Password Authentication and Validation

**Name:**

Rishabh Kumar Jain- PES1201700099

Jeevana R. Hegde- PES1201700633

**Section:**

D

**Objective**:

1. To validate if the password input is secure when creating an account.

2. To authenticate a user’s credentials when he tries to access his account.

**Abstract**:

Our project is a password validation and authentication system.

A file containing the username and password of each valid user is stored in our server. We have our client running on one terminal and our server on another. We establish *a TCP connection between the client and server.* This is chosen instead of a UDP to connect the two, as it ensures *reliable data transfer*.

Once our client has established a connection with the server, the client is presented with four options. The options being:

1. To create an account
2. To login to an already existing account
3. To change the password.
4. Exit.

Any changes made is reflected in the file stored on the server.

* If the user tries to create an account with an existing username, permission is denied and hence no account is created.
* If the user tries to login to a non-existing account or enters the incorrect password, login fails.
* As the password is entered, it is also checked to see if it satisfies all constraints. If not, there is a prompt to enter another valid password.

**Description:**

A socket instance is created and the parameters AF\_INET and SOCK\_STREAM in both client and server are provided. *AF\_INET refers to the family address IPv4. SOCK\_STREAM refers to a connection oriented TCP protocol.*

***client.py***

connect((HOST,PORT)) – To connect the client application to the server application HOST running on the provided port number.

message.encode(‘utf-8’) – Used to convert to a byte addressable object

sendall(message) – Used to send message through socket to server.

recv(message) – Receives message through socket from server.

message.decode(‘utf-8’) – Converts byte addressable object back to string

***server.py:***

bind((HOST,PORT)) – To bind server application on HOST to port number given

listen(1) – Server side socket listens for only one incoming connection

accept() – Accepts an incoming connection

password\_validation(text) – User defined function which takes the new password provided by the client and validates if it is secure enough.

**Technologies** **used:**

Software – Python3,

Hardware –

**Link to the video:**

<https://drive.google.com/drive/folders/197J-O8YYzgdQWaJkfoLuqXEB2JXvdiPm>

**Screenshots :**

***Signup failed since the password is insecure***

A screenshot of a computer screen

Description automatically generated

***Signup successful***

A screenshot of a computer screen

Description automatically generated

***Login failed due to invalid username***

A screenshot of a computer

Description automatically generated

***Login failed due to invalid password***

A screenshot of a computer screen

Description automatically generated

***Login successful***

A screenshot of a computer screen

Description automatically generated

***Password changed***

A screenshot of a computer screen

Description automatically generated