CMPT 361: Group Project
Group members: Brian Hu, Haris K, Mitch Duriez

Preset up: It is assumed that the user will run key_generator.py before running any client or server application

Once key_generator is done, the client and server program has no issues generating the symmetric key. Checks are in place to make sure no invalid user can access the server and send emails.

Choice 1:Send Email Protocol

- Client and server use the symmetric key produced to encrypt and decrypt messages.
- The client program ensures that the user has provided recipients of the email.
- The client program ensures that the user doesn’t send an empty title, as well as the title must be 100 characters or less. They will be reasked for a title if empty
- The client program ensures that the user doesn’t send an email with contents greater than 1000000.
- The server program handles spaces in the title by converting spaces to underscores when creating filenames.
- The program assumes that the user will not send an email with the same title.
- Server saves the email to each recipient's folder in the server folder as a .txt file.
- Each text file saved contains all the information that the user had supplied along with time and content length that the server program calculates.
- For typing a message, the client will be asked to retype the contents if too small or empty, but if the client wants to add a file, if contents are empty or too big it will ask the user to change their file, and go back to the main menu.

Choice 2: Viewing Inbox

- Server ensures that the client and server do not crash if the inbox is empty. Handled by just printing column names with nothing below it to the client.
- Server searches for an email folder with glob using the client's username.
- Server assumes the inside of the email follows a specific format in order to strip the data such as the source sender, date and time sent, and title of the email. Will not crash if invalid format, however, it won’t grab information properly. May crash if the email is completely empty.
- The inbox is by default sorted from oldest to newest emails and the index gets rearranged to match them as well. User is unable to adjust how it gets sorted.
- Inbox data is all stored in a list of lists with index, source sender, date and time, and title of the email.
- Client sends an “OK” acknowledgement to the server once the inbox message has been received. Does not check if the entire message has been received properly.

Choice 3: Viewing email

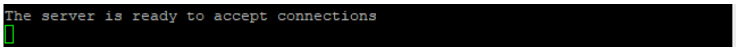
- Server ensures no crash if the index is out of range but assumes it is of integer value.
- Server sends a prompt to the client asking for an index.
- Client sends the user input back to the server.
- Server gets the index and gets the list of email info through a helper function.
- Server checks if the index is in range or if the email info list is empty. If the email list is empty or if the index is out of range, the server sends an error message to the client. If the index is in range and the list has email info in it, the server continues.
- Server calls a helper function to build the file name from the email info selected with index and the client's username. Helper function opens the file with the assumption it exists and reads the contents. Helper function returns the contents and file size.
- Server sends the file size over to the client first.
- Client sends "OK" if it receives the file size or prints a message if the error message is sent.
- Server then sends all content as a binary string after it receives the ok.
- Client receives the file contents and prints the contents that were sent over.
- After the main process is finished, Client sends an ok message to the server to continue.
- Server receives the ok and continues.

Choice 4: Terminate

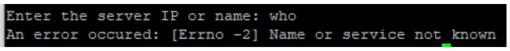
- The client program ends its connection to the server and closes properly, server program continues to run as it is persistent.

Tests:

Invalid IP does not crash server:

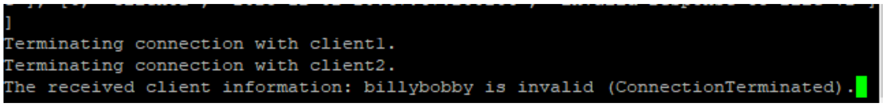


```
The server is ready to accept connections
```

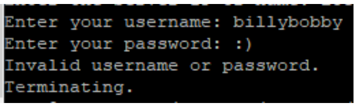


```
Enter the server IP or name: who
An error occured: [Errno -2] Name or service not known
```

Invalid username and password:

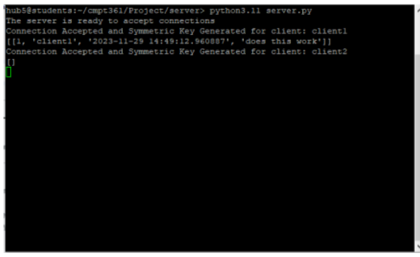


```
]
Terminating connection with client1.
Terminating connection with client2.
The received client information: billybobby is invalid (ConnectionTerminated).
```

