SIL 765

Kerberos

Harsimrat Singh
2015CS50284

Introduction -

Kerberos is a computer network authentication protocol that works on the basis of *tickets* to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. The protocol was named after the character *Kerberos* (or *Cerberus*) from Greek mythology, the ferocious three-headed guard dog of Hades. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.Kerberos builds on symmetric key cryptography and requires a trusted third party, and optionally may use public-key cryptography during certain phases of authentication.

Working -

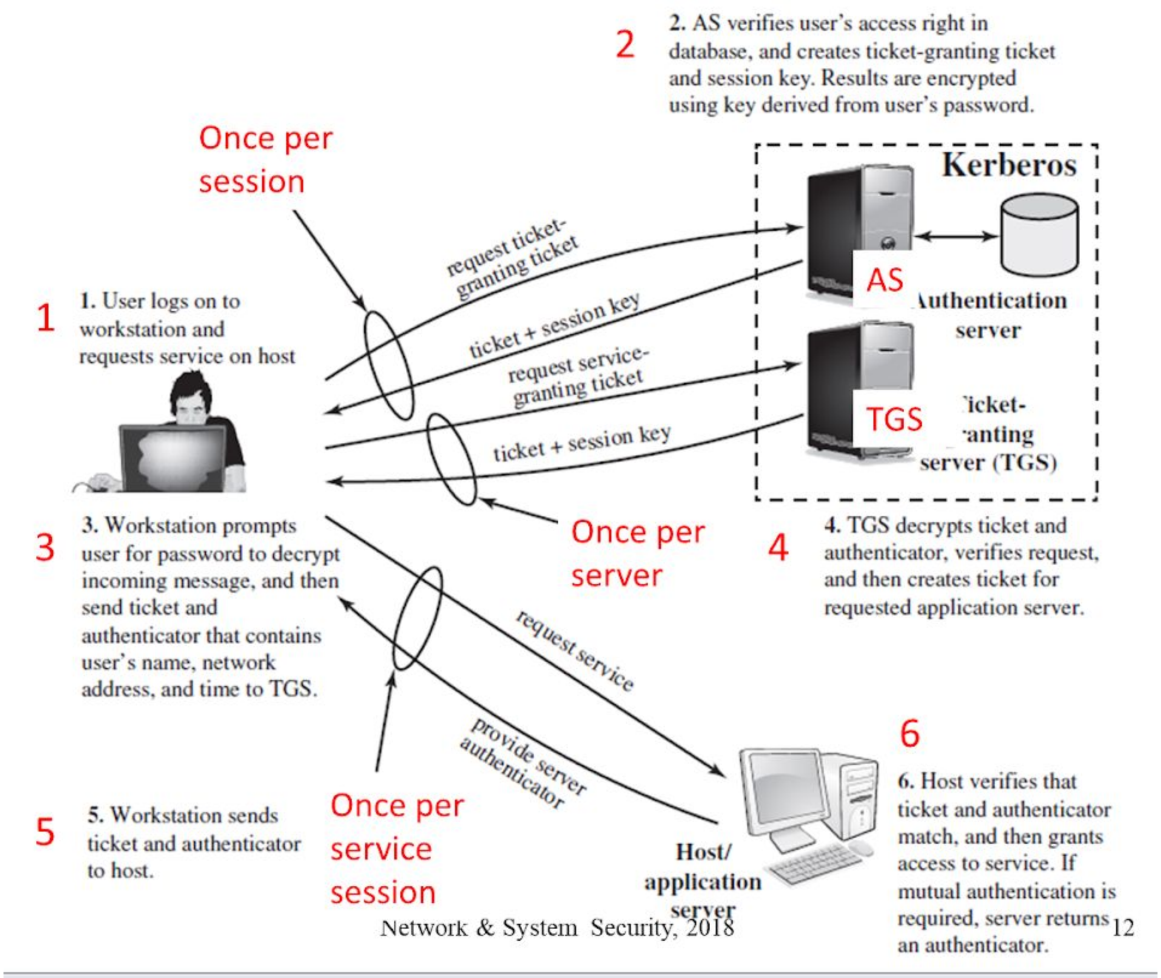
I made a web server which is hosted locally on my pc which opens a login page and ask for username and password from user and then it checks from database whether password is correct or not and whether user exists or not.

The overall picture can be understand by figure 1. To connect to server user need to pass through some steps so that server can be assured that user is authenticated.

Firstly, user need to pass through Authentication Server. User sends AS his client ID and ID of TGS (Ticket Granting Server). Then AS will send him a ticket and key for communication between AS and client which is encrypted by client’s password and client decrypt info by his password and send ticket and some other info to TGS encrypted by key given by AS.

