Kerberos Authentication

Kerberos is an **authentication** protocol for client/server applications.

This protocol relies on a combination of **private key encryption** and access **tickets** to safely verify user identities.

The main reasons for adopting **Kerberos** are: Plain text passwords are never sent across an insecure network.

It designed to provide strong authentication for client and server.

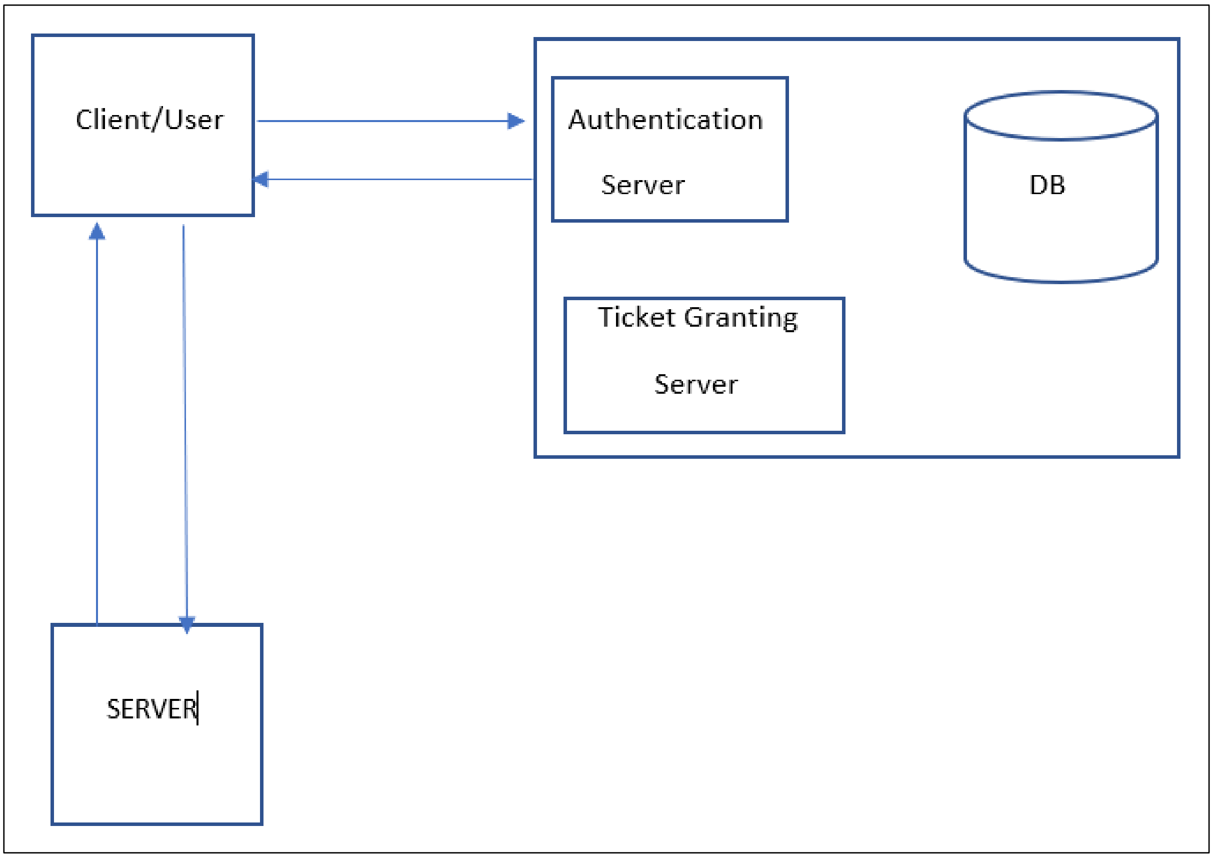
Kerberos Protocol works on UDP/TCP port 88.

Entities involved in Kerberos process are:

I. Client
II. KDC (Key Distribution Center)
III. Server

KDC: Key Distribution Center has 3 entities

a. Authentication Server
b. Ticket Granting Server
c. Database



Kerberos Authentication

Kerberos Authentication Process:

Step1- In first step, client send one request for TGT to Authentication server along with their Client ID. So now Authentication server receive that request and search that Client ID in Database.

If AS (Authentication Server) find client password in respect with that clint ID so Authentication server will generate one secret key and at same time it also generated TGS secret key.

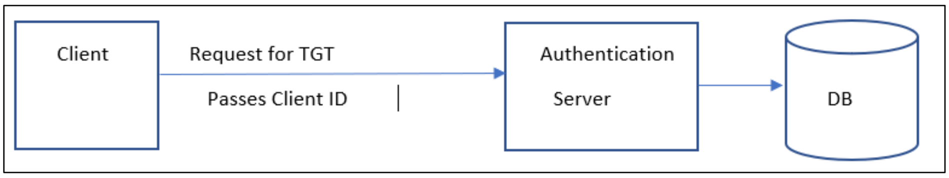


Figure:1 Client Request to Authentication Server

Step2-Now Authentication send a response to Client in which TGT and session key (SK1) are present.

