International Journal For Technological Research In Engineering
Volume 4, Issue 10, June-2017 *ISSN (Online): 2347 - 4718*

CLOUD SECURITY USING KERBEROS PROTOCOL

Shashikumar[1], Prasanna B T[2]
[1]M.Tech 2nd Semester, [2]Assistant Professor
Department of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering(SJCE),
JSS S&T University Campus, Mysore, Karnataka, India

Abstract: Cloud computing is an era it starts after the development of parallel computing, grid computing and distributed computing. Organizations have started to use cloud computing store their data and provide security to their data. In this paper we discuss about Kerberos working principles and security issues.
KeyWords: Secreatekey, authentication, KDS, TGT, TGS.

I. INTRODUCTION

Kerberos is a commonly used authentication service on the Internet. Developed at the MIT's Project Athena, Kerberos is named for the three-headed dog who, according to Greek mythology, guards the entrance of Hades (rather than the exit, for some reason!). Kerberos employs client/server architecture and provides user-to-server authentication rather than host-to-host authentication. Kerberos model, security and authentication based on secret key sharing technology. In cloud every host on the network has its own secret key. It would clearly be unmanageable if every host had to know the keys of all other hosts so a secure,trusted host somewhere on the network, known as a Key Distribution Center (KDC), knows the keys for all of the hosts (or at least some of the hosts within a portion of the network, called a realm). In this way, when a new node is brought online, only the KDC and the new node need to be configured with the node's key. Kerberos fixes these problems because it provides single-sign-on, which lets a user log in to a system and access multiple systems or applications without the need to enter the user name and password multiple times. In addition, Kerberos is designed so that entities have to authenticate themselves by demonstrating control of secret information. In this manner, Kerberos solves traditional problems involved with authentication.

II. KERBEROS IN CLOUD

For getting the proficient cloud computing based services over internet services it can provide a swift and decidedly proficient system with least resource administration activities and minimum interface of service providers. Most of current applications require the client to memorize and utilize different set of credentials (e.g. client name/password or tokens) for each application he/she wants to access. However, this approach is inefficient and insecure with the exponential growth in the number of applications and services a client has to access both inside corporative environments and at the internet. Mainly, it is difficult for a corporation to manage potentially multiple authentication solutions and databases individually used by each application.

Furthermore, most clients tend to rely on the same set of credentials for accessing all of their systems, posing a serious security threat since an attacker who discovers these credentials can easily access all of the client's applications

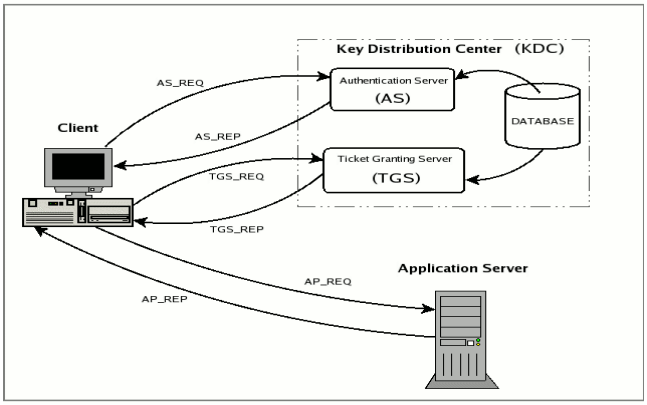

Fig 1:Kerberos

Key Distribution Center (KDC):-KDC is the heart of the Kerberos realm. It provides Kerberos authentication services by issuing encrypted tickets that require secret keys to decode. KDC handles the distribution of keys and tickets.

Ticket Granting Server (TGS):- TGS issues service tickets to clients upon request.

Ticket Granting Ticket(TGT):- Issued by the Authentication Server (A.S.), the TGT is encrypted in the user's password which is known only to the user and KDC.

Session Key:- Session keys are temporary private keys generated by Kerberos. They are known to the client and are used to encrypt the communication between the client and the server.

III. ACCESS CONTROL

In a network of users requiring services rom many separate computers, there are three approaches one can take to access control: [1]One can do nothing, relying on the machine to which the user is logged in to prevent unauthorized access; [2] one can require the host to prove its identity, but trust the host's word as to who the user is; or [3] one can require the user to prove her/his identity for each required service. In a closed environment where all the machines are under strict control, one can use the first approach. When the organization controls all the hosts communicating over the network, this is a reasonable approach. In a more open environment, one might selectively trust only those hosts under organizational control. In this case, each host must be required to prove its identity. The rlogin and rsh programs use this approach. In those protocols, authentication is done by checking the Internet address from which a connection

has been established. In the Athena environment, we must be able to honor requests from hosts that are not under organizational control. Users have complete control of their workstations: they can reboot them, bring them up standalone, or even boot off their own tapes. As such, the third approach must be taken; the user must prove her/his identity for each desired service. The server must also prove its identity. It is not sufficient to physically secure the host running a network server; someone elsewhere on the network may be masquerading as the given server Requirements on an identification mechanism.

