Password Manager(C): Manager de parole multiple pentru diferiți utilizatori.

- Documentație proiect final Rețele de Calculatoare -

Tudor-George Motrescu, grupa IIA7

Universitatea “Alexandru Ioan Cuza”, Iasi, Facultatea de Informatica

[samurake96@gmail.com](mailto:samurake96@gmail.com)

1. Introducere:

Proiectul denumit “Password Manager” presupune implementarea unei aplicații compusă dintr-o componentă server (de tip TCP/IP) și o componentă client care are rolul de a organiza toate parolele unui utilizator pentru a elimina nevoia de a le mai reține. În cadrul aplicației, fiecare utilizator poate să își organizeze parolele salvate în diferite categorii, iar pentru fiecare parolă va avea posiblitatea introducerii unor informații adiționale precum titlu, username, url, notițe, adresă, data nașterii. Utilizatorii se pot conecta folosindu-se de un username și o parolă master.

În această lucrare se vor regăsi detalii ale aplicației Password Manager precum și detalii referitoare la tehnologiile utilizate, sistemele implementate și opțiunile utilizatorilor.

1. Tehnologii utilizate

**Sisteme de operare compatibile și limbaje de programare utilizate.** Aplicația va rula pe sistemele de operare Linux (de preferat, **Ubuntu**) și va fi realizată utilizând cod scris in **C/C++**.

**Comunicarea în rețea.** Relativ la comunicarea in rețea, aplicația va utiliza, la nivelul acesteia, un protocol de comunicare ce va permite verificarea utilizatorilor folosindu-se de userul și parola acestora. Apoi utilizatorii au acces la înformații putându-și organiza parolele și câmpurile corespunzătoare acestora după preferințe. Acest protocol se va ajuta de o conexiune bazată pe TCP. Alegerea poate fi justificată în mod evident prin faptul că fișierele document vor conține ***date sensibile*** (parolele salvate, link-urile unde pot fii folosite, notițe despre ce se află pe link-urile de acolo). În plus, aplicația va trebui să se asigure că informațiile relative la utilizatorii acesteia pot fii accesate doar după validările de securitate de rigoare și că un utilizator conectat nu poate accesa informațiile altor utilizatori(acesta poate lucra doar cu informațiile specifice acestuia).

1. Arhitectura

Aplicația ”Password Manager” va fi alcatuită din mai multe componente ce vor interacționa spre a-și îndeplini scopul primar.

* 1. Client-ul

Client-ul este modulul disponibil utilizatorilor finali, pe care îl vor putea utiliza pe stațiile de lucru. El reprezintă un program conceput pentru Linux prin care clienții se pot conecta la server înregistrându-se la aplicație folosind un user și o parolă sau conecta la aceasta folosindu-se de datele de la înregistrare pentru a își adauga/sterge/edita parolele și informațiile specifice acestora.

* 1. Server-ul

Server-ul este componenta care asigură verificările de rigoare pentru utilizatori(în caz că datele lor de conectare sunt greșite conexiunea cu aceștia este sistată), stocarea de date și respectiv accesarea acestora pentru operațiunile de adăugare/ștergere/editare.

Acesta rulează până ce administratorul dorește oprirea lui asigurându-se că toate eventualele legături la clienți în momentul închiderii programului vor fi sistate, dealocând toate resursele utilizate de aceștia și asigurând salvarea datelor folosindu-se de persistența fișierelor.

* 1. Sistemul de înregistrare/login:

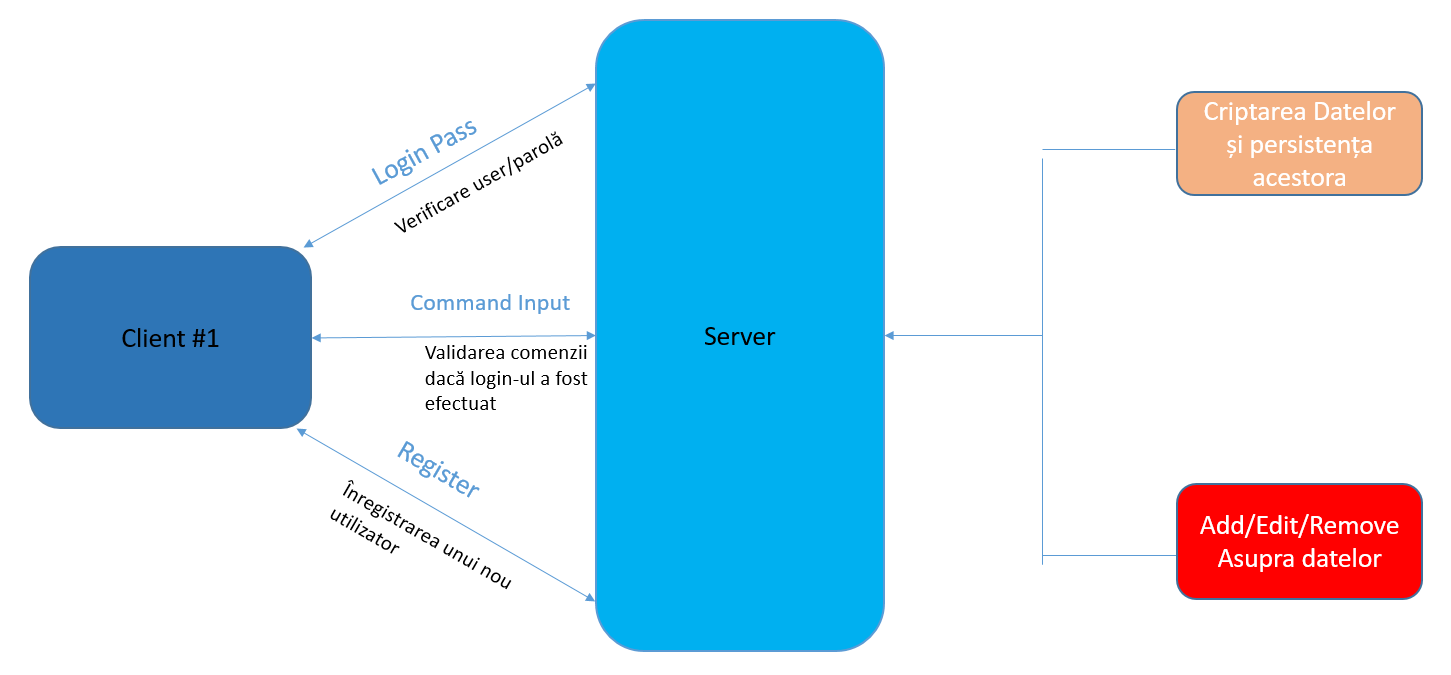
Comunicarea între client și server se va realizaprintr-un protocol de înregistrare/login care permite verificarea utilizatorilor și atribuirea spațiului de lucru corespunzător acestora.

• **Distribuirea resurselor specifice fiecărui client.** Serverul se asigură că datele de login specifice clientului au fost validate și folosindu-se de acestea distribuie acces clientului la ele.

• **Protejarea datelor clienților față de ceilalți utilizatori interni.** Prin sistemul de login se asigură că datele unui client nu sunt intercalate cu datele celorlalți clienți și că clienții au acces doar la datele proprii.

• **Persistența fișierelor.** Server-ul se asigură că datele clienților au fost salvate în caz că acesta a fost închis de administrator sau a avut o eroare și a picat . De asemeni, serverul alocă utilizatorilor câte un fișier individual in functie de numele acestora in care sunt salvate datele sensibile(sensitive data).

* 1. Diagramă arhitecturală



1. Detalii de implementare
   1. Input-ul aplicației

* Utilizatorul alege să se înregistreze dacă e prima dată când folosește aplicația sau să se logheze dacă are deja cont de utilizator și parolă master.
* Utilizatorul poate apoi adăuga/organiza/încărca/șterge apoi datele specifice lui.
  1. Cod relevant
* Functia de validare a username-ului sau parolei.

bool loginCheck(char inBuffer[100], struct user users[100], int userIndex, int &userID)

{

char tokBuffer[100];

CopyRightWord(inBuffer,tokBuffer);

char \*p;

char login[] = "login";

char sep[]= " ";

bool command = false;

bool user = false;

bool password = false;

if(strstr(inBuffer,"login") == NULL)

return false;

for(userID = 0; userID<userIndex; userID++)

{

char buffer[100];

strcpy(buffer,tokBuffer);

p=strtok(buffer,sep);

while(p!=NULL)

{

if(strcmp(login,p)==0)

{

command = true;

}

if(strcmp(p,users[userID].name)==0)

{

user = true;

}

if(strcmp(p,users[userID].masterpass)==0)

{

password = true;

}

p=strtok(NULL,sep);

}

if(command == true && user == true && password == true)

{

if(users[userID].logged\_in == true)

return false;

else users[userID].logged\_in = true;

return true;

}

}

return false;

}

1. Concluzii:

”Password Manager” se adresează clienților care au nevoie de o aplicație simplă de management a parolelor ce asigură o persistență a datelor în cazuri speciale și o manipulare ușoară a acestora.

* 1. Îmbunătățiri posibile:

Ca un adaos la funcționalitatea de bază, prezentată și deja existentă a aplicației, aceasta se poate îmbunătăți prin următoarele:

* **Implementarea unui sistem de criptare**. Se va construi un sistem care să cripteze parolele salvate de către utilizatori..
* **Implementarea unei interfețe grafice folosind Qt / GTK+**. Pentru acces mai ușor la date și realizarea operațiunilor enumerate mai sus într-un mod mai user-friendly se poate folosi o interfață.

1. Bibliografie:

* <https://en.wikipedia.org/wiki/Transmission_Control_Protocol>
* http://stackoverflow.com/questions/11580944/client-to-server-authentication-in-c-using-sockets
* http://vichargrave.com/network-programming-design-patterns-in-c/