

UNIVERSITY OF
SCIENCE AND TECHNOLOGY OF HANOI

NETWORK PROGRAMMING
FINAL PROJECT REPORT



Group 1: Authentication - BI10 | CS

June 2021

Contents

1	Introduction	2
1.1	Topic	2
1.2	Application overview	2
1.3	Why should you use our project	2
1.4	Members and responsibilities	2
2	Object and Trail	3
2.1	How we done it	3
2.1.1	Create a server	3
2.1.2	Create a text file	4
2.1.3	Connect to client through load balancer	5
2.1.4	Verification	5
2.1.5	Return result	5
3	Testing	5
3.1	Receive the information	5
3.2	Checking information	6
3.3	Checking success	6
4	Comments	6
4.1	What we have done	6
4.2	What could be improved	7
4.3	Current errors	7
5	Conclusion	7

1 Introduction

1.1 Topic

Online Storage System: Authentication

1.2 Application overview

- Written in C
- It is a minimal demo run in terminal

1.3 Why should you use our project

- Verify username and password sent from clients
- It is easy to understand and use and it works

1.4 Members and responsibilities

- Le Thanh Hung BI10-074: Programmer
- Nguyen Khanh Lien BI10-097: Slides
- Nguyen Viet Phuong BI10-143: Reports

2 Object and Trail

2.1 How we done it

2.1.1 Create a server

- authen.c

- By creating a server, the client's connection can be accepted for a verification whether if the username matches with the password. This is to prevent any strangers logging into the client's account, creating security and privacy for the clients to feel secure about their accounts.
- In the main function of authen.c, the system will create socket, binding and listening, printing out this message on the screen if the system is connected successful.

```
Socket made.
Socket binded.
Socket listened.
Connection accepted.
Host resolved.
Socket created.
Connected.
lb-recv-creds: Receiving...
lb-recv-creds: Content: hung      123
lb-recv-creds: Received.
lb-send-creds: Sending...
lb-send-creds: Content: hung      123
lb-send-creds: Sent.
```

Picture 1: Client - Load Balancer - Authentication connected successfully

- Connect to client through Load Balancer thanks to 'send wrapper' and 'recv wrapper'

```
lb-recv-auth-status: Receiving...
lb-recv-auth-status: Content: valid
lb-recv-auth-status: Received.
lb-send-auth-status: Sending...
lb-send-auth-status: Content: valid
lb-send-auth-status: Sent.
```

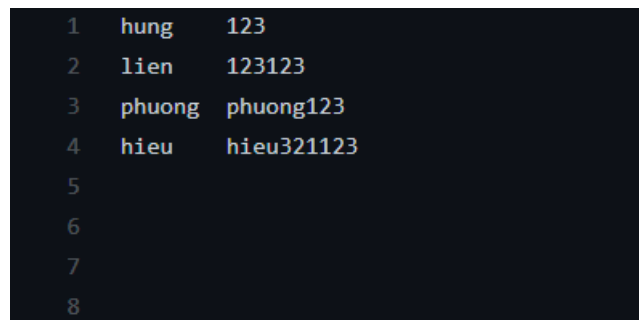
Picture 2: Load balancer checking content validity

- If the username and password does not entered in the right format, the result will come out with the message being planted in, such as:

```
lb-send-auth-status: Receiving...
lb-send-auth-status: Content: invalid
```

2.1.2 Create a text file

- user.txt
 - The placeholder for the clients' usernames and passwords. Therefore, from here, the main code, which is authen.c, can read and check the client's passwords or usernames or both if they are entering the right digit.
 - Because this is a test run, we have set up some examples in for the server to check usernames and passwords.
 - Each usernames and Passwords are separated by a single tab and each individuals are on the same line. Meaning there should be 2 polls for:
 - * 'Usernames'
 - * 'Passwords'



```
1  hung  123
2  lien  123123
3  phuong phuong123
4  hieu  hieu321123
5
6
7
8
```

Picture 3: Example of the usernames and passwords

2.1.3 Connect to client through load balancer

- This is when we receive username and passwords from client when client send through load balancer, which separate by a tab. This is to insure that the information collected is closely resemble to user.txt's format.
- From there, the authentication system will start the next step of verification without index problem or problem to read.

2.1.4 Verification

- After receiving the username and passwords from the connection, the system will start opening the text file, which is the user.txt in this example.
- The system will start comparing the provided information that the client typed in with the one it has in the text file.

2.1.5 Return result

- There are two cases with the results come in return for the client system
 - One: The username and password will let the log in successful message.
 - Two: The wrong username and password will ask the client to retype. their account.
- On the authentication screen, it will come out as 'Valid' if the username and password are correct; or 'invalid' if they are incorrect after verifying in the user.txt.

3 Testing

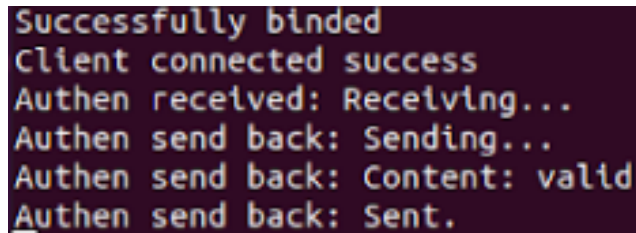
3.1 Receive the information

First of all, we have to receive the username and password of the client.

3.2 Checking information

The system will start opening the user.txt file to do the checking about the information if it true or false.

Once the bind is successfully, the client will connect to the server and they will be pass through the server.

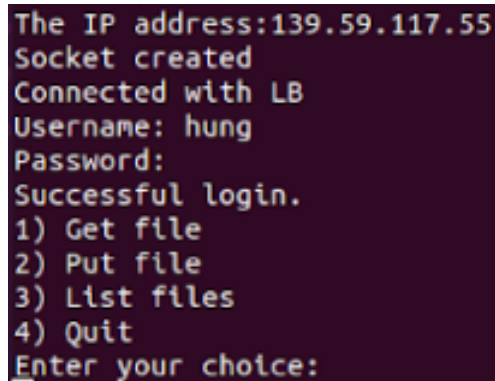


```
Successfully binded
Client connected success
Authen received: Receiving...
Authen send back: Sending...
Authen send back: Content: valid
Authen send back: Sent.
```

Picture 4: Client checking successfully

3.3 Checking success

Once the authentication system has finish checking the information, the client will be able to log in successfully in the same server. This is how it will appear in the screen of the client, provided by the client system project team.



```
The IP address:139.59.117.55
Socket created
Connected with LB
Username: hung
Password:
Successful login.
1) Get file
2) Put file
3) List files
4) Quit
Enter your choice:
█
```

Picture 5: Client receive screen after logging in

4 Comments

4.1 What we have done

- An authentication system that works smoothly when verify the information of the client

- Can be easily use by the clients
- The system can be use as a base for beginners in Network programming.

4.2 What could be improved

- The project can improve a bit of more secure because most of the authentication system that the project is presented are use basics knowledge we learned during the course.
- If we had more time, we would expect our system built in non-blocking mode with hashed user and password when send and received

4.3 Current errors

- So far, after connecting with other projects' system, there are non errors being detected. This is mostly because we use the basic knowledge.

5 Conclusion

The project maybe simple but it works smoothly, of course, it is not a strong authentication system, nor secure in any sides. On the other hand, it has all the function that a system can have to combine with other systems, blocking mode. In the end, this authentication is basically for the project, does not recommended using in reality system, but it is a good base for beginners.