SECURE USER AUTHENTICATION IN CLOUD COMPUTING USING KERBEROS

R.Vijayakumari

Asst. Professor, Dept. of Computer Science, Krishna University, Machilipatnam

***Abstract*— Cloud Computing may be considered as the next logical step in resource outsourcing, but security is recognized as the main stumbling block for wider cloud adoption. The prominence of the place of cloud computing in future converged networks is incontestable. This is due to the obvious advantages of the cloud as a medium of storage with ubiquity of access platforms and minimal hardware requirements on the user end. Secure delivery of data to and from the cloud is however a serious issue that needs to be addressed. In this paper providing user authentication to cloud servers using kerberos is discussed.**

I. INTRODUCTION

Cloud computing as a concept is the result of the natural evolution of our everyday approach to using technology delivered via the Internet. Cloud computing came into the foreground as a result of advances in virtualization (e.g. VMWare) [1], distributed computing with server clusters (e.g. Google) [2] and increase in the availability of broadband Internet access. Industry leaders describe cloud computing simply as the delivery of applications or IT services, which are provided by a third party over the Internet (Rackspace, Microsoft, IBM) [3, 4, 5]. Ironically, the recent global economic recession served as a booster for interest in cloud computing technologies as organizations sought for ways to reduce their IT budget, while keeping up with performance and profits [6]. The cloud computing buzz began in 2006 with the launch of Amazon EC2, gaining traction in 2007 as seen in the Figure 1. The National Institute of Standards and Technology defines cloud computing as follows: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four

Deployment models" [7]. Cloud computing is currently characterized by having an on demand access to elastic resources via a tenancy model. It typifies the holy grail of no-worries in computing, allowing a company to focus on its core business, paying for all its IT resources as a service.

II. CLOUD COMPUTING SERVICE MODELS

In cloud computing, everything is delivered as a Service (XaaS), from testing and security, to collaboration and metamodeling [8]. The cloud was rapidly becoming a conflagration of buzzwords "as a service". Today there are three main service models, which are agreed on and defined in the NIST document [9].

1. Software as a Service {SaaS}

This simply means delivering software over the Internet. It is the most widely known model of cloud computing. SaaS has been around since early 2001 when it was commonly referred to as the Application Service Provider (ASP) Model [8]. Software as a Service consists of software running on the provider's cloud infrastructure, delivered to (multiple) clients (ondemand) via a thin client (e.g. browser) over the Internet. Typical examples are Google Docs and Salesforce.com CRM.

2. Platform as a Service {PaaS}

This gives a client (developer) the flexibility to build (develop, test and deploy) applications on the provider's platform (API, storage and infrastructure). PaaS stakeholders include the PaaShoster who provides the infrastructure (servers etc), 3. Infrastructure as a Service {IaaS}

Rather than buy servers and build a data centre from ground up, and consequently having to worry about what happens when the website hits a million users, IaaS offers users elastic on demand access to resources (networking, servers and storage), which could be accessed via a service API.

The underlying infrastructure is transparent to the end user, whiles/he retains control over the platform and software running on the infrastructure. IaaSruns on a tenancy model, which employs a usage-based payment approach allowing users to pay for only those resources they actually use.

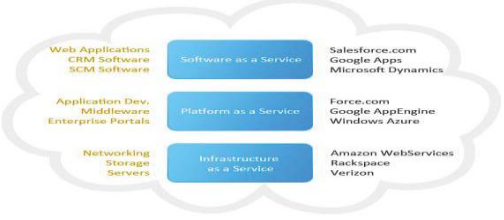


Figure 1: Cloud Computing Service Model

III. CLOUDCOMPUTING DEPLOYMENT MODELS

Depending on infrastructure ownership, there are four deployment models of cloud computing each with its merits and demerits. This is where the security issues start.

1. The Public Cloud

This is the traditional view of cloud computing in every day lingua. It is usually owned by a large organization (e.g. Amazon's EC2, Google's AppEngine and Microsoft's Azure). The owner-organisation makes its infrastructure available to the general public via a multi-tenant model on a self-service basis delivered over the Internet. This is the most cost-effective model leading to substantial savings for the user, albeit with attendant privacy and security issues since the physical location of the provider's infrastructure usually traverses numerous national boundaries.

