FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IMAP Client with TLS Support

Project Documentation

Autor: Slabik Yaroslav

Login: xslabi01

Obsah

---

1. Introduction

This documentation describes an IMAP client named **imapcl**, created as part of the ISA subject project at the Faculty of Information Technology, VUT in Brno. The project's goal was to implement a client for the IMAP4rev1 protocol (RFC 3501), which allows downloading electronic mail from a specified server and saving messages to a designated directory. The client supports secure connections using SSL/TLS, user authentication, selection of various mailboxes, and the ability to download only new messages or only their headers.

2. Project Description

The **imapcl** program has the following functionalities:

- **Connecting to an IMAP Server:** Connects to a specified IMAP server using the IMAP4rev1 protocol.
- **User Authentication:** Performs user login using provided credentials.

- **Downloading Messages:** Downloads messages from a specified mailbox (default is "INBOX") and saves them to a specified directory.
- **Displaying Message Count:** Outputs information about the number of downloaded messages to the standard output.
- **Parameter Support:** Allows setting additional parameters to change functionality, such as using TLS, selecting a mailbox, choosing a port, downloading only new messages, or only message headers.

3. Application Design

3.1 Program Architecture

The program was designed modularly with an emphasis on clarity and ease of maintenance. The main components of the program are:

- **Command-Line Argument Parsers:** Processes input parameters and sets the program's behavior.
- **Network Module:** Ensures establishing a connection with the server and communication using the IMAP protocol.
- **Authentication Module:** Performs user login to the server.
- **Message Processing Module:** Handles downloading, processing, and saving messages.
- **Local Message Index Management Module:** Responsible for reading, updating, and managing the UID index of messages.
- **Message Decoding Module:** Processes message encodings such as Base64 and Quoted-Printable.
- **SSL/TLS Module:** Ensures secure connection with the server.

3.2 Code Structure

The code is divided into several files and functions, each performing a specific task:

- **main.cpp:** Contains the main function of the program, processes arguments, and orchestrates the main steps (connection, authentication, message downloading).
- **imap_client.cpp / imap_client.h:** Implements functions for communicating with the IMAP server, processing messages, and decoding.
    - **generate_tag**: Generates a unique tag for IMAP commands.
    - **ssl_read**: Function for reading data.
    - **ssl_write**: Function for writing data.
    - **connect_to_server**: Establishes a connection with the server, sets up SSL/TLS if required.
    - **send_command**: Sends an IMAP command to the server and receives a response.
    - **read_line**: Reads one line from the server's response.

  - **read_literal**: Reads a data block of specified size from the server, used for loading message content or other large data blocks.
  - **login**: Performs user authentication.
  - **select_mailbox**: Selects a specified mailbox on the server.
  - **read_local_index**: Reads the local UID index from the **index.txt** file.
  - **update_local_index**: Updates the local UID index in the **index.txt** file.
  - **search_unseen_messages**: Searches for unseen messages in the selected mailbox.
  - **save_message**: Saves individual messages to files.
  - **fetch_messages**: Downloads messages and saves them to a directory.
  - **read_credentials**, **directory_exists**: Helper functions for working with files and directories.
  - Decoding Functions**: base64_decode, decode_quoted_printable, decode_encoded_word.**
- **ssl_utils.cpp / ssl_utils.h:** Implements functions for working with SSL/TLS.
  - **initialize_ssl**: Initializes the OpenSSL library.
  - **create_context**: Creates an SSL context.
  - **configure_ssl_context**: Configures the SSL context with certificates.
  - **cleanup_ssl**: Cleans up the SSL context and releases resources.

4. Implementation Description

4.1 Command-Line Argument Processing

The program uses the **getopt** library for processing command-line parameters.

- **server**: Required argument specifying the IP address or domain name of the IMAP server.
- **-p port**: Specifies the server's port number. The default value is 143 for unencrypted connections and 993 when using TLS.
- **-T**: Enables encryption using SSL/TLS.
- **-c certfile**: Specifies the certificate file for server verification.
- **-C certaddr**: Specifies the directory containing certificates (default: **/etc/ssl/certs**).
- **-n**: Downloads only new messages.
- **-h**: Downloads only message headers.
- **-a auth_file**: Required parameter specifying the path to the file with authentication credentials.
- **-b MAILBOX**: Specifies the name of the mailbox on the server (default: **INBOX**)
- **-o out_dir**: Required parameter specifying the output directory for saving downloaded messages.

