Tema 2
MySSH

Starica Daria Stefana (IIB1)

January 14, 2020

Contents

1 Introducere

Proiectul pe care l-am ales sa-l implementez este MySSH. Acesta implementeaza o pereche client/server capabila de autentificare si comunicare encriptata. Serverul va executa comenzi de la client, si va returna output-ul lor clientului. Comenzile sunt executabile din path, cu oricate argumente. Comenzile cd si pwd vor functiona normal. Se pot executa comenzi multiple legate intre ele au redirectate prin: $|, <, >, 2>, \&\&, ||, ;$.

Proiectul impune deci simularea unui protocol SSH de autentificare si tranfer encriptat de date.

2 Tehnologii utilizate

Tipul de server ales pentru implementarea acestui proiect este de tipul TCP/IP concurent. Alegerea serverului de tip TCP/IP este datorata faptului ca este de incredere pentru transferul de date. Serverul de tip UDP este des folosit in cazurile in care viteza de tranfer este esentiala (de exemplu: livestreaming), insa corectarea erorilor din timpul tranferului sunt ignorate.

Serverul este concurent deoarece este necesara conectarea a mai multor clienti la acelasi server. Concurenta va fi asigurata prin crearea thread-urilor, pentru comunicarea datelor intre client si server.

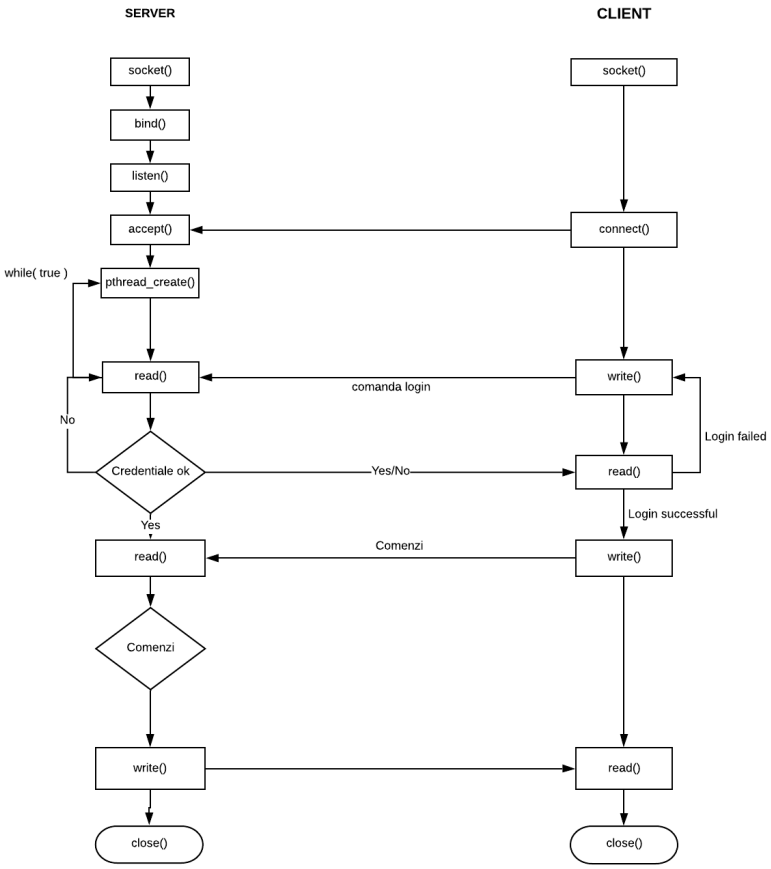
Encriptia este realizata prin o functie ce implementeaza cifrul lui Caesar.

3 Arhitectura aplicatiei

3.1 Concepte implicate

Serverul va astepta conexiuni de la client, astfel incat, in momentul cand un nou utilizator se concteaza, se va creea un nou thread, iar serverul va astepta conexiuni in continuare.
Clientul odata conectat la server, va trimite in prima rand contul si parola autentificatorului, ce vor fi encriptate si valitate de server. Odata reusita autentificarea, se pot trimite comenzi specifice protocolului SSH, pe care servereul le va rula corespunzator.Comenzile specifice linux sunt create cu ajutorul functiilor din libraria unistd. Rezultatul acestora este trimis inapoi la client, care va afisa un mesaj/datele cerute de client.