2. The Private Cloud

It refers to cloud infrastructure in a single tenant environment. It defers from the traditional datacenter in its predominant use of virtualization. It may be managed by the tenant organization or by a third party within or outside the tenant premises. A private cloud costs more than the public cloud, but it leads to more cost savings when compared with a datacenter as evidenced by Concur Technologies (est. savings of $7million in 3 years from 2009) [11]. The private cloud gives an organization greater control over its data and resources. As a result, the private cloud is more appealing to enterprises especially in mission and safety critical organizations.

3. The Community Cloud

According to NIST, the community cloud refers to a cloud infrastructure shared by several organizations within a specific community. It may be managed by any one of the organizations or a third party. A typical example is the Open Cirrus Cloud

Computing Test bed, which is a collection of Federated data centers across six sites spanning from North America toAsia [12].

4. The Hybrid Cloud

It comprises of a combination of any two (or all) of the three models discussed above. Standardization of APIs has lead to easier distribution of applications across different cloud models. This enables newer models such as "Surge Computing" in which Computing Test bed, which is a collection of Federated data centers across six sites spanning from North America toAsia [12].

workload spikes from the private cloud is offset to the public cloud. A comparison of the different issues of cloud computing vis-à-vis deployment models is given in Table 1.

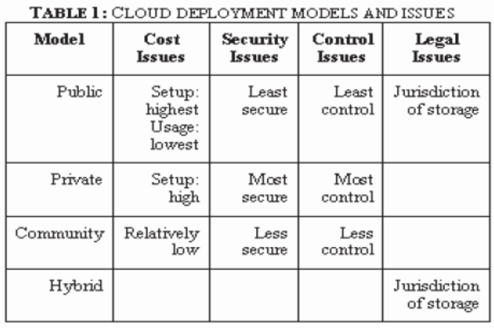


TABLE 1: CLOUD DEPLOYMENT MODELS AND ISSUES

Model	Cost Issues	Security Issues	Control Issues	Legal Issues
Public	Setup: highest Usage: lowest	Least secure	Least control	Jurisdiction of storage
Private	Setup: high	Most secure	Most control	
Community	Relatively low	Less secure	Less control	
Hybrid				Jurisdiction of storage

IV. SECURITY ISSUES IN CLOUD COMPUTING

Security has always been the main issue for IT Executives when it comes to cloud adoption. In two surveys carried out by IDC in 2008 [14] and 2009 [15] respectively, security came top on the list (see Figure 3). However, cloud computing is an agglomeration of technologies, operating systems, storage, networking, virtualization, each fraught with inherent security issues. For example, browser based attacks, denial of service attacks and network intrusion become carry over risks into cloud computing. There are potentials for a new wave of large-scale attacks via the virtualization platform. Chow et al. [16] described the "Fear of the Cloud" by categorizing security concerns into three traditional concerns, availability and third party data control. Research firm Gartner [17] posited seven security risks ranging from data location and segregation to recovery and long-term viability. The European Network and Information Security Agency [18] also published a list of 35 issues in cloud computing in 4 categories. Organizations such as ISACA and Cloud Security Alliance publish guidelines and best practices to mitigate the security issues in the cloud [19, 20].

The cloud service providers must be able to recognize correct user in order to grant the service to him over a network. Otherwise an intruder can also get access to the server if the security mechanism is not safe. An intruder can also deny the service to the user from the server and can act as though he is a server. All these problems must be considered and solved.

V. KERBEROS

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It is a technology that allows for strong authentication in open, distributed networks.

In an open network computing environment, a workstation cannot be trusted to identify its users correctly to network services. Kerberos provides an alternative approach whereby a trusted third−party authentication service is used to verify users' identities.

Kerberos is a trusted third−party authentication service based on the model presented by Needham and Schroeder. It is trusted in the sense that each of its clients believes Kerberos' judgment as to the identity of each of its other clients to be accurate. Timestamps (large numbers representing the current date and time) have been added to the original model to aid in the detection of replay. Replay occurs when a message is stolen off the network and resent later. Kerberos keeps a database of its clients and their private keys. The private key is a large number known only to Kerberos and the client it belongs to. In the case that the client is a user, it is an encrypted password. Network services requiring authentication register with Kerberos, as do clients wishing to use those services. The private keys are negotiated at registration. Because Kerberos knows these private keys, it can create messages which convince one client that another is really who it claims to be. Kerberos also generates temporary private keys, called session keys, which are given to two clients and no one else. A session key can be used to encrypt messages between two parties.

