Network Security
Implementing Kerberos

Chirag Singla 2k17/CO/097
Ayushdeep Dabas 2k17/CO/89

Introduction

This application relates to building a kerberos system, which will include authentication server(AS), ticket granting server(TGS), clients(C) and different servers(V). The clients will try to get tickets for different servers which would be done as per kerberos procedure. After the complete procedure of kerberos, the client would be able to access the webpage of that server(in our case the html file).

Working

Initially the client connects to the Authentication Server(AS). AS verifies the user's access rights and creates a ticket-granting ticket and session key to the user. The client decrypts the message and then sends the ticket and authenticator to the TGS. The TGS decrypts the authenticator, verifies the request, and also creates a ticket for the requested server(requested by client).

The client now has the ticket which he can use to access the server for which he has requested earlier. The client sends the ticket and authenticator to the Server. The Server verifies the ticket and the authenticator and then sends the webpage of that server to the client. The client receives the webpage and the webpage is then opened which shows that he can has successfully accessed the server by mutual authentication.

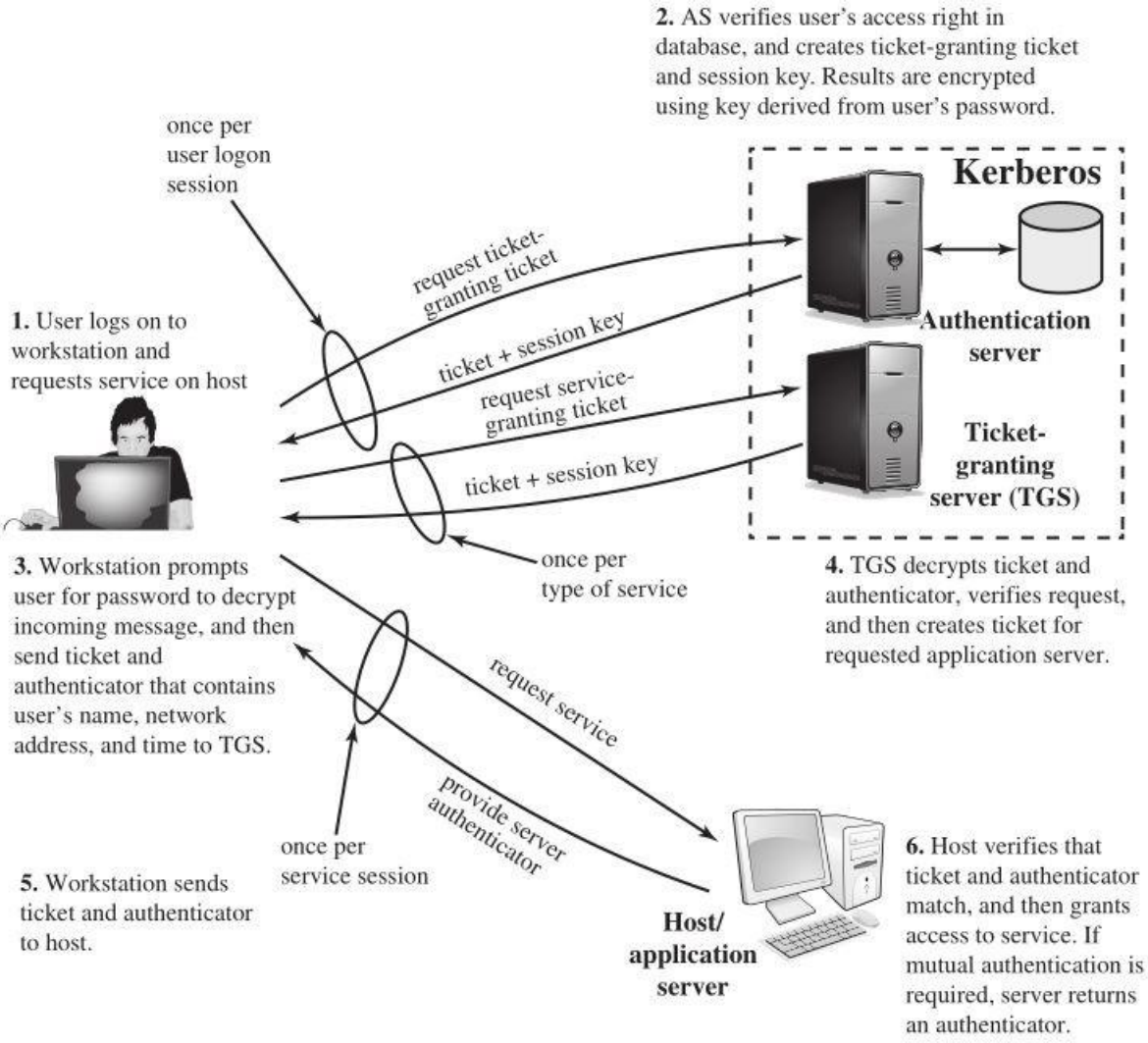


Figure-A: Workflow of the kerberos system

Explanation of Code:

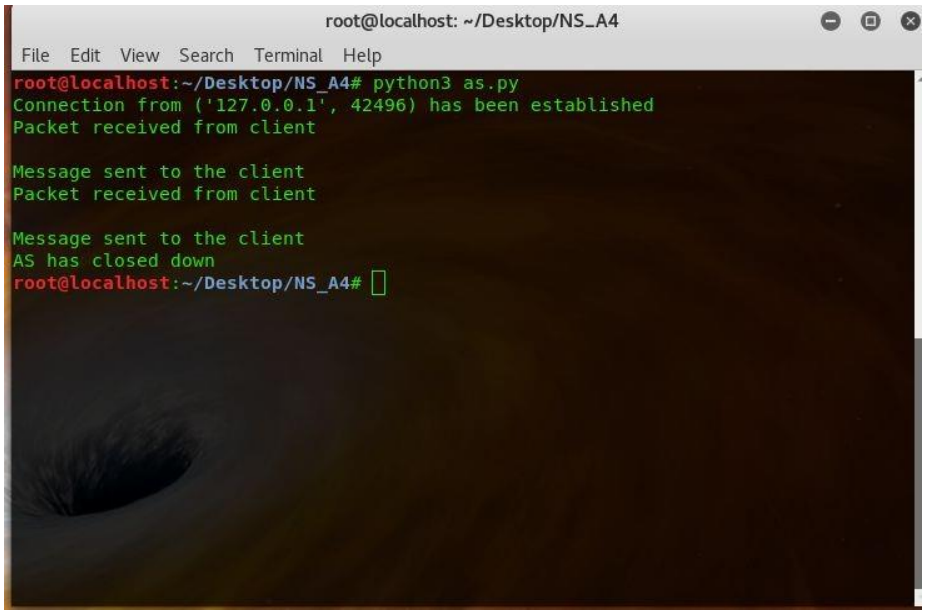
The code of this application has been written in Python. There are 7 Python files namely: AES, as, client1, database, functions, server, tgs.

1) AES.py: This file contains functions related to AES i.e. encrypt, decrypt and generating Keys.

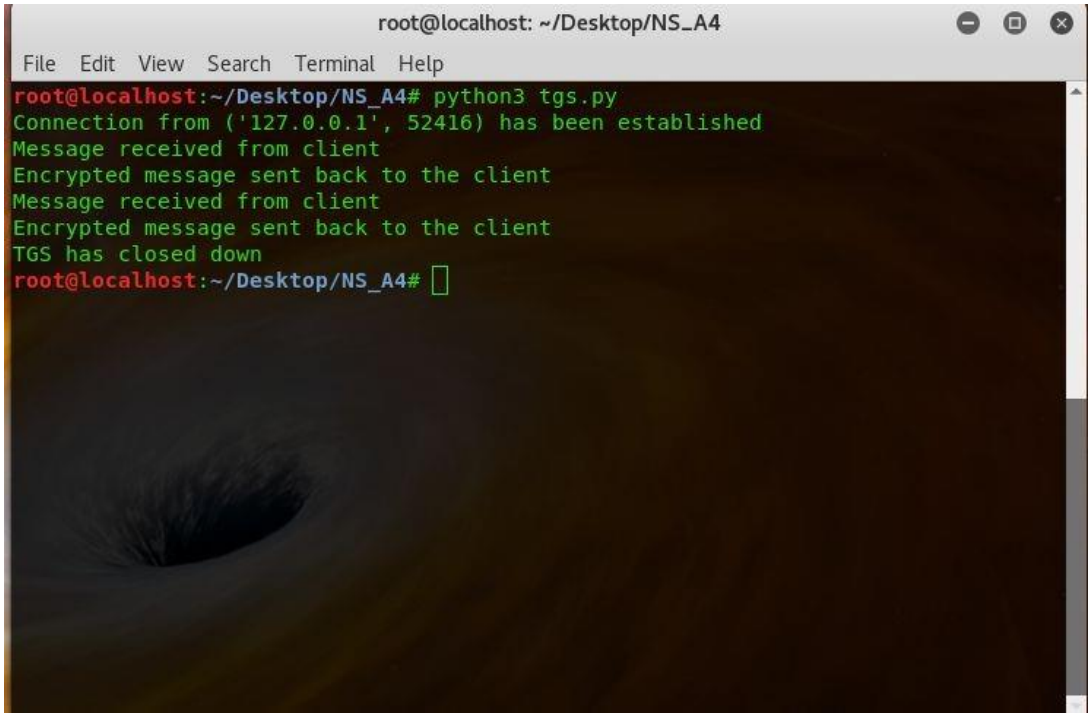
2) as.py(authenticationserver):Theclientrequestsaticketfortheticket granting server and the Authentication server responds by sending a ticket of TGS to theclient.
3) Client1.py: This is the client file where a client tries to generatea sessionkeywiththeapplicationserverinorderto downloadthehomepageofthewebserver.
4) Database.py: Database file at the authentication server endwhich storesthecliensuser_idsandtheirhashedpasswords.
5) functions.py: This file contains helper functions which are neededby otherfiles.Containsfunctionslikegenerate Timestamp,generatePassword,checkTimestamp,etc.
6) server.py: the server contains a webpage(in form html file), after the client connects(based on ticket which) to the server, the serversendsthe webpagetoclient.
7) tgs.py(Ticket Granting Server): TGS receives the ticket provided by AS to the client, after authenticating and the TGS provides asession ticket(key)totheclientfortherequestedserver.

