

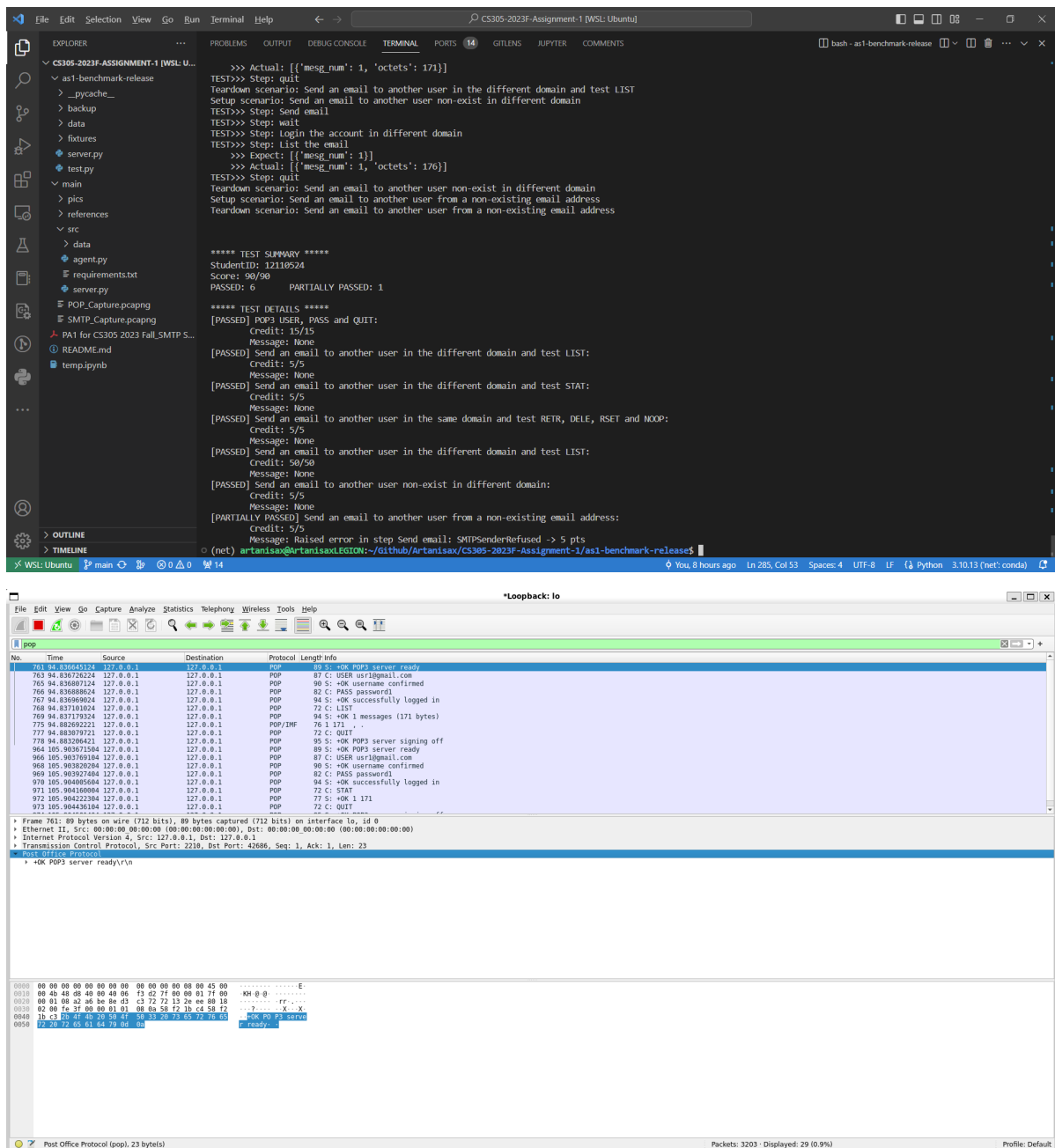
- Basic Test
- Advance
  - Error handling
  - More commands

