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:

1. [Summary](#_0x1_Summary)
2. [Key strengths](#_0x2_Key_strengths)
3. [Issues,](#_0x2_Key_strengths) [Improvement and POC](#_0x3_Issues_and)

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

However, there are a few issues:

1. Three coding problems and two vulnerabilities were identified.
2. The configuration settings are hard-coded.

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](#_0x3_Issues_and)

1. (vulnerability) Outdated RSA encryption methods.

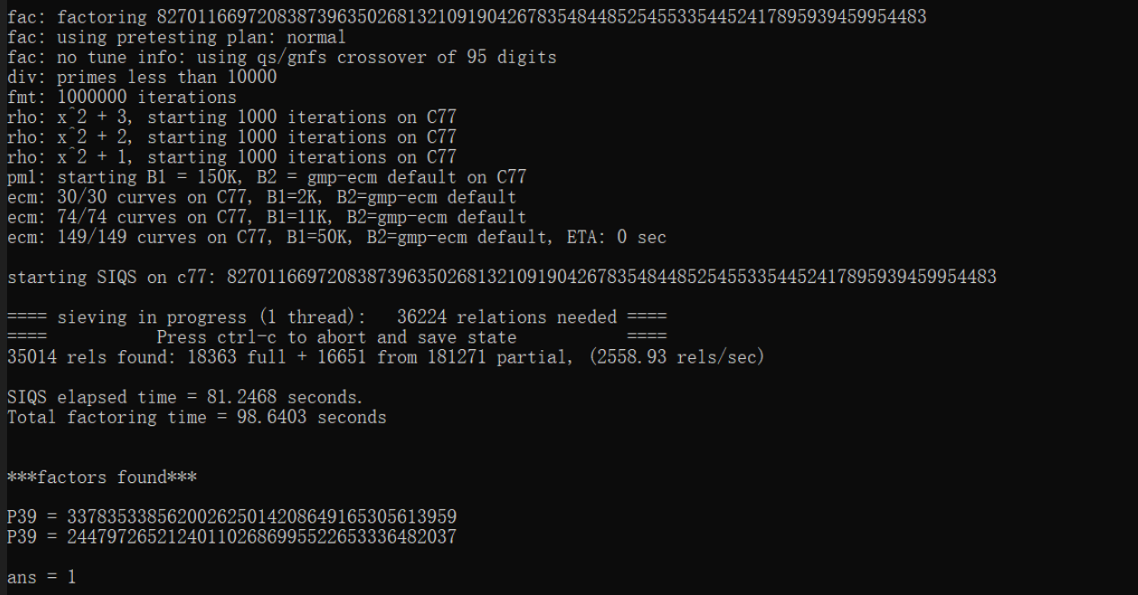
(PubKey, PrivateKey) = rsa.newkeys(512) # encrypt.py line 6

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:

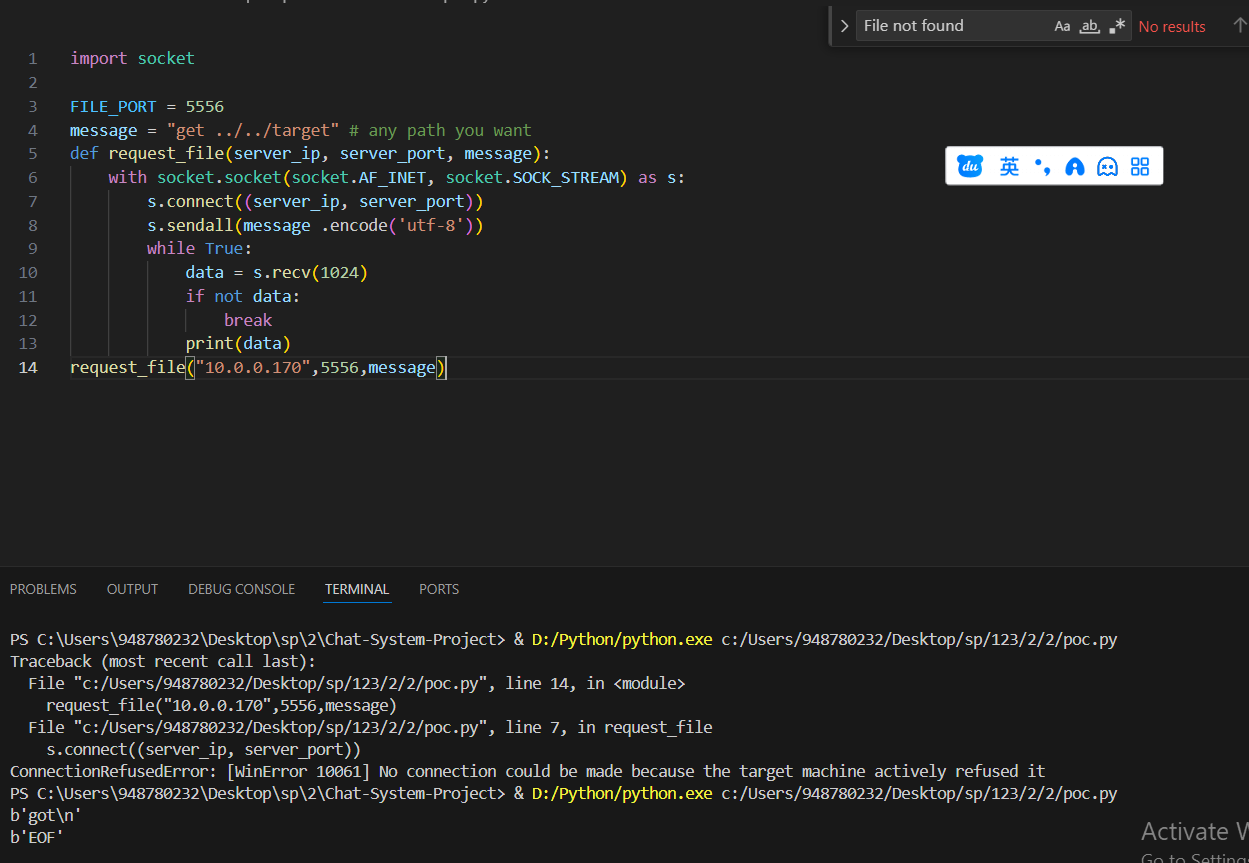


Tools: <https://github.com/bbuhrow/yafu>

1. (vulnerability) Unsafe file transfer service (file\_server.py).

The file server does not perform any checks on the input commands, leading to anyone being able to use the file server.

Improvement: Use Pipes

POC: 

1. (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

Pickle allows serialized data to contain instructions that get executed during the deserialization process. If an attacker can control the serialized data being deserialized.

Improvement: Avoid to use pickle

POC: Over write any pickle file by unsafe file transfer

A screen shot of a computer code

Description automatically generated

1. (vulnerability) Local user login

Bad design, user verification occurs locally, not on the server.

Improvement：Move it onto server.

POC：

A screen shot of a computer screen

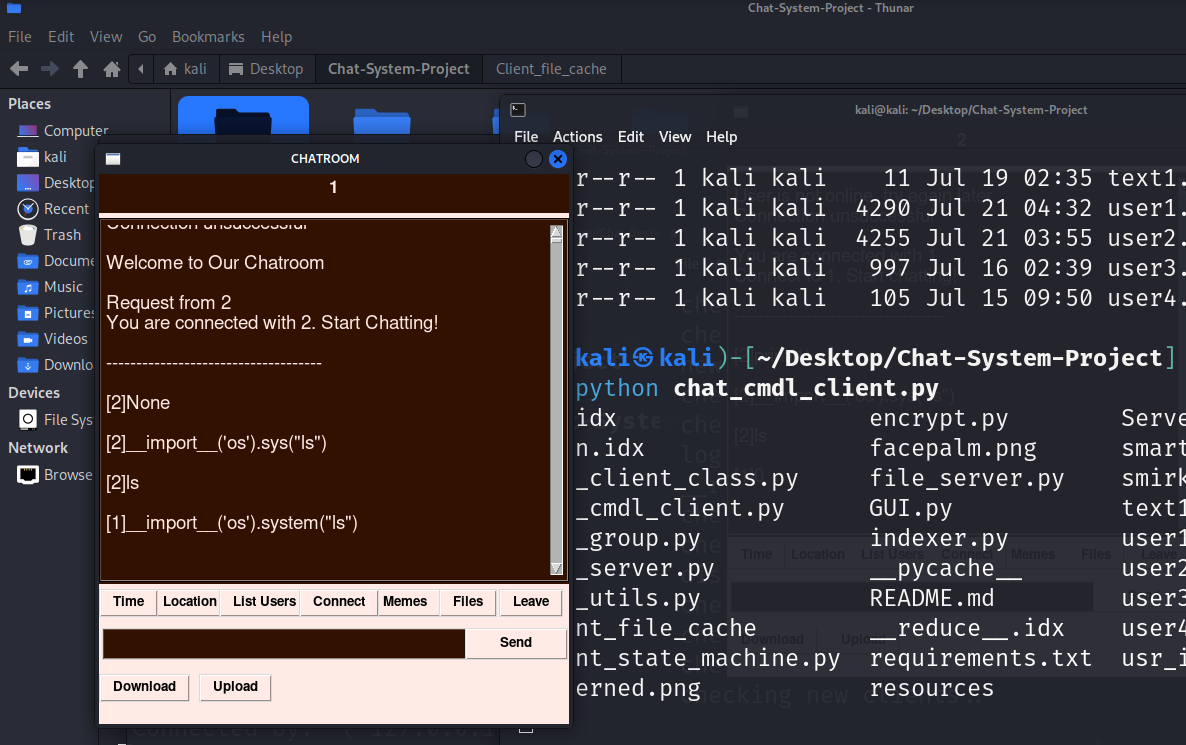
Description automatically generated

1. (vulnerability) Unsafe function used.

   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.

Improvement：

POC：

1. (issue) Configuration