Peer review feedback

0x0 General information

Projection name: Chat App
Code reviewed: All file in the project
Reviewer: Zhihao Cheng
Test method: White box audit, Unit test, Dynamic test.
Feedback Structure:

0x1 Summary

- ✓ Message to person
- ✓ Group message
- ✓ Peer to Peer File Transfer
- ✓ Show online list

Overall, all the requirements have been met, and the program follows the Object-Oriented Programming (OOP) design structure:

1. Relatively user-friendly interface.
2. End to end encryption implemented.
3. Clear structure, good code readability and extensibility.

However, there are a few issues:

1. Outdated RSA encryption methods.
2. Unsafe file transfer service.
3. Pickle deserialization vulnerability
4. Local user login
5. Eval RCE backdoor

The feedback will also provide the improvement and POC for code issues.

0x2 Key strengths

1. A user-friendly GUI
   It makes using the software easier, helping users figure things out quickly and avoid

mistakes.

2. End to end encryption implemented

   End-to-end encryption keeps messages private by making sure only sending user and the target user can read them.

3. OOP are introduced, clear code structure

   Using OOP makes the code more organized and easier to work with by breaking it down into manageable pieces. This makes updates and changes simpler and less confusing.

0x3 Issues, Improvement and POC

1. **(vulnerability) Outdated RSA encryption methods.**

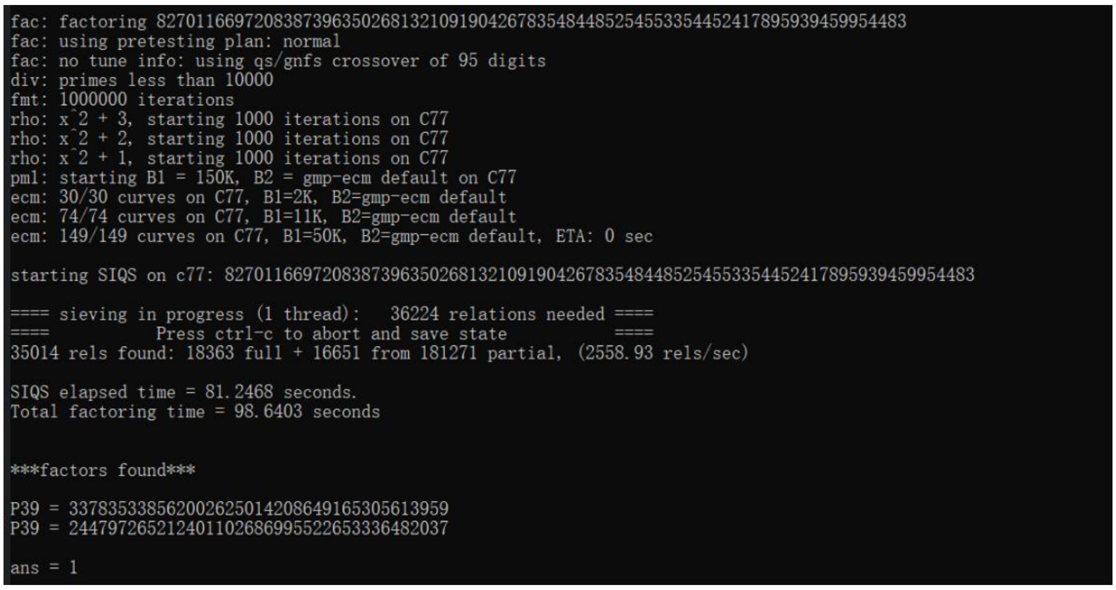
```
(PubKey, PrivateKey) = rsa.newkeys(512)  # encrypt.py line 6
```

RSA's security is based on the difficulty of factoring the product of P and Q. RSA with a 512-bit key is no longer considered secure for modern applications.

Improvement:

```
(PubKey, PrivateKey) = rsa.newkeys(2048) # change to this
```

POC:



```
fac: factoring 82701166972083873963502681321091904267835484485254553354452417895939459954483
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 95 digits
div: primes less than 10000
fmt: 1000000 iterations
rho: x^2 + 3, starting 1000 iterations on C77
rho: x^2 + 2, starting 1000 iterations on C77
rho: x^2 + 1, starting 1000 iterations on C77
pml: starting B1 = 150K, B2 = gmp-ecm default on C77
ecm: 30/30 curves on C77, B1=2K, B2=gmp-ecm default
ecm: 74/74 curves on C77, B1=11K, B2=gmp-ecm default
ecm: 149/149 curves on C77, B1=50K, B2=gmp-ecm default, ETA: 0 sec

starting SIQS on c77: 82701166972083873963502681321091904267835484485254553354452417895939459954483

==== sieving in progress (1 thread):   36224 relations needed ====
====            Press ctrl-c to abort and save state           ====
35014 rels found: 18363 full + 16651 from 181271 partial, (2558.93 rels/sec)

SIQS elapsed time = 81.2468 seconds.
Total factoring time = 98.6403 seconds


***factors found***

P39 = 337835338562002625014208649165305613959
P39 = 244797265212401102686995522653336482037

ans = 1
```

Figure 1 Big int factor

Tools: https://github.com/bbuhrow/yafu

2. **(vulnerability) Unsafe file transfer service** (file_server.py).

