE International Journal of Computer Science and Engineering, Vol. 2 (7), 2010.