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. Kerberos 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 itself for sending passwords over the network.

Important aspects of the model are:

1. It characterizes the environment and the interaction between agents in a way that is compatible with most distributed system approaches, making it easy to integrate Kerberos into applications and systems.

2. It concentrates the maintenance of secrets (i.e., stored passwords) in a small number of places (that can be hardened appropriately) rather than distributing them all over the system.

3. Kerberos separates authentication from the services themselves. The file server, for example, does not know, or ask for, the user's password. Instead, it delegates that job to Kerberos, and relies on information provided by Kerberos to determine the authenticity of a request.

4. It does not require that all the communicating parties have prior relationships with each other, nor to have previously shared any authentication information with each other. Instead, all parties establish prior relationships with the Kerberos service and rely upon it to verify credentials and authorize sessions.

The simple architecture of Kerberos protocol is depicted below. This deliberately oversimplified description highlights the salient points.

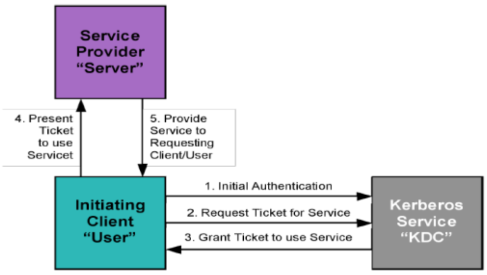

Service Provider "Server"
4. Present Ticket to use Servicet
5. Provide Service to Requesting Client/User
Initiating Client "User"
1. Initial Authentication
2. Request Ticket for Service
3. Grant Ticket to use Service
Kerberos Service "KDC"

Figure 2: Simplified Kerberos System Model

The Kerberos architecture is designed around messages exchanged among three kinds of entities:

1. Clients wishing to use services,

2. Servers that provide services (note that clients and servers are collectively referred to as principals)

3. Servers that manage the Kerberos protocol itself. These servers are often called

"KDCs" (Key Distribution Centers), and actually comprise several modular services.

Clients and servers authenticate each other by means of a protocol involving the exchange of tickets: cryptographically secure, time stamped data structures that contain authentication information and other particulars about a specific proposed interaction between a client and a server.When a new principal (client or server) is added to the system, or when credentials are changed, a secret (e.g., a password) is shared between the principal and a Kerberos server. This set-up stage is the only time that secrets need to be exchanged.

• Subsequently, when a client wishes to log into the system, it obtains a ticket from a Kerberos server. The client does not reveal its secret in the course of obtaining a ticket.Instead, the ticket is constructed such that only the possessor of the client secret can decrypt and use it.

• When clients wish to access servers, they authenticate themselves to the servers by presenting tickets. Once again, no secrets are exchanged; a service ticket is encrypted and delivered in such a way that only the legitimate client could have obtained it, and only the legitimate server can decrypt it.

• Note that there is no real-time communication between the server and the Kerberos infrastructure.

VI. PROBLEM STATEMENT

A. System Model

Representation network architecture for cloud data storage with effect of Kerberos authentication service is illustrated in

Figure 1.Seven different network entities can be identified as follows:

- User: User, who should at the first refer to third party and created the account in the

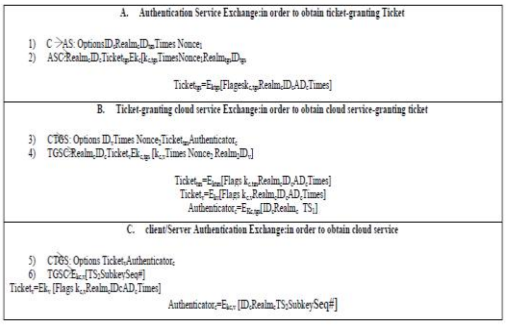
Third party data base and get the password, session key want store data in the cloud and rely on the cloud for data computation, consist of both individual consumer and organization consume.

