MySSH

Bicu Andrei Ionel

Universitatea Alexandru Ioan Cuza
Facultatea de informatica

1 Introducere

1.1 Abstract

Tinta este sa construim o aplicatie care permite clientilor logati la server sa execute comenzi la distanta si sa comunice intr-un mod sigur, protejat.

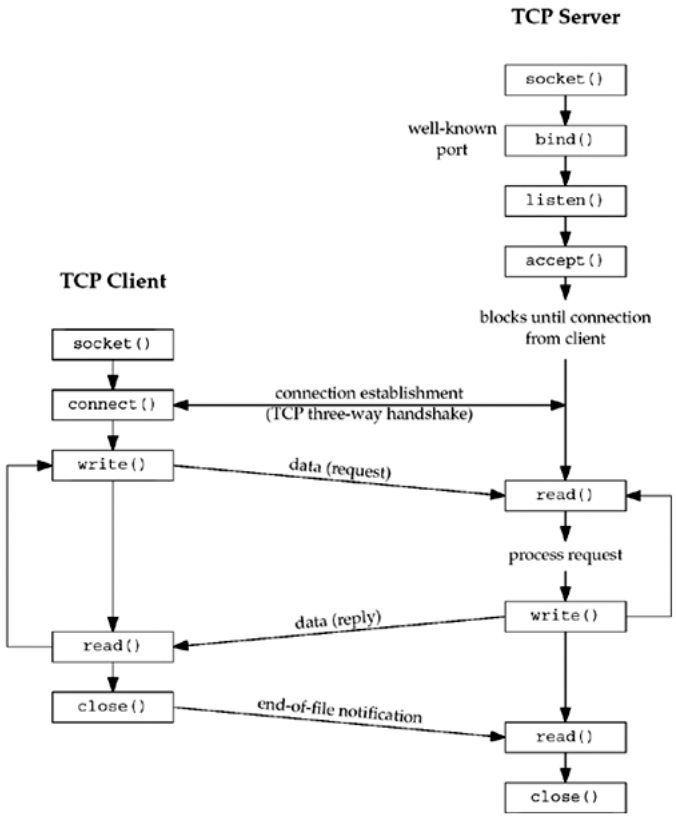
1.2 Utilizarea aplicatiei

Atunci cand un utilizator deschide aplicatia client, el trebuie sa se logheze cu date valide , un nume si o parola. Daca logarea esueaza atunci clientul va primi mesaje de avertizare. In caz de succes clientul va putea executa comenzi la distanta . Comenzile nu au restrictii de acces , clientul are libertate deplina. Comenzile sunt executabile din path, cu oricat de multe argumente.Se pot executa comenzi multiple legate intre ele sau redirectate prin: $|. <, >, 2>, , ||, ;$. Comenzile cd si pwd functioneaza normal.

2 Tehnologi utilizate

2.1 Protocolul de comunicare

Ca si tehnologie am ales sa folosesc suita de protocoale din TCP-IP deoarece sunt nevoit sa trasfer date intre diferiti clienti iar acele date trebuie obligatoriu sa ajunga la destinatar iar mai mult trebuie sa ajunga nealterate, integritatea fiind mai prioritara decat viteza de trasfer. TCP-IP ofera mecanisme de transfer in siguranta a datelor facand autentificare prin 3- way-handshacking si trasmiterea datelor prin ACK-uri ale pachetelor trimise.



2.2 Baze de date

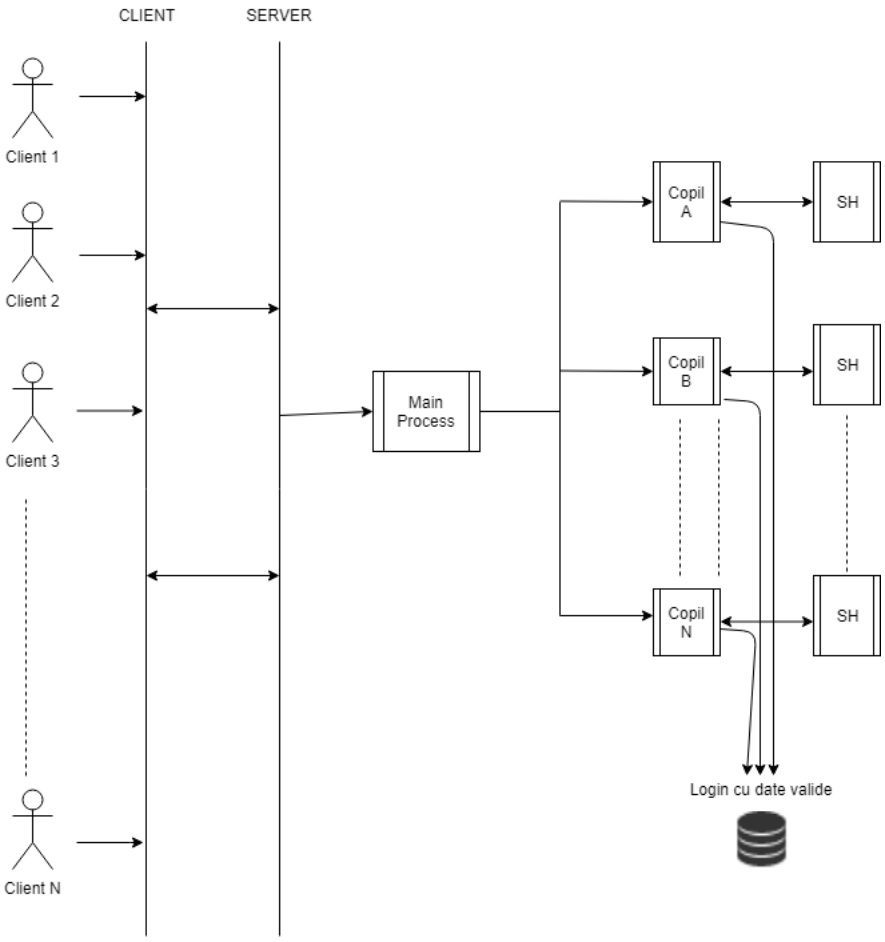
Pentru a realiza logarea am folosit baza de date. In baza de date sunt memorate: userul si parola clientilor. Pentru a implementa bazele de date am folosit sqlite3, interfata C.

2.3 Criptare

Criptarea se realizeaza folosind AES ,algoritmul rijndael-128. Pentru a realiza o encriptia am folosit mcrypt.

3 Arhitectura aplicatiei

3.1 Diagrama aplicatiei detaliata



3.2 Protocolul de comunicare

Pentru a realiza logarea se foloseste comanda login cu atributele -n "USER" -p "PAROLA" in orice ordine. Dupa logare clientul poate efectua comenzi specifice bash.

4 Detalii de implementare

In cazul in care un client pierde conexiunea catre server , serverul nu mai primeste comenzi si inchide conexiunea. Daca "pica" serverul atunci clientul nu va mai primi raspunsul comenzilor.

5 Concluzii

Idee utile de implementat :
-confirmarea executiei unei comenzi
-un timp limita de a realiza a comanda
-cheie de criptare generata random
-limitari de acces catre unele comenzi