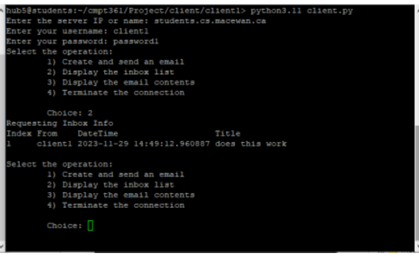


```
Enter your username: billybobby
Enter your password: :)
Invalid username or password.
Terminating.
```

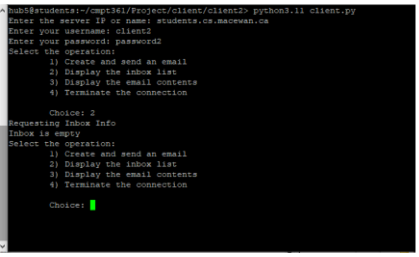
Testing for multiple connections to the server and also tests view inbox for empty inbox and inbox with 1 item:



```
hub5@students:~/cmpt361/Project/server> python3.11 server.py
The server is ready to accept connections
Connection Accepted and Symmetric Key Generated for client: client1
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work']]
Connection Accepted and Symmetric Key Generated for client: client2
[]
```

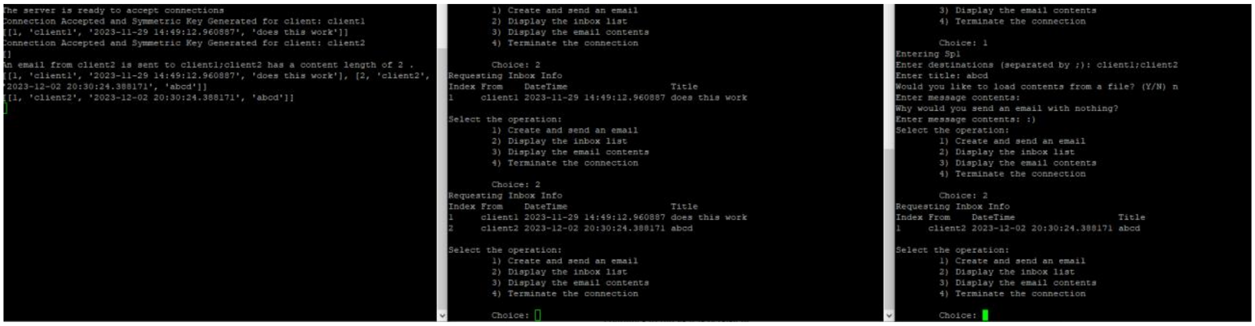


```
hub5@students:~/cmpt361/Project/client/client1> python3.11 client.py
Enter the server IP or name: students.cs.macewan.ca
Enter your username: client1
Enter your password: password1
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                   Title
1     client1 2023-11-29 14:49:12.960887 does this work

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```



```
hub5@students:~/cmpt361/Project/client/client2> python3.11 client.py
Enter the server IP or name: students.cs.macewan.ca
Enter your username: client2
Enter your password: password2
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Inbox is empty
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

Sending an empty email, sending mail with no file to self and other clients, and sorting mail by time:

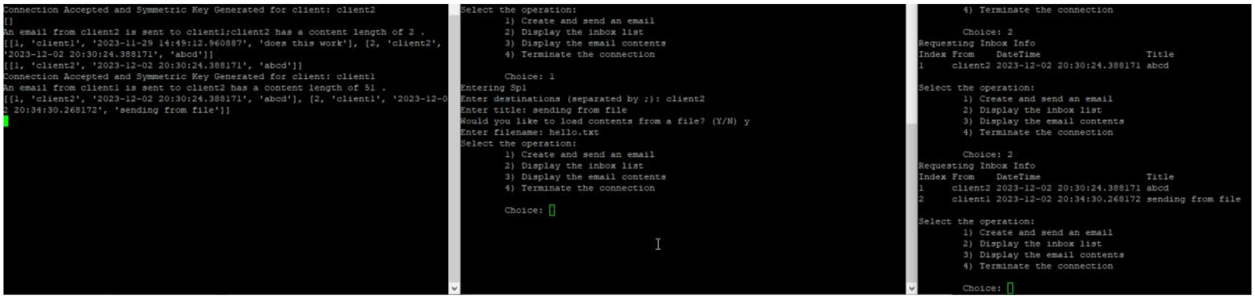


```
The server is ready to accept connections
Connection Accepted and Symmetric Key Generated for client: client1
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work']]
Connection Accepted and Symmetric Key Generated for client: client2
[]
An email from client2 is sent to client1;client2 has a content length of 2 .
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work'], [2, 'client2',
'2023-12-02 20:30:24.388171', 'abcd']]
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd']]
```

```
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client1 2023-11-29 14:49:12.960887 does this work

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client1 2023-11-29 14:49:12.960887 does this work
2     client2 2023-12-02 20:30:24.388171 abcd

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