- Cloud service provider: Cloud service providers offer cloud solutions, like Google Apps, that are delivered electronically over the internet. Unlike a managed service provider, cloud service providers do not sell or install hardware everything they offer is stored online and accessible securely from anywhere. There are many advantages to working with a cloud service provider like Cloud Sherpa when switching from your old email and collaboration software.
- Kerberos operation: Kerberos is an authentication mechanism that provides a secure means of authentication for network users. It prevents transmission of clear text passwords over the network by encrypting authentication messages between clients and servers. In addition, Kerberos provides a system for authorization in the form of administering tokens, or credentials [9].In the other definition of Kerberos is an authentication protocol for trusted hosts on un-trusted networks.
- The maximum time a ticket associated to the principal may be renewed.
- The attributes or flags characterizing the behavior of the tickets.
- The password expiration date.
- The expiration date of the principal, after which no tickets will be issued.
- Third party: The third party defines who has the correctness, expertise, capabilities to access and utilize the cloud service provider.B. Design Goals: To ensure the security of storage the data in cloud server we design efficient mechanisms with 4 part for achieve the following goals:
- Lightly the work: each user can perform storage and register in minimum time.
- Trustworthy: to ensure user that their data is store in trust manner and can execute their job in accurate type.

VII. IMPLEMENTATION OF PROCESS

In cloud data storage system, users store their data in the cloud and for accessing must refer to cloud server provider. Thus the correctness of the user being refer to the distributed cloud server must be guaranteed because the data stored in the cloud may be frequently, updated with user including insertion, deletion, modification, appending, reordering, etc. To ensure this updating is under correctness user is important so in this paper we introduce one model based on kerberos. In this model each user for gain the cloud server must be register and authentication with third party. After added the requirement information into the data base it can get some qualification. After getting the qualification it should refer to the Kerberos authentication service and do this scenario:

In this scenario [5]



A. Authentication Service Exchange:in order to obtain ticket-granting Ticket 1) C → AS: Options ID$_c$ Realm$_c$ ID$_{tgs}$ Times Nonce$_1$ 2) AS → C: Realm$_c$ ID$_c$ Ticket$_{tgs}$ Ek$_c$[k$_{c,tgs}$ Times Nonce$_1$ Realm$_{tgs}$ ID$_{tgs}$ Ticket$_{tgs}$=E$_{ktgs}$[Flagesk$_{c,tgs}$ Realm$_c$ ID$_c$ AD$_c$ Times]
B. Ticket-granting cloud service Exchange:in order to obtain cloud service-granting ticket 3) C → TGS: Options ID$_v$ Times Nonce$_2$ Ticket$_{tgs}$ Authenticator$_c$ 4) TGS → C: Realm$_c$ ID$_c$ Ticket$_v$ Ek$_{c,tgs}$ [k$_{c,v}$ Times Nonce$_2$ Realm$_v$ ID$_v$] Ticket$_{tgs}$=E$_{ktgs}$[Flags k$_{c,tgs}$ Realm$_c$ ID$_c$ AD$_c$ Times] Ticket$_v$=E$_{kv}$[Flags k$_{c,v}$ Realm$_c$ ID$_c$ AD$_c$ Times] Authenticator$_c$=E$_{Kc,tgs}$[ID$_c$ Realm$_c$ TS$_1$]
C. client/Server Authentication Exchange:in order to obtain cloud service 5) C → TGS: Options Ticket$_v$ Authenticator$_c$ 6) TGS → C: E$_{kc,v}$[TS$_2$ SubkeySeq#] Ticket$_v$=Ek$_v$ [Flags k$_{c,v}$ Realm$_c$ IDc AD$_c$ Times] Authenticator$_c$=E$_{kc,v}$ [ID$_c$ Realm$_c$ TS$_2$ SubkeySeq#]

Table 2: Summary of Kerberos message exchange in cloud service

A. Each client after register in third party it should send the requests access for a ticket granting ticket on behalf of the user by sending its user's ID to the AS, together with TGS ID, indicating a request to use the TGS service and following elements:

- Realm: Represent realm of user
- Option: Used to request that certain flags be set in the returned tickets, as explained in table 1.
- Times: Used by client to request the following time setting in the ticket:
- From: the desired start time for the requested ticket
- Till: the requested expiration time for the requested ticket
- Rtime: requested renew-till time
- Nonce: A random value to be repeated in message A assure that the response is fresh and has not been replaced by an opponent.

B. Get back a ticket-granting ticket, identifying information for the client, and a block encrypted using the encryption key based on the user's password. This block includes the session key to be used between the client and the TGS, times specified in message A, the nonce from message A, and TGS identifying information. The ticket itself include the session key, identifying information for the client, the requested time value, and flags that reflect the status of this ticket and the requested option.