TGT contains Client ID + Timestamp + Session Key (SK1) and TGT is encrypted by TGS secret key.

Session Key (SK1) contains Client ID + Timestamp and encrypted by Client secret key.

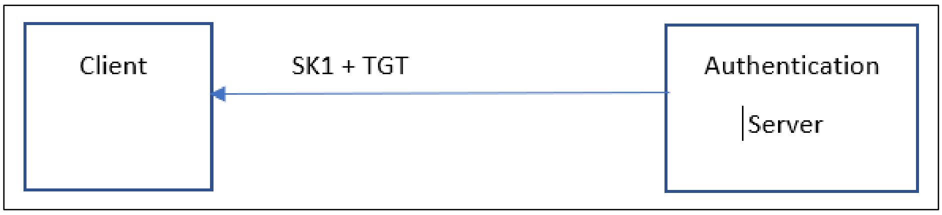


Figure:2 Authentication Server Response to Client

Step3-In third step, client send request to TGS server in which Client send Client ID + Timestamp and both are encrypted by Session Key (SK1) and TGT which contains Client ID + Timestamp + Session Key (SK1) which is encrypted by TGS secret key.

So now TGS server first decrypt TGT from their TGS secret key and once it decrypt it will find session key (SK1) inside TGT and from that session key (SK1) it will decrypt Client ID and Timestamp.

Now TGS server compare both Client ID and Timestamp, if both matches so it will generate new session key (SK2) and TGT2.

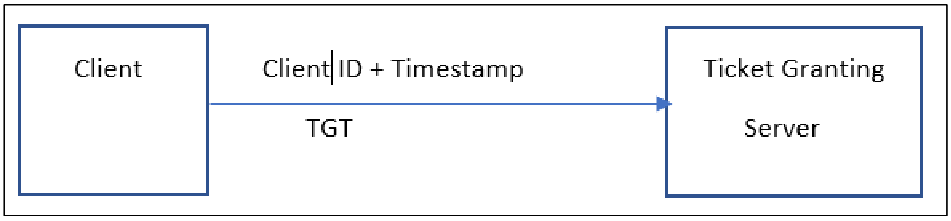


Figure:3 Client Request to TGS Server

Step4- Now Ticket Granting server send response to Client in which it sends SK2 and TGT2.

SK2 is a new session which contains **Client ID and Timestamp** and encrypted by SK1.

TGT2 is a new ticket which contains **SK2 + Client ID + Timestamp** and encrypted by **Server secret key**.

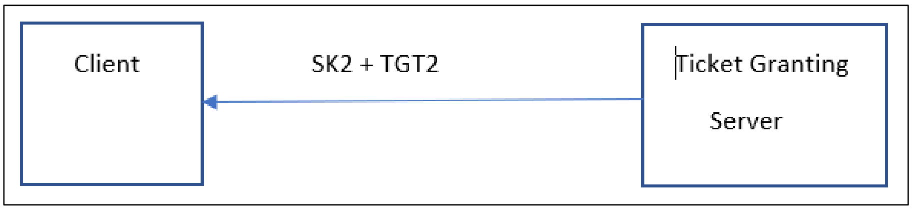


Figure:4 TGS Server Response to Client

Step5-Now client has decrypted SK2 from SK1 and send request to server in which he sends SK2 which contains Client ID + Timestamp and also send TGT2 which contains SK2 + Client ID + Timestamp to server.

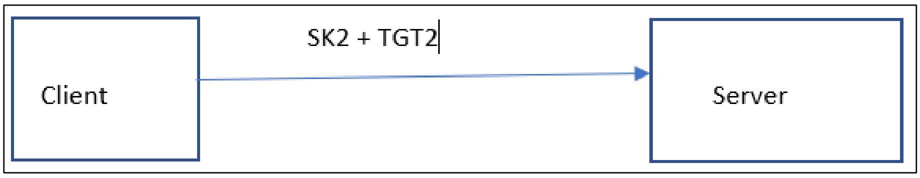


Figure:5 Client Request to Server

Step6- In this step, server decrypt TGT2 from **server secret key**. Once TGT2 decrypt server will find **SK2 , Client ID and Timestamp**. From SK2 it decrypts **Client ID and Timestamp** which is encrypted by SK2. Now server matches both client ID and timestamp and if both matches server will start providing their services to client.

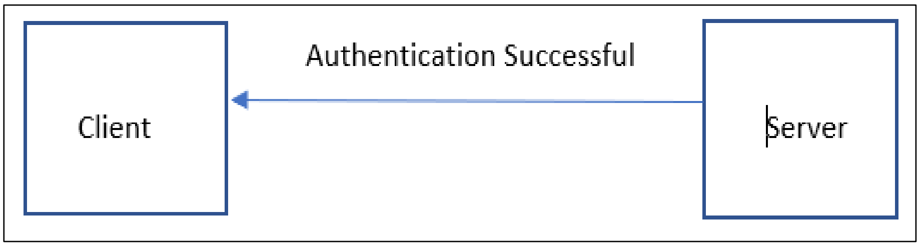


Figure:6 Server Response to Client

=============================END==============================