3.2 Diagrama aplicatiei



4 Detalii de implementare

4.1 Cod relevant

```
1  void* thread_main( void* arg)
2  {
3      char msg[256];
4      thread_inf ld;
5      ld=*((thread_inf*)arg);
6      while(!login)
7      {
8          read(ld.sockd,&lenght,sizeof(int));
9          memset(sConsolaUsername,0, sizeof(sConsolaUsername));
```

```
10          read(ld.sockd,sConsolaUsername,lenght);
11              FILE* fin=fopen("login.txt","r");
12              while(!feof(fin)){
13                  fgets(sFisierUsername,300,fin);
14                  fgets(sFisierParola,300,fin);
15                  sFisierUsername[strlen(sFisierUsername)-1]=0;
16                  sFisierParola[strlen(sFisierParola)-1]=0;
17                  if(strcmp(sConsolaUsername,sFisierUsername)==0){
18                      login=1;
19                      break;
20                     }
21              }
22              fclose(fin);
23              write(ld.sockd,&login,sizeof(int));
24      }
25      login=0;
26      while(!login)
27      {
28          read(ld.sockd,&lenght,sizeof(int));
29          memset(sConsolaParola,0, sizeof(sConsolaParola));
30          read(ld.sockd,sConsolaParola,lenght);
31          if(strcmp(sConsolaParola,sFisierParola)==0)
32                      login=1;
33
34          write(ld.sockd,&login,sizeof(int));
35      }
36
37      while(1)
38      {
39          read(ld.sockd,&lenght,sizeof(int));
40          memset(comanda,0,sizeof(comanda));
41          read(ld.sockd,comanda,lenght);
42          if (strcmp(comanda,"quit")==0){
43                      break;
44                  }
45          if(strcmp(comanda,"find")==0){
46              memset(path,0,sizeof(path));
47              read(ld.sockd,&lenght,sizeof(int));
48              read(ld.sockd,path,lenght);
49              strcpy(startPath,"/home/daria");
50              find(path,startPath,result);
51              write(ld.sockd,result,256);
52              }
53          if(strcmp(comanda,"pwd")==0){
54              pwd(result);
55              write(ld.sockd,result,256);
56          }
57          if(strcmp(comanda,"ls")==0)
58          {
59              ls(result);
```

```
60              write(ld.sockd,result,256);
61          }
62          if(strcmp(comanda,"cd")==0)
63          {
64              memset(path,0,sizeof(path));
65              read(ld.sockd,&lenght,sizeof(int));
66              read(ld.sockd,path,lenght);
67              cd(path,result);
68              write(ld.sockd,result,256);
69          }
70          if(strcmp(comanda,"touch")==0)
71          {
72              memset(path,0,sizeof(path));
73              read(ld.sockd,&lenght,sizeof(int));
74              read(ld.sockd,path,lenght);
75              touch(path,result);
76              write(ld.sockd,result,256);
77          }
78          if(strcmp(comanda,"rm")==0)
79          {
80              memset(path,0,sizeof(path));
81              read(ld.sockd,&lenght,sizeof(int));
82              read(ld.sockd,path,lenght);
83              rm(path,result);
84              write(ld.sockd,result,256);
85          }
86          if(strcmp(comanda,">")==0)
87          {
88              read(ld.sockd,&lenght,sizeof(int));
89              read(ld.sockd,result,lenght);
90
91              memset(path,0,sizeof(path));
92              read(ld.sockd,&lenght,sizeof(int));
93              read(ld.sockd,path,lenght);
94              maimare(path,result);
95              write(ld.sockd,result,256);
96          }
97      }
98      close(ld.sockd);
99      return nullptr;
100 }
```

4.2 Scenarii de utilizare

Utilizatorul va trebui sa introduca structura "ssh-d", urmat de numele de utilizator cu care este inregistrat. Daca numele si parola sunt valitate, se va afisa un mesaj, iar clientului i se va permite sa introduca comenzi specifice SSH. Rezultatul comenzilor va fi afisat in consola.

5 Concluzii

Proiectul MySSH este o implementare minimala a conceptului de protocol SSH. Se pot aduce numeroase imbunatatiri, precum o encriptie mai buna (in loc de cifrul lui Caear, se poate schimba functia, cu o encriptie RSA), o interfata user-friendly sau optiunea de retinere a utilizatotului autentificat, la parasirea aplicatiei.

6 Bibliografie

https://profs.info.uaic.ro/ computernetworks/files/4rc-NivelulTransport-Ro.pdf
https://www.howtogeek.com/190014/htg-explains-what-is-the-difference-between-tcp-and-udp/
https://sites.google.com/view/fii-rc/laboratoare https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/