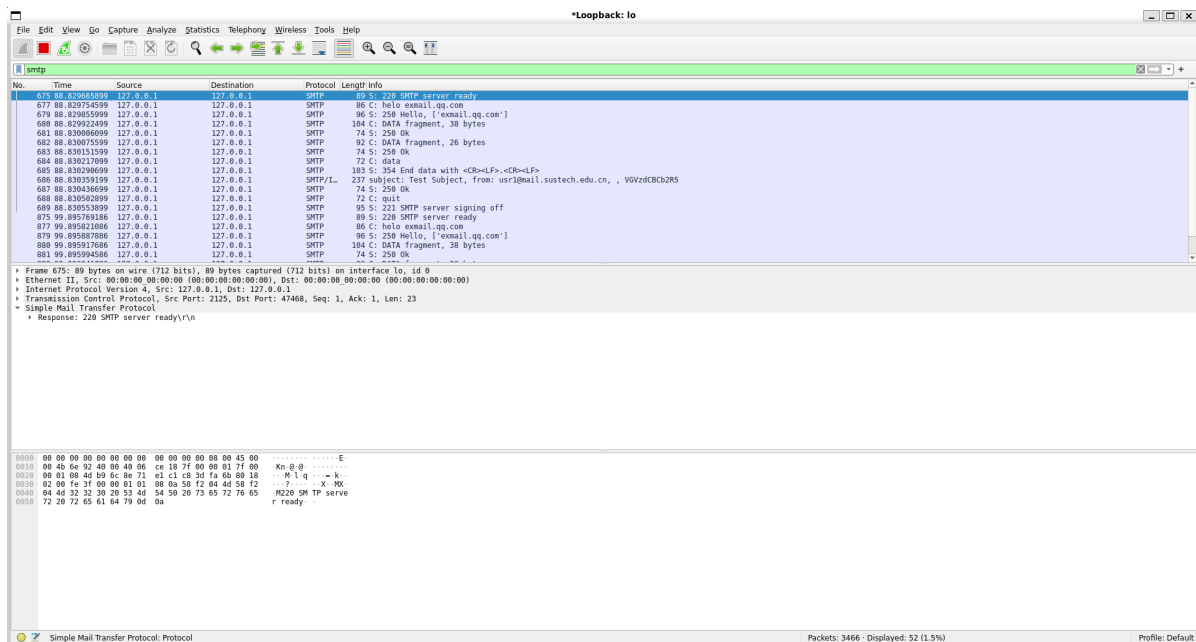
# CS305-2023Fall PA1

Name: 陈康睿

SID: 12110524

## Basic Test





# Advance

## Error handling

### 1. Connection refused

```

1  if self.login:
2      self.handle_op[cmd](args)
3  else:
4      if not self.username:
5          if cmd == 'USER':
6              self.handle_op['USER'](args)
7          else:
8              self.send('-ERR need username for login')
9      else:
10         if cmd == 'PASS':
11             self.handle_op['PASS'](args)
12         else:
13             self.send('-ERR need password for login')

```

### 2. Illegal Commands

```

1  # all commands and args are checked to some extent
2  # here are just some examples
3
4  def _RETR(self, args):
5      if len(args) != 1:
6          self.send('-ERR invalid arguments')
7          return
8
9      idx = int(args[0]) - 1
10     if idx in self.pre_del or idx < 0 or idx >= len(self.mailbox):
11         self.send(f'-ERR inexistant email')
12     else:
13         size = len(self.mailbox[idx])

```

```

14         self.send(f'+OK {size} bytes')
15         self.request.sendall(self.mailbox[idx].encode())
16         print(f'>>> {self.mailbox[idx]}')
17
18
19     def _MAIL(self, args):
20         if len(args) != 1:
21             self.send(501, 'Invalid arguments')
22         if not self.authorization or self.mail_from:
23             self.send(503, 'Bad sequence')
24
25         username = args[0][6: -1]
26         if username not in ACCOUNTS:
27             self.send(551, 'User not local')
28         else:
29             self.mail_from = username
30             self.send(250, 'ok')

```

```

artanisax@ArtanisaxLEGION: ~/Github/Artanisax/CS305-2023F-Assignment-1/main/src
(net) artanisax@ArtanisaxLEGION: ~/Github/Artanisax/CS305-2023F-Assignment-1/main/src$ python agent.py -e usr@gmail.com -p pass
[smtp|pop|exit]>>> pop
b'+OK POP3 server ready'
b'+OK username confirmed'
b'+OK successfully logged in'
[pop]>>> retr -1
-ERR!!
error_proto(b'-ERR inexistant email')
[pop]>>> retr 114514
-ERR!!
error_proto(b'-ERR inexistant email')
[pop]>>>

```

### 3. Try-Except Handling

```

1  # almost every where in the server source code
2  conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3  for server in outsiders:
4      try:
5          host = 'localhost'
6          port = int(fdns_query(server, 'P'))
7          conn.connect((host, port))
8          assert conn.recv(1024).strip().decode().startswith('220')
9
10         conn.sendall(f'hello {args.name}\r\n'.encode())
11         assert conn.recv(1024).strip().decode().startswith('250')
12
13         conn.sendall(f'dema FROM:<{self.mail_from}>\r\n'.encode())
14         assert conn.recv(1024).strip().decode().startswith('250')
15
16         for rcpt in outsiders[server]:
17             conn.sendall(f'derc TO:<{rcpt}>\r\n'.encode())

```

```

18         data = conn.recv(1024).strip().decode()
19         if data.startswith('250'):
20             continue
21         elif data.startswith('551'):
22             flag = True
23         else:
24             raise AssertionError
25
26     conn.sendall(b'data\r\n')
27     assert conn.recv(1024).strip().decode().startswith('354')
28
29     conn.sendall(self.data_content.encode())
30     assert conn.recv(1024).strip().decode().startswith('250')
31
32     conn.sendall(b'quit\r\n')
33     assert conn.recv(1024).strip().decode().startswith('221')
34 except AssertionError as e:
35     self.send(-1, 'An error occurred when sending emails')
36 finally:
37     conn.close()

```

## More commands

- **HELP**

```

1  def __init__(...):
2      ...
3      self.handle_op = {
4          ...
5          'HELP': self._HELP,
6      }
7      ...
8
9  def _HELP(self, args):
10     if len(args) > 0:
11         self.send('-ERR invalid arguments')
12         return
13
14     self.send('+OK')
15     self.send('USER <username>')
16     self.send('PASS <password>')
17     self.send('STAT')
18     self.send('LIST [index]')
19     self.send('RETR <index>')
20     self.send('DELE <index>')
21     self.send('RSET')
22     self.send('NOOP')
23     self.send('QUIT')
24     self.send('HELP')
25     self.send('.')

```

```
artanisax@ArtanisaxLEGION: ~/Github/Artanisax/CS305-2023F-Assignment-1/main/src
(net) artanisax@ArtanisaxLEGION:~/Github/Artanisax/CS305-2023F-Assignment-1/main/src$ python
agent.py -e usr1@mail.sustech.edu.cn -p pass1
[smtp|pop|exit]>>> pop
b'+OK POP3 server ready'
b'+OK username confirmed'
b'+OK successfully logged in'
[pop]>>> help
(b'+OK', [b'USER <username>', b'PASS <password>', b'STAT', b'LIST [index]', b'RETR <index>',
b'DELE <index>', b'RSET', b'NOOP', b'QUIT', b'HELP'], 106)
[pop]>>> _
```