Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 5, Special Issue 2, December (2015)

An Eager Strategy for TGT Generation at Client Side for Kerberos Protocol

Deepika Yadav
M.Tech Scholar, B.S.Anangpuria Institute of Technology &Management, Faridabad.

Dr. A.K.Sharma
Dean & Prof., Department of computer science &engineering
B.S.Anangpuria Institute of Technology &Management, Faridabad.

Abstract – In this paper we present architecture of a proposed strategy called "An eager strategy for TGT generation at client side". The strategy provides an alternative approach whereby a client is authenticated by an authentication server located at the client side. Here main aim is to eliminate the complex re-authentication process if client requests for same service within a time period. An authentication server database is maintained at the client side to store TGT and Service ticket. The storage of ticket enables client to easily connect to the application server and thus reducing the complex process of ticket generation and exchange.

Index Terms – Kerberos, client side agent, distributed operating system, authentication server, ticket granting ticket.

This paper is presented at International Conference on Recent Trends in Computer and information Technology Research on 25th & 26th September (2015) conducted by B. S. Anangpuria Institute of Technology & Management, Village-Alampur, Ballabgarh-Sohna Road, Faridabad.

1. INTRODUCTION

An Eager Strategy for TGT generation at Client side is an authentication strategy whose main aim is to eliminate the complex re-authentication process. When client requests for same network service then he needs to be re-authenticated. The re-authentication process requires lot of time and message flow. By applying the proposed strategy, re-authentication can be eliminated. An agent is located at the client side that helps in efficient message flow between client and the authentication server/Key Distribution Centre (KDC). Authentication server is also maintained at the client side that verifies the identity of the user. Thus, the dependency on the KDC for authentication also gets reduced by applying the strategy. Authentication server database stores the TGT and the session keys that helps client to easily connect to the application server if he needs the same network service within a time period. Thus, eliminating the need of re-authentication.

The strategy basically works on the client–server model. Client requests service and the server replies back to the client. To provide mutual authentication—both the client and the server verify each other's identity. It protects the client messages against eavesdropping and replay attacks. It builds on symmetric key cryptography and optionally may use public-key cryptography during certain phases of authentication. It provides the security over the distributed network by transforming the password entered by the client with the help of one-way hash function depending upon the cipher suit used and authenticate the clients with the help of an agent.

2. RELATED WORK

Various research papers have been reviewed to understand the working of Kerberos protocol. The Kerberos protocol is a trusted third party authentication protocol which basically authenticates the client with the help of the authentication server and issues a Ticket Granting Ticket to the client. It also issues service ticket to the client. With the help of the service ticket, client can directly communicate with the server. It provides the reliable communication over the distributed environment by identifying the client's identities of the same domain. Various limitations arise after reviewing the papers. These are as follows:

- Complex re-authentication process.
- Heavy network load on key distribution center.
- Large no. of steps to access the network service.

2.1. Complex re-authentication process.

When client wants network service then he needs to be authenticated by authentication server. After sometime if connection gets lost or over then again, client has to re-authenticate, that requires lot of time and effort.

2.2. Heavy network load on key distribution center.

Authentication and ticket generation process is carried out at key distribution center that puts a heavy network load on it. Thus affects performance of key distribution center.

2.3. Large no. of steps to access the network service.

No. of steps are required to connect to the application server. Client needs to first authenticate before requesting for the service. Authentication is performed by the authentication server that provides ticket grating ticket (TGT) to the client. After receiving the TGT from authentication server, client is able to send request to Ticket granting server (TGS). The Ticket granting server identifies the client and provides a service ticket to the client. When client receives service ticket then it gets connected to the application server. It is a very time consuming process since it requires no. of steps to connect to application server.

3. PORPOSED MODELLING

In this strategy, the authentication of the client is provided by the agent situated at the client side. The service ticket is generated by the ticket granting server (TGS) .It basically works in two steps. The first step includes that client need to authenticate with the help of the authentication server and obtain a tgt (ticket granting ticket) .In the second step the agent obtains the service ticket to access the network service from the key distribution center. The entities used in our proposed architecture are as follows:

3.1. Client

Client is the application acting on behalf of user who needs to access a resource, such as opening a file, querying a database, or printing a document. Every client requests must be authenticated before the resource is accessed.

3.2. Agent

Agent is software situated at the client side. It acts as an intermediate between client and authentication server/Key distribution center. Once the agent receives messages from client, it attempts to decrypt message with the secret key generated by the password entered by the user. Client is verified by matching the password with the user password which is stored in the authentication server data base. If it matches then, he is a genuine user otherwise not. After verifying, the agent sends the TGT (Ticket Granting Ticket) to the key distribution center and the authenticator, encrypted with the Logon session key generated by the authentication server.

3.3. Authentication Server

The Authentication server is also situated at the client side. Client sends a message to the authentication server for requesting a service. Authentication server verifies client and after verification it generates the Logon session key and TGT (Ticket Granting Ticket).