```
        3) Display the email contents
        4) Terminate the connection

        Choice: 1
Entering Spl
Enter destinations (separated by ;): client1;client2
Enter title: abcd
Would you like to load contents from a file? (Y/N) n
Enter message contents:
Why would you send an email with nothing?
Enter message contents: :)
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

Sending a file to other clients

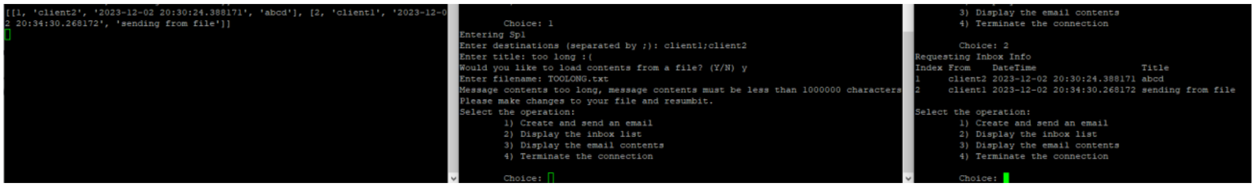


```
Connection Accepted and Symmetric Key Generated for client: client2
[]
An email from client2 is sent to client1;client2 has a content length of 2 .
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work'], [2, 'client2',
'2023-12-02 20:30:24.388171', 'abcd']]
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd']]
Connection Accepted and Symmetric Key Generated for client: client1
An email from client1 is sent to client2 has a content length of 51 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file']]
```

```
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 1
Entering Spl
Enter destinations (separated by ;): client2
Enter title: sending from file
Would you like to load contents from a file? (Y/N) y
Enter filename: hello.txt
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

```
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

Sending an email that is too long

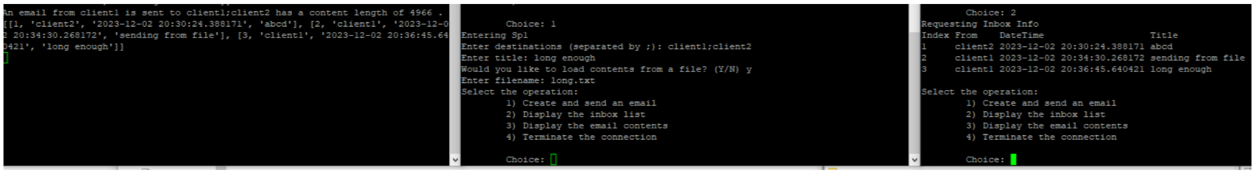


```
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file']]
```

```
        Choice: 1
Entering Spl
Enter destinations (separated by ;): client1;client2
Enter title: too long :(
Would you like to load contents from a file? (Y/N) y
Enter filename: TOOLONG.txt
Message contents too long, message contents must be less than 1000000 characters
Please make changes to your file and resumbit.
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

