**Лабораторна робота №3**

Тема: ”Розробка і дослідження засобів ідентифікації користувачів в комп’ютерних системах”

Мета роботи: засвоїти методику та отримати практичні навики побудови засобів ідентифікації користувачів.

**Теоретичні відомості**

Ідентифікація об’єкта – це одна з функцій підсистеми захисту. Перед тим, як отримати доступ до комп’ютерної системи (КС), користувач повинен призначена для одноразового встановлення особи користувача та надання йому прав доступу в систему. Одним з найпростіших способів ідентифікації є парольна ідентифікація, яка здійснюється шляхом порівняння введеного імені та паролю, із тими які зберігаються у файлі паролів.

Проте такий підхід не захищає КС від програмних та апаратних засобів сканування клавіатури та ліній передачі, а отже може призвести до витоку конфіденційної інформації. Тому, як правило, в сучасних КС пароль не передається в явній формі по лініях передачі, а натомість в якості паролю використовують якесь його відображення використовуються важко оборотні однонапрямлені функції, застосування яких гарантує неможливість розкриття пароля за його відображенням за розумний час.

В такому випадку процедура ідентифікації описується таким алгоритмом:

1 Користувач вводить свій ідентифікатор.

2. Засоби ідентифікації переглядають список зареєстрованих ідентифікаторів. Якщо ідентифікатор не зареєстрований, - то виводиться повідомлення, що такий користувач в системі не зареєстрований і далі перехід на крок 1, або ж завершення роботи програми входу в систему. Якщо ж ідентифікатор зареєстрований, - то перехід на крок 3.

3. КС генерує випадкове число х, та обчислює значення важкооборотної

однонапрямленої функції у, яка використовується в системі для

відображення паролю користувача.

4. Число х передається користувачу.

5. Користувач обчислює значення важкооборотної функції у' та передає

його в КС.

6. КС порівнює значення у і у'. Якщо вони співпадають то КС дозволяє

вхід користувача в систему. В інакшому випадку видається повідомлення про помилку вводу паролю, перехід на крок 3, або ж завершення роботи програми входу в систему.

**Хід роботи**

Варіант 8

Для розробки програми ідентифікації користувача було вирішено використати мову програмування Python. Для реалізації зв’язку клієнта з центральною системою використовуватиметься технологія socket. Дані користувачів будуть зберігатись в файлі user.txt.

Лістинг 1.1 – файл serv.py

import socket

import threading

import random

import math

import time

import json

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

Dict\_data = {}

MaxSizeofUsers = 9

def run():

    while True:

        socket\_user, adress\_user = server.accept()

        userID = socket\_user.recv(2048).decode()

        if(CheckID(userID)):

            print(userID)

            print(socket\_user, " connected")

            rnd\_value = (random.randint(0, 20))

            socket\_user.send(str(rnd\_value).encode())

            #Варіант 8

            result = int(userID)\*math.sin(1/rnd\_value)

            #Варіант 8

            print("result = ", result)

            UserY = float(socket\_user.recv(2048).decode())

            if(result == UserY):

                socket\_user.send("Connected".encode())

                login = getUserLogin(userID)

                f = open('result.txt', 'a')

                f.write(str(userID)+'\t'+login+'\t'+time.ctime()+'\t'+"connect"+'\t\t'+"online"+'\n')

                f.close()

                thread\_for\_listen = threading.Thread(target=ping\_user, args=(socket\_user, userID, login,))

                thread\_for\_listen.start()

                thread\_for\_listen = threading.Thread(target=getDataFrmUser, args=(socket\_user,))

                thread\_for\_listen.start()

        else:

            socket\_user.send("Your ID is not registered!".encode())

            socket\_user.close()

def getUserLogin(id):

    with open('user.txt', 'r') as json\_file:

        obj = json.load(json\_file)

        for i in obj['users']:

            if (i['ID'] == id):

                return i['login']

def getDataFrmUser(user):

    while True:

        try:

            data = user.recv(2048)

            data = data.decode()

            print(data)

        except:

            pass

def ping\_user(user, userID, login):

    while True:

        time.sleep(2)

        try:

            user.sendall(b"ping")

        except:

            user.close()

            f = open('result.txt', 'a')

            f.write(str(userID) + '\t' + login + '\t' + time.ctime() + '\t' + "disconnect" + '\t' + "offline" + '\n')

            f.close()

            break

def getSizeofUsers():

    count = 0

    with open('user.txt', 'r') as json\_file:

        obj = json.load(json\_file)

        for i in obj['users']:

            count+=1

    return count

def CheckID(id):

    with open('user.txt', 'r') as json\_file:

        obj = json.load(json\_file)

        for i in obj['users']:

            if (i['ID'] == id):

                return True

    return False

def CreateFile():

    f = open('result.txt', 'w')

    f.write('ID\tLOGIN\tDATE\t\t\t\tOPERATION\tSTATUS\n')

    f.close()

def CheckSizeJson():

    with open('user.txt', 'r') as json\_file:

        try:

            obj = json.load(json\_file)

            if(obj == None):

                print("Null")

        except:

            print("mu;;")

def isFileExist():

    with open('user.txt', 'r') as json\_file:

        try:

            obj = json.load(json\_file)

        except:

            Dict\_data['users'] = []

if \_\_name\_\_ == '\_\_main\_\_':

    CreateFile()

    isFileExist()

    server.bind(("127.0.0.1", 5555))

    server.listen(MaxSizeofUsers)

    run()

Лістинг1.2 – файл client.py

import socket

import threading

import math

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

def listen\_system():

    while True:

        try:

            data = client.recv(2048)

            if(data.decode("utf-8") != "ping"):

                print(data.decode("utf-8"))

        except:

            pass

def ConnectSystem():

    print("Input ID")

    MyID = input()

    client.send(MyID.encode("utf-8"))

    rnd\_x = client.recv(2048).decode()

    try:

        rnd\_x = int(rnd\_x)

        print("x = ", rnd\_x)

        intId = int(MyID)

        #Варіант 8

        result = intId\*math.sin(1/rnd\_x)

        #Варіант 8

        print("result = ", result)

        client.send(str(result).encode())

        thread\_for\_listen = threading.Thread(target=listen\_system)

        thread\_for\_listen.start()

    except ValueError:

        print(rnd\_x)

        client.close()

if \_\_name\_\_ == '\_\_main\_\_':

    client.connect(("127.0.0.1", 5555))

    ConnectSystem()

Вивід логів про дії клієнтів зображено на рисунку 1.1.

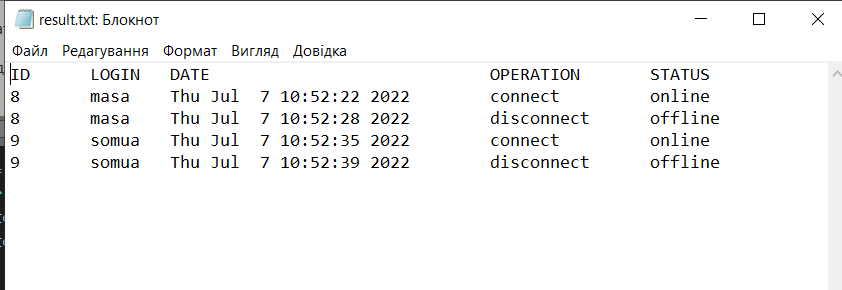


Рисунок 1.1 – Зображено вміст файлу result.txt

Висновки: на даній лабораторній роботі я отримав практичні навики побудови засобів ідентифікації користувачів.