Peer review feedback

0x0 Overview

Projection name: Chat App
Code reviewed: client.py, server.py
Reviewer: Zhihao Cheng
Test method: White box audit, Unit test, Dynamic test
Feedback Structure:

1. Summary
2. Key strengths
3. Issues, Improvement and POC

0x1 Summary

- ✓ Message to person
- ✓ Group message
- ✓ Peer to Peer File Transfer (only server part implemented)
- ✓ Show online list

Overall, the program followed the conception of S2S protocol. The code readability is lacking due to the absence of comments. Fortunately, the clear code structure partially compensates for this drawback.
Some issues and vulnerabilities have been identified:

1. Broken login access
2. Plaintext passwords transmission

The feedback would also provide the evidence, improvement and POC.

0X1 Program strengths

1. End to end encryption by RSA2048-OAEP
   E2EE ensures that only the communicating users can read the messages, which helps keep data privacy.
2. Configuration and optimization strategy used.
   Make the setting convenient and use optimization strategy to robust the system, including reconnection, attendance etc.
3. **Excellent error exception**.
   All exceptional cases in testing have been handled.

0X2 Vulnerability, mistakes and improvements

1. (vulnerability) Broken login access

   Python json abuse: For Python's json module, using get to access a non-existent element will return None. Therefore, an attacker only needs to construct a non-existent user and use null as the login password to bypass the login verification.

```
141         username = credentials.get("username")
142         password = credentials.get("password")
143
144         if not check_admin(username, password):
145             client_socket.sendall(b"Error: Invalid admin credentials")
146             client_socket.close()
147             return
```

Figure 1 server.py line 141 - 147

   Improvement: Check null value

```
if password == None:
    return
```

   POC: https://github.com/FrogGuaGua/SQRPRGRM/blob/main/1/1/poc.py

2. (vulnerability) Plaintext passwords transmission

   Data breach: Interceptors can directly read the passwords, leading to account theft or data breaches.

```
    self.username, self.password = self.get_credentials()
    credentials = json.dumps({"username": self.username, "password": self.password, 'pubkey': self.keys["publickey"] })
    self.client_socket.sendall(credentials.encode('utf-8'))
    response = self.client_socket.recv(1024).decode('utf-8')
    print(response)
    return response
```

Figure 2 client.py 167-173