Results:

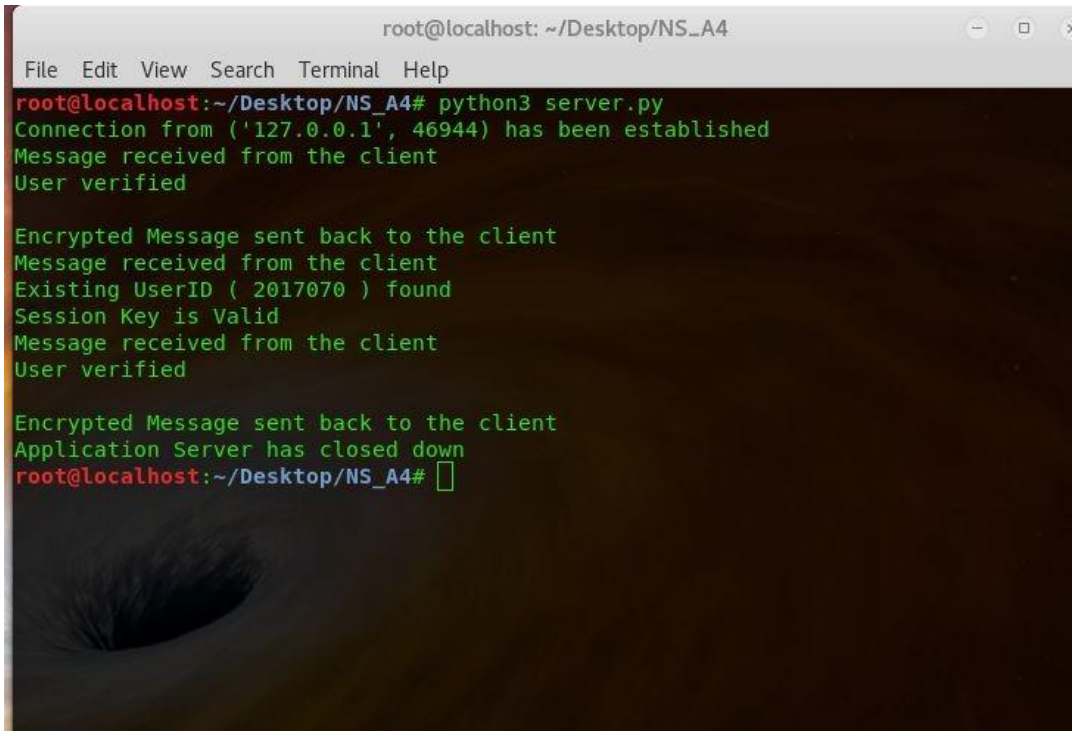
Authentication Server



TGS



Application Server



Client

```
root@localhost: ~/Desktop/NS_A4
File  Edit  View  Search  Terminal  Help
Homepage is downloaded.

1) Run the Client
2) Exit
2
root@localhost:~/Desktop/NS_A4# clear

root@localhost:~/Desktop/NS_A4# python3 client1.py
1) Run the Client
2) Exit
1
Enter the ClientID: 2017070
Password:
Message sent to AS
Message received from AS

Message sent to TGS
Message received from TGS

Message sent to Application Server
This is the final_msg:  1587483867

Do you want to download the homepage of the webserver (y/n): y
Homepage is downloaded.

1) Run the Client
2) Exit
1
Enter the ClientID: 2017070
Password:

Do you want to download the homepage? (y/n): y
Homepage is downloaded.

1) Run the Client
2) Exit
1
Enter the ClientID: 2017304
Password:
Message sent to AS
Message received from AS

Message sent to TGS
Message received from TGS

Message sent to Application Server
This is the final_msg:  1587483908

Do you want to download the homepage of the webserver (y/n): y
Homepage is downloaded.

1) Run the Client
2) Exit
2
root@localhost:~/Desktop/NS_A4#
```

Assumptions: -

1. The client can access different clients simultaneously, but this would require multi threading the client code, so we have assumed that the client connects to one server at a time.
2. The last message from application server to the client i.e. [TS5+1], here TS5 is assumed to be in seconds.