```
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

Sending a file that is long but not too long

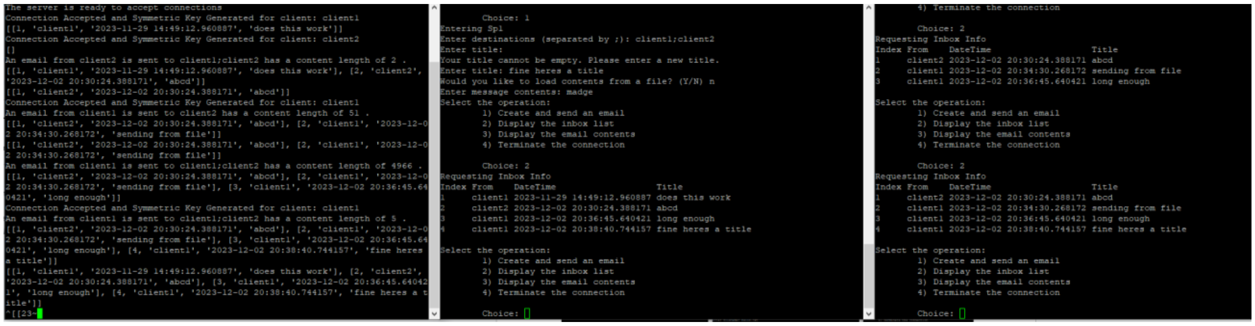


```
An email from client1 is sent to client1;client2 has a content length of 4966 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough']]
```

```
        Choice: 1
Entering Spl
Enter destinations (separated by ;): client1;client2
Enter title: long enough
Would you like to load contents from a file? (Y/N) y
Enter filename: long.txt
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

```
        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file
3     client1 2023-12-02 20:36:45.640421 long enough

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

Invalid Title error checking

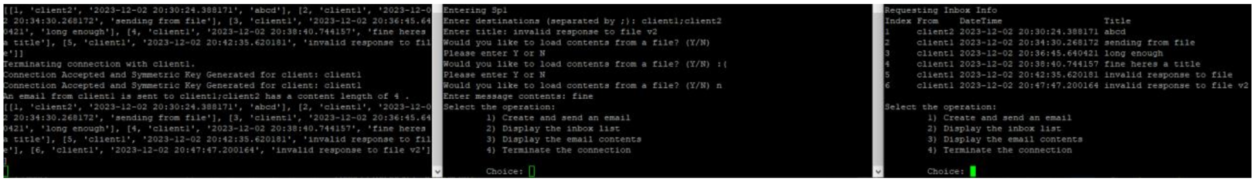


```
The server is ready to accept connections
Connection Accepted and Symmetric Key Generated for client: client1
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work']]
Connection Accepted and Symmetric Key Generated for client: client2
[]
An email from client2 is sent to client1;client2 has a content length of 2 .
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work'], [2, 'client2',
'2023-12-02 20:30:24.388171', 'abcd']]
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd']]
Connection Accepted and Symmetric Key Generated for client: client1
An email from client1 is sent to client2 has a content length of 51 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file']]
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file']]
An email from client1 is sent to client1;client2 has a content length of 4966 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough']]
Connection Accepted and Symmetric Key Generated for client: client1
An email from client1 is sent to client1;client2 has a content length of 5 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
a title']]
[[1, 'client1', '2023-11-29 14:49:12.960887', 'does this work'], [2, 'client2',
'2023-12-02 20:30:24.388171', 'abcd'], [3, 'client1', '2023-12-02 20:36:45.64042
1', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres a t
itle']]
^[[23~
```

```
        Choice: 1
Entering Spl
Enter destinations (separated by ;): client1;client2
Enter title:
Your title cannot be empty. Please enter a new title.
Enter title: fine heres a title
Would you like to load contents from a file? (Y/N) n
Enter message contents: madge
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client1 2023-11-29 14:49:12.960887 does this work
2     client2 2023-12-02 20:30:24.388171 abcd
3     client1 2023-12-02 20:36:45.640421 long enough
4     client1 2023-12-02 20:38:40.744157 fine heres a title

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

```
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file
3     client1 2023-12-02 20:36:45.640421 long enough

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 2
Requesting Inbox Info
Index From    DateTime                      Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file
3     client1 2023-12-02 20:36:45.640421 long enough
4     client1 2023-12-02 20:38:40.744157 fine heres a title