4.2 Establishing Connection with Server

The **connect_to_server** function ensures establishing a connection with the IMAP server. The process includes:

1. **Retrieving Server Information:** Uses the **getaddrinfo** function to obtain server information based on the provided IP address or domain name.
2. **Creating a Socket:** Creates a TCP socket for communication with the server.
3. **Establishing TCP Connection:** Connects to the server on the specified port.
4. **Initializing SSL/TLS (if required):**
    - 4.1. Creates a new SSL object and attaches it to the socket.
    - 4.2. Establishes an SSL connection using **SSL_connect**.
    - 4.3. Verifies the server's certificate validity.
    - 4.4. If verification fails, the program outputs an error message and terminates.

4.3 User Authentication

The **login** function performs user authentication using the LOGIN command. The process includes:

1. **Receiving Welcome Message:** Receives the welcome message from the server after establishing the connection.
2. **Generating a Unique Tag:** Creates a unique tag for the IMAP command.
3. **Sending Login Command:** Sends the LOGIN command along with the username and password.
4. **Processing Response:** Checks if the authentication was successful based on the server's response.

4.4 Selecting Mailbox

The **select_mailbox** function allows selecting a specified mailbox on the server using the **SELECT** command. The process includes:

1. **Sending SELECT Command:** Selects the desired mailbox (e.g., INBOX, Sent, Trash).
2. **Retrieving List of Message UIDs:** After successfully selecting the mailbox, retrieves the unique IDs of all messages in the mailbox using the **UID SEARCH ALL** command.
3. **Storing UIDs in a Vector:** Stores the UIDs in a vector for subsequent processing.

4.5 Downloading and Saving Messages

The **fetch_messages** function ensures downloading messages from the server and saving them to the specified directory. The process includes:

1. **Reading Local Index:** Loads the local index of downloaded messages from **index.txt** in the output directory.
2. **Determining Messages to Download:** Based on the **-n** and **-h** parameters, determines which messages need to be downloaded (all or only new, headers or full messages).
3. **Sending FETCH Commands:** For each message, sends a UID FETCH command to download the requested parts of the message.
4. **Processing Responses:** Reads the server's responses, decodes the messages (e.g., Base64, Quoted-Printable), and saves them to files in RFC 5322 format.
5. **Updating Local Index:** After successfully downloading messages, updates **index.txt** with the new UIDs.

4.6 Working with SSL/TLS Certificates

The **configure_ssl_context** function and others in the **ssl_utils** module ensure proper handling of SSL/TLS certificates:

1. **Loading Certificates:** The program loads certificates from the file specified by **-c certfile** or the directory specified by **-C certaddr**. If not specified, it uses the default system certificates.
2. **Verifying Server Certificate:** After establishing an SSL connection, verifies the server's certificate validity using OpenSSL functions.
3. **Setting Verification Mode:** The SSL context is set to verify the server's certificate (**SSL_VERIFY_PEER**).

5. Solution Description

5.1 Content Synchronization Logic

The program synchronizes the content of the output folder with the server's content as follows:

- **Switching to a Different Mailbox:** When selecting a new mailbox, messages from the previous mailbox are removed from the output folder. This ensures that the folder's content always corresponds to the currently selected mailbox.

- **Reusing the Program in the Same Mailbox:** If the program is run again in the same mailbox, it displays the message "No changes in the 'mailbox' mailbox." indicating that the content on the server and in the output folder is fully synchronized, and no changes are necessary. Once a new message arrives, the program downloads the missing messages and updates the **index.txt** file with the new message index. If the **-h** flag is specified, the program downloads only the headers of all messages from the server. This logic conserves bandwidth and allows users to fully synchronize the server's content with the output folder.

This logic works thanks to the auxiliary **index.txt** file, which stores the UIDs of messages and their type (header/full message) based on the selected flags. The file is located in the output folder for quick access and easy viewing.

5.2 Management of the "Unseen" Flag

After downloading an unseen message, the client resets the "unseen" (**UNSEEN**) flag on the server. This step aims to synchronize the message's state, as downloading a message may indicate that the user has read it. Although it does not precisely determine if the user has actually read the message, resetting the flag helps avoid repeatedly downloading already processed messages.