   The file server lacks commend validation, allowing unrestricted access to anyone. Malicious attacks can arbitrarily read and write access to all files.

   Improvement: Use Pipes (example) to inter-process communication

   POC: https://github.com/FrogGuaGua/SQRPRGRM/blob/main/2/2/poc.py

3. **(vulnerability) Pickle deserialization vulnerability.**

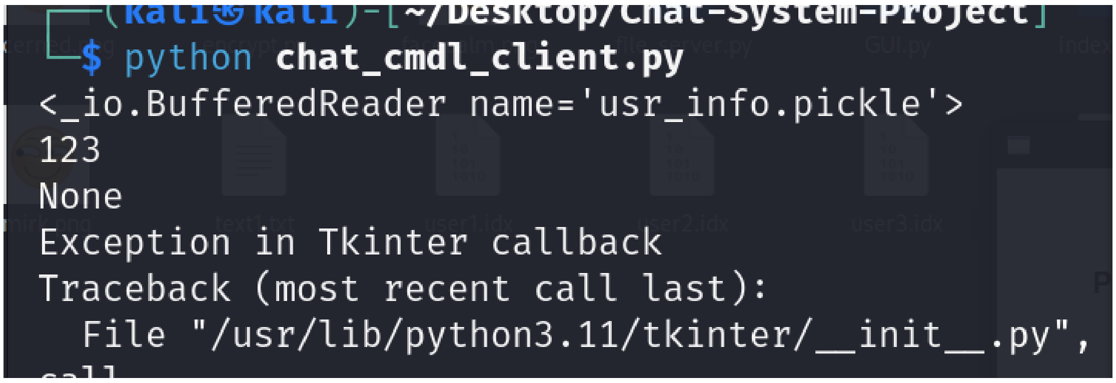
```
usrs_info = pickle.load(usr_file)   #GUI.py line 112
```

```
exist_usr_info = pickle.load(usr_file) #GUI.py line 156
(recvdata, PrivateKey) = pickle.loads(Message) #encrypt.py line 56
```

While using pickle.load or pickle.loads to load objects, it can invoke the __reduce__ method. If an attacker can control the serialized data being deserialized, they could exploit this mechanism to execute malicious code.

Improvement: Using safer library(json) as an alternative.

POC: https://github.com/FrogGuaGua/SQRPRGRM/blob/main/2/3/poc.py



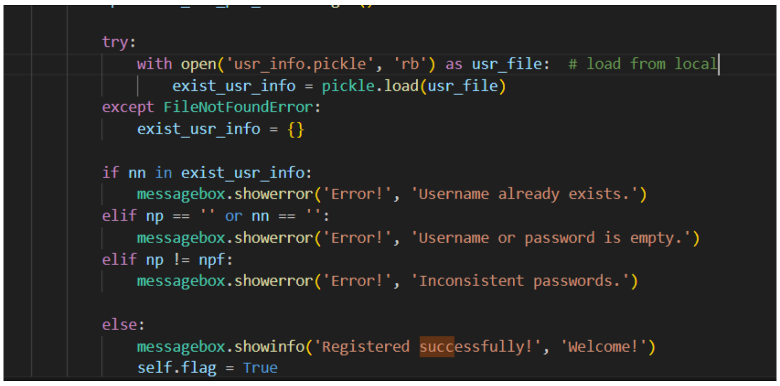
```
┌──(kali㉿kali)-[~/Desktop/Chat-System-Project]
└─$ python chat_cmdl_client.py
<_io.BufferedReader name='usr_info.pickle'>
123
None
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.11/tkinter/__init__.py",
```

Figure 2 Pickle RCE

4. **(vulnerability) Local user login**

User verification is performed on the client side instead of being handled by the server. This approach may be less secure and more susceptible to tampering or attacks.