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice:
```

Invalid response to file choice checking

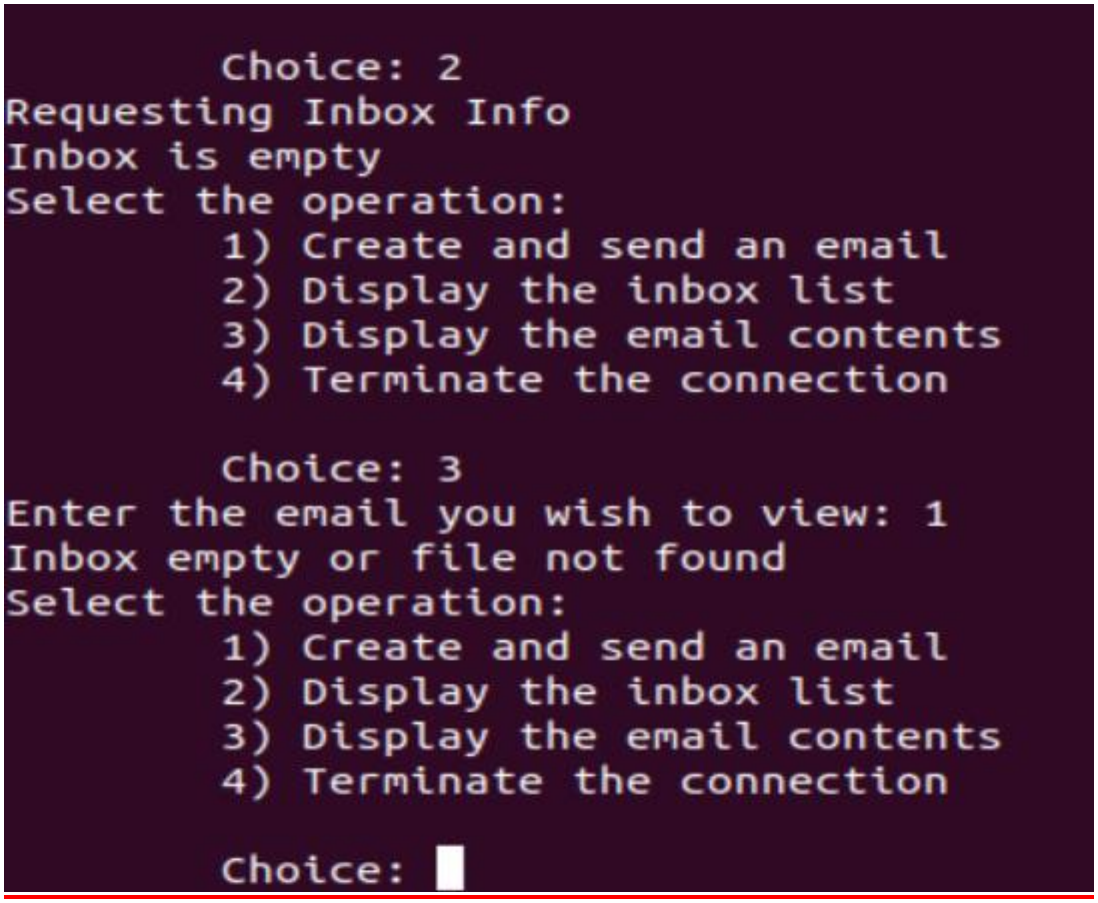


```
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
a title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
e']]
Terminating connection with client1.
Connection Accepted and Symmetric Key Generated for client: client1
Connection Accepted and Symmetric Key Generated for client: client1
An email from client1 is sent to client1;client2 has a content length of 4 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
a title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
e'], [6, 'client1', '2023-12-02 20:47:47.200164', 'invalid response to file v2']
]
```

```
Entering Sp1
Enter destinations (separated by ;): client1;client2
Enter title: invalid response to file v2
Would you like to load contents from a file? (Y/N)
Please enter Y or N
Would you like to load contents from a file? (Y/N) :(
Please enter Y or N
Would you like to load contents from a file? (Y/N) n
Enter message contents: fine
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