3.4. Key Distribution Center

The Key Distribution Center (KDC) is a network service that supplies service tickets and temporary session keys to users within an active directory domain. It acts as a trusted third party between client and the application server.

3.5. Ticket Granting Server

Ticket Granting Server is a logical entity situated at the Key Distribution center (KDC). The server issues tickets for connection to computers in its own domain. When clients want access to a network service, they contact ticket-granting server, present a TGT, and ask for a ticket. The ticket can be reused until it expires, but the first access to any computer always requires a trip to the ticket-granting server.

3.6. Application Server

Application server is the host which provides the service needed by the client. Agent provides service ticket to the application server. For mutual authentication, server also verifies itself by sending an authenticator back to the client.

3.7. Authenticator

Authenticator is a piece of information that helps in verifying client and server. The information in the authenticator must be different each time the protocol is executed; otherwise an old authenticator could be reused by any one who happens to overhear the communication.

3.8. User Key

When a new client is added into a realm then a private key is assigned to it. The information related to the user (private key) is stored with the user object in the active directory domain of the KDC (Key Distribution Center).The private key Kuser is used to prepare the authenticator by the agent for obtaining the service ticket.

3.9. Logon Session key

The Logon session key SKTGS is prepared by the authentication server which is used to encrypt the authenticator prepared by the TGS (Ticket Granting Server).The Log on session key SK_{TGS} is embedded in the TGT (Ticket Granting Ticket) and in the authenticator prepared by the authentication server at the client side for

receiving the service ticket from the KDC (Key Distribution center).

3.10. Session key

The session key SKservice is prepared by the TGS (Ticket Granting Server). It is shared between the key distribution center and the application server.

3.11. Long term key of the Key Distribution center

It is a secret key used by the authentication server to encrypt the TGT (Ticket Granting Ticket).It is shared between the KDC and the Authentication server situated at the client side.

3.12. Secret key of the server

Kservice is the secret key of the application server which is only known to the KDC (Key distribution center). It is used to encrypt the service ticket prepare by the TGS (Ticket Granting Server).

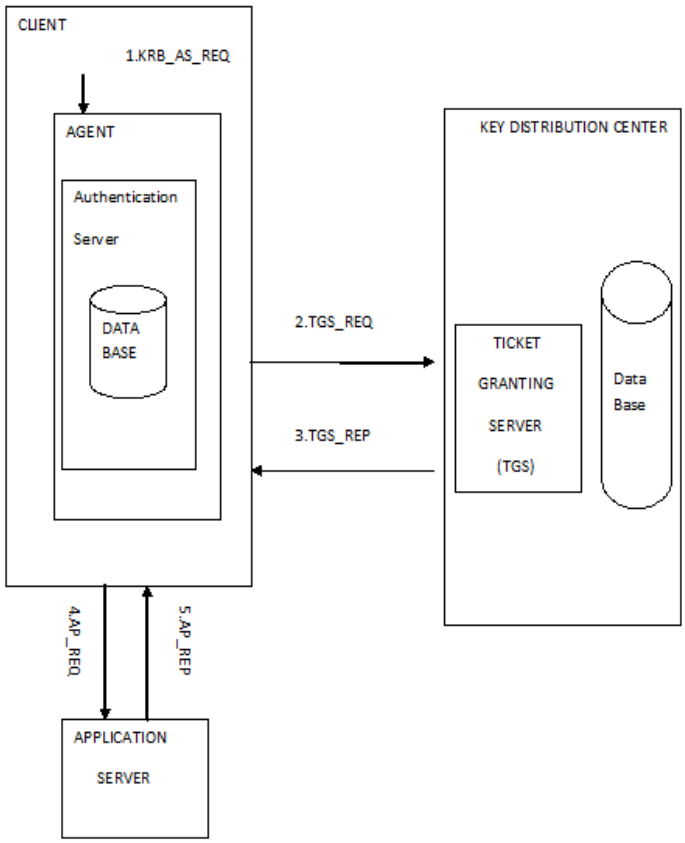

Figure 3: Architecture of Kerberos protocol

Working steps of Kerberos protocol are as follows:

Step1: First client enters username and password to login to a system. After login, if client wants network services then agent helps the client in authentication.

For Authentication:

Client sends a message KRB_AS_REQ to the agent. The message includes the following things:-

- Client principal name.
- Principal service name(i.e. service the client requested for)
- Lifetime (i.e. maximum validity time for the ticket to be used).

The Request format is as follows:-

KRB_AS_REQ= {Principal Client, Principal Service, Lifetime} User key

Step 2: Agent receives the request and decrypts it by transforming the user key into the plain text. Then it searches for principal client requested for the set of principal services by looking into the data base. If a match is found, then a prompt message indicates that the client has been authenticated. Now agent prepares two things:-

- Ticket Granting Ticket (TGT)
- Log on session key SK_{TGS}.