Improvement: Move user verification to the server side. This ensures security and integrity that all authentication and authorization processes are controlled.



```
try:
    with open('usr_info.pickle', 'rb') as usr_file:  # load from local
        exist_usr_info = pickle.load(usr_file)
except FileNotFoundError:
    exist_usr_info = {}

if nn in exist_usr_info:
    messagebox.showerror('Error!', 'Username already exists.')
elif np == '' or nn == '':
    messagebox.showerror('Error!', 'Username or password is empty.')
elif np != npf:
    messagebox.showerror('Error!', 'Inconsistent passwords.')

else:
    messagebox.showinfo('Registered successfully!', 'Welcome!')
    self.flag = True
```

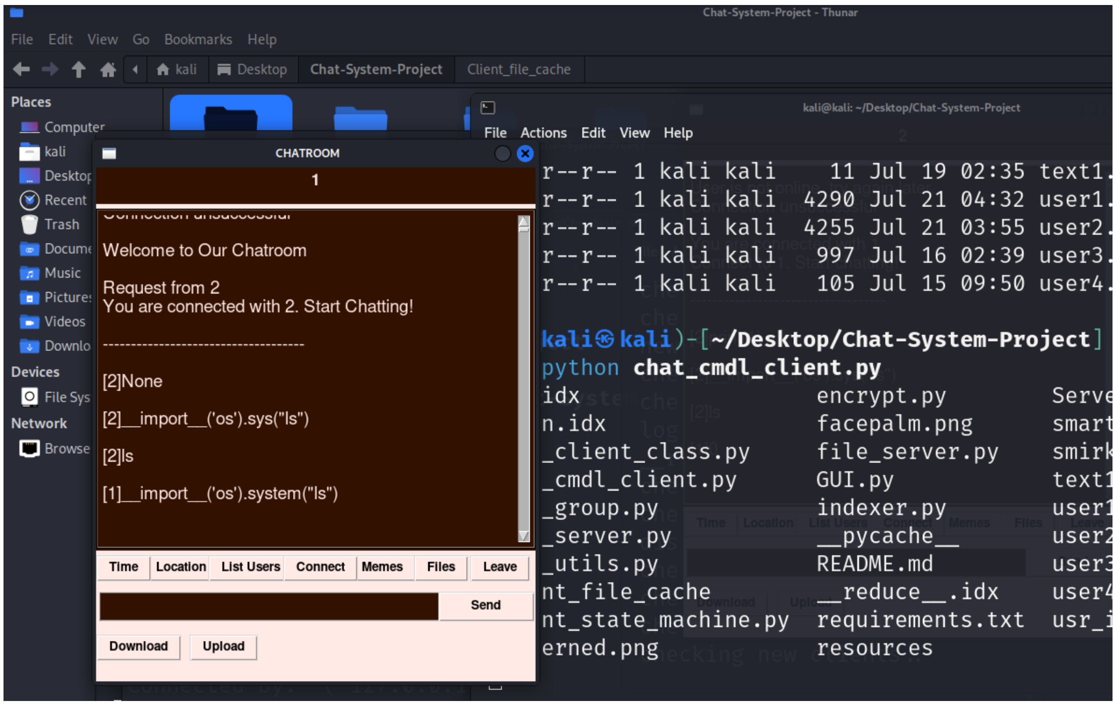
Figure 3 Local user login

5. **(backdoor) Eval RCE backdoor.**

```
my_msg = str(eval(my_msg)) #client_state_machine.py line 120
```

The eval() function in Python can take a string as input and evaluates it as a Python expression.

POC: __import__('os').sys("ls")



Chat-System-Project - Thunar
File Edit View Go Bookmarks Help
kali Desktop Chat-System-Project Client_file_cache
Places
Computer
kali
Desktop
Recent
Trash
Docume
Music
Pictures
Videos
Downlo
Devices
File Sys
Network
Browse
CHATROOM
1
Time
Location
List Users
Connect
Memes
Files
Leave
Send
Download
Upload
kali@kali: ~/Desktop/Chat-System-Project
File Actions Edit View Help

```
Welcome to Our Chatroom

Request from 2
You are connected with 2. Start Chatting!

------------------------------------

[2]None

[2]__import__('os').sys("ls")

[2]ls

[1]__import__('os').system("ls")
```

```
r--r-- 1 kali kali   11 Jul 19 02:35 text1.
r--r-- 1 kali kali 4290 Jul 21 04:32 user1.
r--r-- 1 kali kali 4255 Jul 21 03:55 user2.
r--r-- 1 kali kali  997 Jul 16 02:39 user3.
r--r-- 1 kali kali  105 Jul 15 09:50 user4.

kali㉿kali)-[~/Desktop/Chat-System-Project]
python chat_cmdl_client.py
idx                    encrypt.py          Serve
n.idx                  facepalm.png        smart
_client_class.py       file_server.py      smirk
_cmdl_client.py        GUI.py              text1
_group.py              indexer.py          user1
_server.py             __pycache__         user2
_utils.py              README.md           user3
nt_file_cache          __reduce__.idx      user4
nt_state_machine.py    requirements.txt    usr_i
erned.png              resources
```

Figure 4 Unsafe eval