6. Usage Guide

6.1 Format of the Authentication Credentials File

The authentication credentials file (**auth_file**) must be in plain text format and contain the following entries, each on a separate line:

```
username = uživatelské_jméno
password = heslo
```

6.2 Program Execution Examples

1. Download all messages without TLS:

   ```
   ./imapcl server -a auth.txt -o zpravy
   ```

2. Download new messages with TLS and certificate:

   ```
   ./imapcl server -T -c cert.pem -n -a auth.txt -o zpravy
   ```

3. Download message headers:

   ```
   ./imapcl server -h -a auth.txt -o -h
   ```

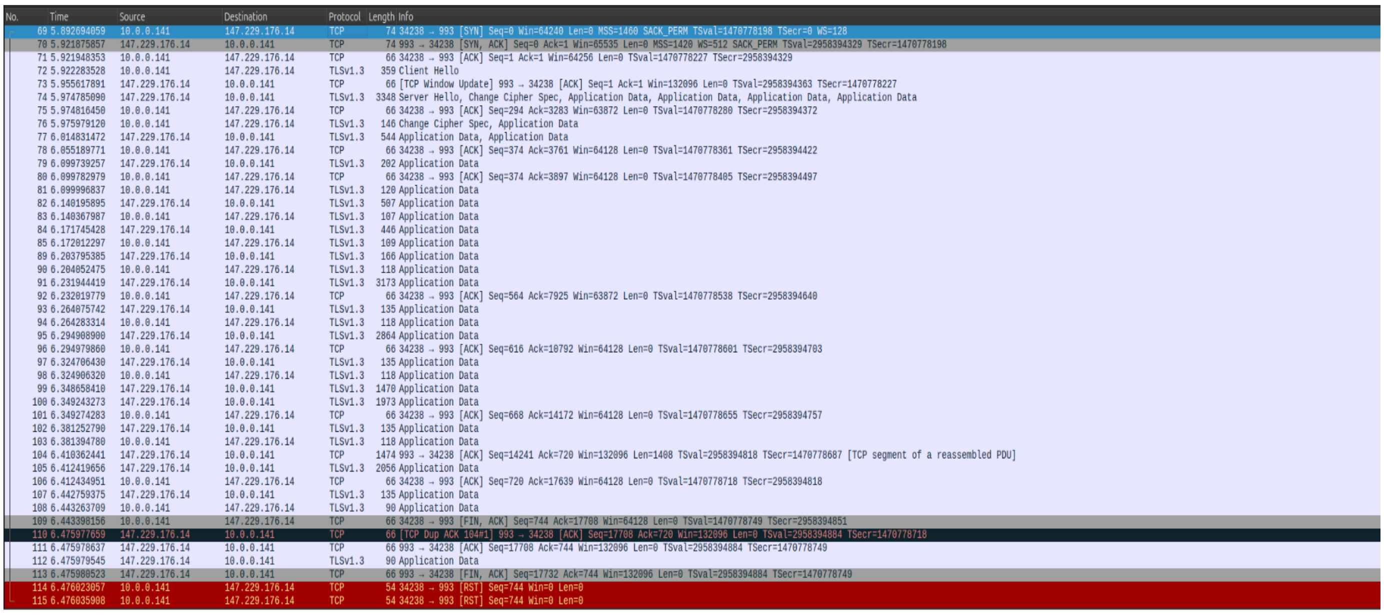
7. Application Testing

7.1 Description of Performed Tests and Test Results

1. **Test Connection with and without TLS:**
   1.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt*
   Output: ***Staženo 9 zpráv ze schránky INBOX.***

   1.2. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -T*
   Output: ***Staženo 9 zpráv ze schránky INBOX.***