TGT includes the following things:-

- Client principal name.
- Principal Service name (i.e. service the client requested for).
- IP_list (i.e. IP address of the client machine).
- Log on session key SK_{TGS}.
- Lifetime (i.e. maximum validity time for the ticket to be used).

TGT is encrypted with the Long term key of the Key Distribution Center (K_{TGS}).

TGT format is as follows:

TGT= {Principal client, Principal Service, IP_list, Timestamp, Lifetime, SK_{TGS} }K_{TGS}.

Agent stores both the TGT and the log on session key into the user credential cache.

Now agent sends the authenticator and the TGT to the Ticket Granting Server as TGS_REQ. The format of the request is as follows:

TGS_REQ= {{Principal Service, Time stamp, Lifetime, SKTGS} User key {TGT} K_{TGS}

Step 3: Ticket Granting server (TGS) decrypt the authenticator with the help of the user key stored in its data base and extracts the Log on session key.

Now the Ticket Granting Server (TGS) checks the following information:

- Ticket Granting Ticket (TGT) has not been expired.
- Principal client time stamp matches the one present in the TGT.
- IP address of the request packet is one of those contained in the list.
- The log on session key obtained from the authenticator is same as that of the TGT.

If match is found then, TGS prepares an authenticator and encrypt it with the help of the log on session key obtained in the previous step. It also prepares a service ticket and sends both the authenticator and the service ticket as TGS_REP to the agent. The authenticator and service ticket contains the following information:

Authenticator contains:-

- Principal Service name (i.e. Service the client requested for).
- Session key SKservice shared between the TGS and the application server.
- Timestamp (Time of the KDC).
- Lifetime (i.e. maximum validity time for the ticket to be used).

Authenticator format is as follows:

Authenticator= {Principal Service, Timestamp, Lifetime, SKservice} SK_{TGS}.

Service ticket contains:

- Client principal name.
- Principal Service name (i.e. Service the client requested for).
- IP_list (i.e. IP address of the client machine).
- Timestamp (time of the KDC).
- Session key Skservice.
- Lifetime (i.e. maximum validity time for the ticket to be used).

Service ticket is as follows:

Tservice= {Principal client, Principal Service, IP_list, Timestamp, Lifetime, Skservice} Kservice.

TGS sends TGS_REP to agent.

TGS_REP= {Principal Service, Timestamp, Life time,SKservice}SK_{TGS} {Tservice} Kservice.

Step 4: After receiving TGS_REP from the ticket granting server, agent decrypt the authenticator with the help of the logon session key SK_{TGS} and obtain the session key Skservice. Agent also prepares an authenticator and sends it to the application server along with the service ticket.

Authenticator includes the following things:-

- Principal client name (i.e. name of the client).
- Timestamp (time of the agent).

Authenticator format is as follows:-

Authenticator= {Principal client, Timestamp} SKservice.

After preparing authenticator agent sends AP_REQ to the application server.

The format of the request is as follows:-

AP_REQ={Authenticator}SKservice{ Tservice}Kservice.

Step 5: Application server decrypts authenticator and service ticket and provides the service to the client. If mutual authentication is required then application server prepares an authenticator and sends it back to the client for authenticating itself.

Step 6: If client wants same network service then, agent fetch TGT and session keys stored in the authentication server database and prepares a request on the behalf of a client. Thus by storing session keys and TGT, there is no need to authenticate client again.

4. CONCLUSION

Kerberos protocol is a trusted third party authentication protocol which basically authenticates the client with the help of the authentication server and issues a Ticket granting ticket to the client. The proposed strategy tries to overcome the number of steps to access the network service. It provides the

reliable communication over the distributed environment by substituting the authentication server at the client side. Its main aim is to overcome the problem of re-authentication if client requests for same network service within certain time duration. With the help of authentication server database, TGT and session keys are stored that helps in avoiding re-authentication process.

REFERENCES

[1] William Stalling"cryptography And Network Security".Edition 5 year 2007.
[2] Eman EI-Emam+,Margdy Koutb++"A Network Authentication Protocol Based on Kerberos" IJCSNS International Journal of Computer Science and Network Security,VOL.9 No.8,AUGUST 2009.
[3] Saurabh Ratnaparkhi, Anup Bhange "Protecting Against Distributed Denial of ServiceAttacks and its classification :An Network Security Issue" Internatiional Journal of Advanced Research In Computer Science and Software Engineering,Volume 3,Issue 1,January 2013.
[4] http://www.inf.uni-konstanz.de/dbis/teaching/ss06/os/ch14-wrongNumber.pdf
[5] https://www.cs.columbia.edu/~smb/classes/s06-4118/l26.pdf
[6] http://www.cs.helsinki.fi/u/jakangas/Teaching/DistSys/DistSys-08f-1.pdf