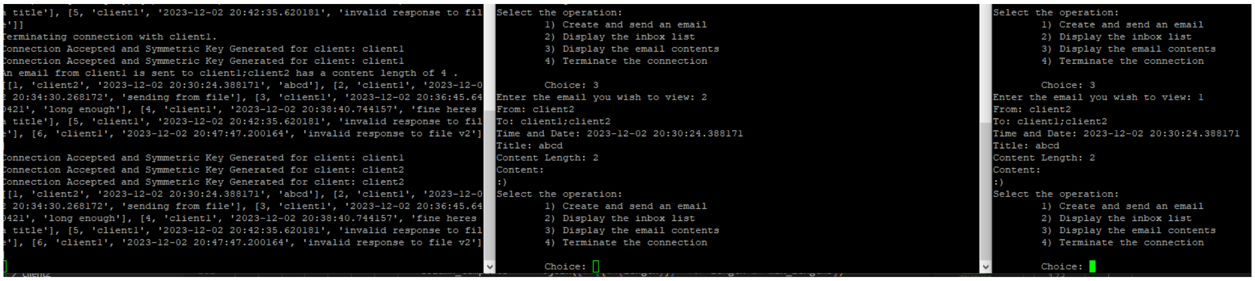
```
Requesting Inbox Info
Index From    DateTime                   Title
1     client2 2023-12-02 20:30:24.388171 abcd
2     client1 2023-12-02 20:34:30.268172 sending from file
3     client1 2023-12-02 20:36:45.640421 long enough
4     client1 2023-12-02 20:38:40.744157 fine heres a title
5     client1 2023-12-02 20:42:35.620181 invalid response to file
6     client1 2023-12-02 20:47:47.200164 invalid response to file v2

Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

Invalid Email Index Error Checking:



```
        Choice: 2
Requesting Inbox Info
Inbox is empty
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 3
Enter the email you wish to view: 1
Inbox empty or file not found
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

Viewing email contents:

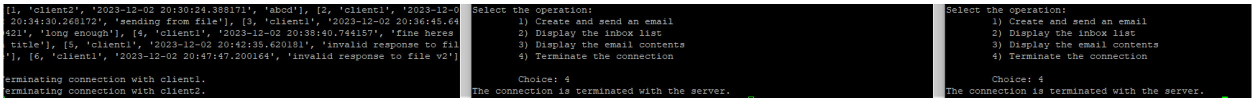


```
a title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
e']]
Terminating connection with client1.
Connection Accepted and Symmetric Key Generated for client: client1
Connection Accepted and Symmetric Key Generated for client: client1
An email from client1 is sent to client1;client2 has a content length of 4 .
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
a title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
e'], [6, 'client1', '2023-12-02 20:47:47.200164', 'invalid response to file v2']
]
Connection Accepted and Symmetric Key Generated for client: client1
Connection Accepted and Symmetric Key Generated for client: client2
Connection Accepted and Symmetric Key Generated for client: client2
[[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
2 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
0421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
a title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
e'], [6, 'client1', '2023-12-02 20:47:47.200164', 'invalid response to file v2']
]
```

```
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 3
Enter the email you wish to view: 2
From: client2
To: client1;client2
Time and Date: 2023-12-02 20:30:24.388171
Title: abcd
Content Length: 2
Content:
:)
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

```
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 3
Enter the email you wish to view: 1
From: client2
To: client1;client2
Time and Date: 2023-12-02 20:30:24.388171
Title: abcd
Content Length: 2
Content:
:)
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 
```

Persistent server after terminating connection:



```
[1, 'client2', '2023-12-02 20:30:24.388171', 'abcd'], [2, 'client1', '2023-12-0
 20:34:30.268172', 'sending from file'], [3, 'client1', '2023-12-02 20:36:45.64
421', 'long enough'], [4, 'client1', '2023-12-02 20:38:40.744157', 'fine heres
 title'], [5, 'client1', '2023-12-02 20:42:35.620181', 'invalid response to fil
'], [6, 'client1', '2023-12-02 20:47:47.200164', 'invalid response to file v2']

erminating connection with client1.
erminating connection with client2.
```

```
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 4
The connection is terminated with the server.
```