C. The client requests a service-granting ticket on behalf of the user. For this purpose, the client transmits a message to the TGS containing the user's ID, the ID of the desire cloud service, and the ticket-granting ticket. D. The TGS decrypt the incoming ticket and verifies the success of the decryption by the presence of its ID. It checks to make sure that the lifetime has not expired. Then it compares the user ID and network address with the incoming information to authenticate the user. If the use is permitted access to V, the TGS issues a ticket to grant access to the requested cloud service provider. If the user wants access to the same cloud service at a later time, the client can simply use the previously acquired service-granting ticket and need not bother the user for a password. Note that the ticket is encrypted with a secret key (Kv) known only to the TGS and the server, preventing alteration. Finally, with a particular cloud service granting ticket, the client can gain access to the corresponding service with step E.

E. In this step, the client may request as an option that mutual authentication is required. The authentication includes:

• Subkey: The client's choice for an encryption key to be used to protect this specific application session. If this field is omitted, the session key from the ticket (kc,v) is used.

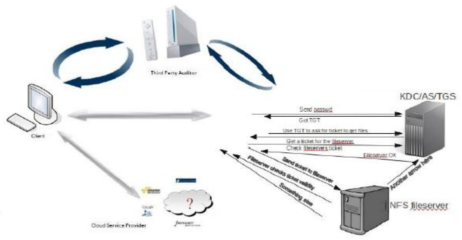


Figure 3: Cloud Data Storage Architecture

VIII. CONCLUSION

In this paper we have discussed about fundamentals of cloud computing. We elaborated the need of secure cloud computing and provided a solution for secure cloud computing by making use of Kerberos. Kerberos provides a centralized authentication server whose function is to authenticate the user to the cloud server and the cloud server to the user. To access the cloud server, all users should make the profile and set a password, and then they can use the cloud server with some restriction which it can make by Kerberos. As we know, the unique attribute of network is security, so in order to make more secure networks; we must make the way for controlling the cloud system and store the information of users. We want the cloud servers to be able to provide access to only authorized users and to be able to authenticate request for service. In an unprotected network environment, any client can apply to any cloud server for service, but Kerberos operation with the use of DES, in a rather elaborate protocol can provide the authentication service. So in my opinion, this task is a best strategy for enhancing the issue of secure cloud computing.

IX. REFERENCES

[1] Virtualization Overview. White Paper. Vmware. Retrieved April 6, 2011, available at:http://www.vmware.com/pdf/virtualization.pdf

[2] Web Search For A Planet: The Google Cluster Architecture. Retrieved April 6, 2011, available at: http://labs.google.com/papers/googlecluster-ieee.pdf

[3] What is Cloud. Retrieved April 6, 2011, availableat: http://www.rackspace.co.uk/cloudhosting/learn-more/whatis-cloud/

[4] What is Cloud Computing. Retrieved April 6, 2011, available at: http://www.microsoft.com/business/engb/solutions/Pages/Cloud.aspx

[5] What is Cloud Computing. Retrieved April 6, 2011, available at: http://www.ibm.com/developerworks/cloud/newto.html#W HATIS

[6] Recession is good for cloud computing – Microsoft agrees - http://www.cloudave.com/2425/recession-isgoodfor-cloud-computing-microsoft-agrees/

[7] National Institute of Standards and Technology -Computer Security Division http://csrc.nist.gov/groups/SNS/cloud-computing/

[8] Bhaskar P., Admela J•, Dimitrios K•, Yves G.: Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. J. Grid Computing 9(1), 3-26 (2011)

[9] What the Hell is Cloud Computing. Retrieved April 6, 2011, available at:http://www.youtube.com/watch?v=0FacYAI6DY0

[10] Boniface, M., Nasser, B., Papay, J., Phillips, S., Servin, A., Zlatev, Z., Yang, K. X., Katsaros, G., Konstanteli, K.,Kousiouris, G., Menychtas, A., Kyriazis, D. and Gogouvitis, S., "Platform-as-a-Service Architecture for Real-time Quality of Service Management in Clouds", Fifth International Conference on Internet and Web Applications and Services, ICIW 2010, May 2010, Barcelona