Now, TGS decrypts the data sent by client and verifies the ticket. If ticket is correct then TGS gives another ticket to client which is meant for actual server and key for communication between server and client.

Server receives the ticket generated by TGS, sent from client and can trust the client for further communication.



2 2. AS verifies user's access right in database, and creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

1 1. User logs on to workstation and requests service on host

3 3. Workstation prompts user for password to decrypt incoming message, and then send ticket and authenticator that contains user's name, network address, and time to TGS.

4 4. TGS decrypts ticket and authenticator, verifies request, and then creates ticket for requested application server.

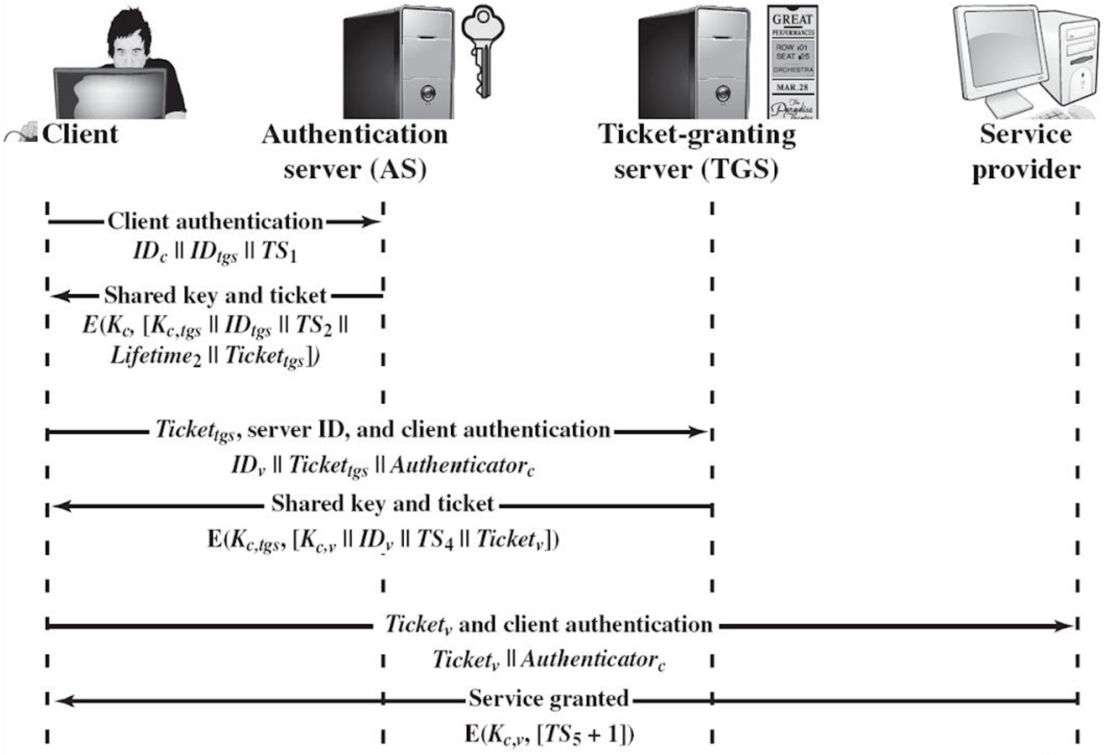
5 5. Workstation sends ticket and authenticator to host.

6 6. Host verifies that ticket and authenticator match, and then grants access to service. If mutual authentication is required, server returns an authenticator.

Network & System Security, 2018 12

Figure 1

Figure 2 shows the complete information exchanged between client, AS, TGS and server along with their encryption with mentioned key. This protocol is very secure and good for authentication. I implemented it as it is. Web server which is hosted locally is the actual server and user enter his credentials there and is sent directly to AS and then back to client and then to TGS and again back to client and finally to server and client then can download homepage.


Client
Authentication server (AS)
Ticket-granting server (TGS)
Service provider
GREAT PERFORMANCES
ROW 01
SEAT 25
ORCHESTRA
MAR.28
Client authentication
$ID_c \parallel ID_{tgs} \parallel TS_1$
Shared key and ticket
$E(K_c, [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}])$
$Ticket_{tgs}$, server ID, and client authentication
$ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$
Shared key and ticket
$\mathrm{E}(K_{c,tgs}, [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v])$
$Ticket_v$ and client authentication
$Ticket_v \parallel Authenticator_c$
Service granted
$\mathrm{E}(K_{c,v}, [TS_5 + 1])$