```
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 4
The connection is terminated with the server.
```

Protocol analysis and Enhancement (Section V):

Attack Type: Replay Attack

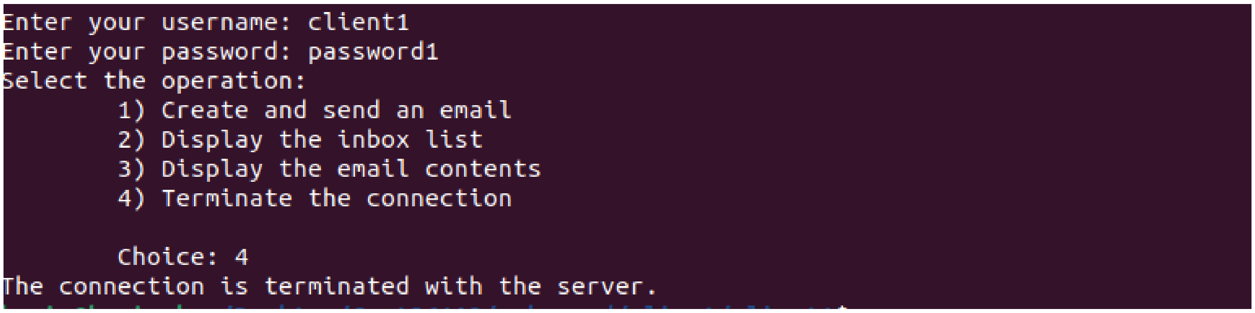
An attacker can store the packets sent from the client that contain username and password. With these packets, the attacker can send a valid username and password and start communication with the server. This can take up spots which are ment for valid clients and prevent access to the server.

Solution: Apon connecting to the server, both client and server create timestamps, when the client sends username and password, the password has the client's timestamp appended to it. The server will split the password and timestamp. Then the server will first compare the client timestamp to the server's timestamp, and if it is found that the client timestamp is too old, it will terminate the connection.

Enhanced Version Tests:

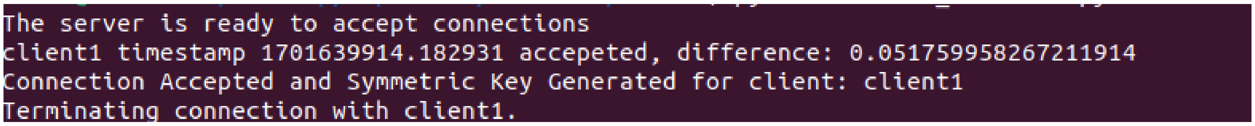
Real client login attempt

Client:



```
Enter your username: client1
Enter your password: password1
Select the operation:
        1) Create and send an email
        2) Display the inbox list
        3) Display the email contents
        4) Terminate the connection

        Choice: 4
The connection is terminated with the server.
```

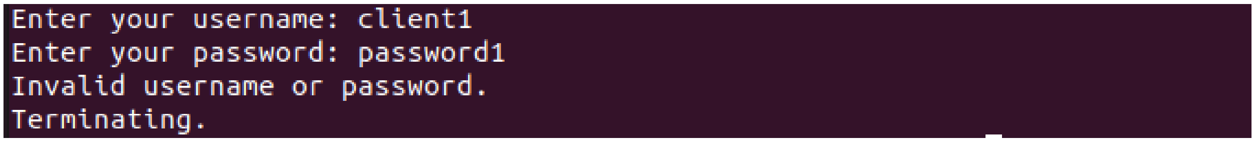
Server:



```
The server is ready to accept connections
client1 timestamp 1701639914.182931 accepeted, difference: 0.051759958267211914
Connection Accepted and Symmetric Key Generated for client: client1
Terminating connection with client1.
```

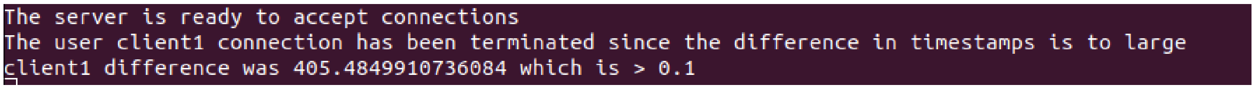
Simulate Attacker replay attempt by using old time stamp:

client:



```
Enter your username: client1
Enter your password: password1
Invalid username or password.
Terminating.
```

server:



```
The server is ready to accept connections
The user client1 connection has been terminated since the difference in timestamps is to large
client1 difference was 405.4849910736084 which is > 0.1
```