1. First, it must be secure.
2. Second, it must be reli-able. Access to many services will depend on the authentication service.

Kerberos provides three distinct levels of protection. The application programmer determines which is appropriate, according to the requirements of the application. For example, some applications require only that authenticity be established at the initiation of a network connection, and can assume that further messages from a given network address originate from the authenticated party. Our authenticated network file system uses this level of security. Other applications require authentication of each message, but do not care whether the content of the message is disclosed or not. For these, Kerberos provides safe messages. Yet a higher level of security is provided by private messages, where each message is not only authenticated, but also encrypted. Private messages are used, for example, by the Kerberos server

Database Replication

Each Kerberos realm has a master Kerberos machine, which houses the master copy of the authentication database. It is possible (although not necessary) to have additional, read-only copies of the database on slave machines elsewhere in the system.

The advantages of having multiple copies of the database are those usually cited for replication:

higher availability and better performance. If the master machine is down, authentication can still be achieved on one of the slave machines. The ability to perform authentication on any one of several machines reduces the probability of a bottleneck at the master machine. Kerberos From the Outside Looking In Keeping multiple copies of the database introduces the problem of data consistency. We have found that very simple methods suffice for dealing with inconsistency. The master database is dumped every hour. The database is sent, in its entirety, to the slave machines, which then update their own databases. A program on the master host, called kprop, sends the update to a peer program, called kpropd, running on each of the slave machines (see Figure 13). First kprop sends a checksum of the new database it is about to send. The checksum is encrypted in the Kerberos master database key, which both the master and slave Kerberos machines possess. The data is then transferred over the network to the kpropd on the slave machine. The slave propagation server calculates a checksum of the data it has received, and if it matches the checksum sent by the master, the new information is used to update the slave's database.

IV. GOALS AND LIMITATIONS OF KERBEROS

Here we have some goals,limitations and drawbacks of kerberos which are listed below

Goals:

Starting with those assumptions, the system's key goals can be summarized as follows:

1. Never transmit unencrypted passwords over the network, i.e. "in the clear".
2. Protect against the misuse of intercepted credentials (also called "replay attacks").
3. Do not require the user to repeatedly enter a password to access routine services.

Limitations:

1. Since the system relies entirely on passwords for user authentication, if the passwords themselves are stolen, the possibility of system attack is unlimited. These points to the requirement that the Key Distribution Center be protected. If it is compromised, the entire system is unsafe .
2. For long processes, limited duration tickets can present problems. Kerberos Version 5 addresses this problem with renewable tickets.

Kerberos Drawbacks

The protocol weaknesses can be summarized as follows:

1. Kerberos requires continuous availability of the KDC. When the Kerberos server is down, the system will be vulnerable to the single point of failure problem. This can be mitigated by using multiple Kerberos servers.
2. The system clocks of the hosts that are involved in the protocol should be synchronized. The tickets have a time availability period and if the host clock is not synchronized with the Kerberos server clock, the authentication will fail. In practice, Network Time Protocol daemons are usually used to keep the host clocks synchronized.
3. "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.
4. There are no standards for the administration of the Kerberos protocol. This will differ between server implementations.

V. CONCLUSION

This paper provides the details of kerberos security in cloud. By using kerberos protocol we can make secure communication between cloud server and client. Kerberos protocol is the one of the standard protocol now a day we are using. Kerberos will provide secure three way hand shanking authentication mechanism by this any organization can use this protocol for security purpose.

REFERENCE

[1] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos Authentication System," Proc. USENIX, Winter 1991.
[2] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," IEEE Communications, September 1994, pp. 33-38.
[3] J. Kohl, C. Neuman,RFC1510: The Kerberos

Network Authentication Ser-vice (V5) , Internet Engineering Task Force, September 1993

[4] J. Kojl, C. Neuman, T. Ts'o, The Evolution of the Kerberos Authentication Service , 1991

[5] B. Bryant, Designing an Authentication System: a Dialogue in Four Scenes , 1988.

[6] Kerberos V5 Administrators Guide, Michigin Institute of Technology, 2002

[7] Windows 2000 Kerberos Authentication, Microsoft Corporation, 1999

[8] A. Niemi, J. Arkko, V. Torvinen RFC3310: Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement(AKA), Internet Engineering Task Force, September 2002.

[9] B. Tung, C. Neuman, J. Wray, A. Medvinsky, M. Hur, and J. Trostle, Public Key Cryptography for Initial Authentication in Kerberos, Internet Engineering Task Force draft, May 2005