   Wireshark Screenshot:



```
No.  Time        Source          Destination     Protocol Length Info
 69 5.892694059 10.0.0.141      147.229.176.14  TCP        74 34238 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1470778198 TSecr=0 WS=128
 70 5.921875857 147.229.176.14  10.0.0.141      TCP        74 993 → 34238 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 WS=512 SACK_PERM TSval=2958394329 TSecr=1470778198
 71 5.921948353 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1470778227 TSecr=2958394329
 72 5.922283528 10.0.0.141      147.229.176.14  TLSv1.3   359 Client Hello
 73 5.955617891 147.229.176.14  10.0.0.141      TCP        66 [TCP Window Update] 993 → 34238 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=2958394363 TSecr=1470778227
 74 5.974785090 147.229.176.14  10.0.0.141      TLSv1.3  3348 Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data, Application Data
 75 5.974816450 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=294 Ack=3283 Win=63872 Len=0 TSval=1470778280 TSecr=2958394372
 76 5.975979120 10.0.0.141      147.229.176.14  TLSv1.3   146 Change Cipher Spec, Application Data
 77 6.014831472 147.229.176.14  10.0.0.141      TLSv1.3   544 Application Data, Application Data
 78 6.055189771 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=374 Ack=3761 Win=64128 Len=0 TSval=1470778361 TSecr=2958394422
 79 6.099739257 147.229.176.14  10.0.0.141      TLSv1.3   202 Application Data
 80 6.099782979 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=374 Ack=3897 Win=64128 Len=0 TSval=1470778405 TSecr=2958394497
 81 6.099996837 10.0.0.141      147.229.176.14  TLSv1.3   120 Application Data
 82 6.140195895 147.229.176.14  10.0.0.141      TLSv1.3   507 Application Data
 83 6.140367987 10.0.0.141      147.229.176.14  TLSv1.3   107 Application Data
 84 6.171745428 147.229.176.14  10.0.0.141      TLSv1.3   446 Application Data
 85 6.172012297 10.0.0.141      147.229.176.14  TLSv1.3   109 Application Data
 89 6.203795385 147.229.176.14  10.0.0.141      TLSv1.3   166 Application Data
 90 6.204052475 10.0.0.141      147.229.176.14  TLSv1.3   118 Application Data
 91 6.231944419 147.229.176.14  10.0.0.141      TLSv1.3  3173 Application Data
 92 6.232019779 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=564 Ack=7925 Win=63872 Len=0 TSval=1470778538 TSecr=2958394640
 93 6.264075742 147.229.176.14  10.0.0.141      TLSv1.3   135 Application Data
 94 6.264283314 10.0.0.141      147.229.176.14  TLSv1.3   118 Application Data
 95 6.294908900 147.229.176.14  10.0.0.141      TLSv1.3  2864 Application Data
 96 6.294979860 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=616 Ack=10792 Win=64128 Len=0 TSval=1470778601 TSecr=2958394703
 97 6.324706430 147.229.176.14  10.0.0.141      TLSv1.3   135 Application Data
 98 6.324906320 10.0.0.141      147.229.176.14  TLSv1.3   118 Application Data
 99 6.348658410 147.229.176.14  10.0.0.141      TLSv1.3  1470 Application Data
100 6.349243273 147.229.176.14  10.0.0.141      TLSv1.3  1973 Application Data
101 6.349274283 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=668 Ack=14172 Win=64128 Len=0 TSval=1470778655 TSecr=2958394757
102 6.381252790 147.229.176.14  10.0.0.141      TLSv1.3   135 Application Data
103 6.381394780 10.0.0.141      147.229.176.14  TLSv1.3   118 Application Data
104 6.410362441 147.229.176.14  10.0.0.141      TCP      1474 993 → 34238 [ACK] Seq=14241 Ack=720 Win=132096 Len=1408 TSval=2958394818 TSecr=1470778687 [TCP segment of a reassembled PDU]
105 6.412419656 147.229.176.14  10.0.0.141      TLSv1.3  2056 Application Data
106 6.412434951 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [ACK] Seq=720 Ack=17639 Win=64128 Len=0 TSval=1470778718 TSecr=2958394818
107 6.442759375 147.229.176.14  10.0.0.141      TLSv1.3   135 Application Data
108 6.443263709 10.0.0.141      147.229.176.14  TLSv1.3    90 Application Data
109 6.443398156 10.0.0.141      147.229.176.14  TCP        66 34238 → 993 [FIN, ACK] Seq=744 Ack=17708 Win=64128 Len=0 TSval=1470778749 TSecr=2958394851
110 6.475977659 147.229.176.14  10.0.0.141      TCP        66 [TCP Dup ACK 104#1] 993 → 34238 [ACK] Seq=17708 Ack=720 Win=132096 Len=0 TSval=2958394884 TSecr=1470778718
111 6.475978637 147.229.176.14  10.0.0.141      TCP        66 993 → 34238 [ACK] Seq=17708 Ack=744 Win=132096 Len=0 TSval=2958394884 TSecr=1470778749
112 6.475979545 147.229.176.14  10.0.0.141      TLSv1.3    90 Application Data
113 6.475980523 147.229.176.14  10.0.0.141      TCP        66 993 → 34238 [FIN, ACK] Seq=17732 Ack=744 Win=132096 Len=0 TSval=2958394884 TSecr=1470778749
114 6.476023057 10.0.0.141      147.229.176.14  TCP        54 34238 → 993 [RST] Seq=744 Win=0 Len=0
115 6.476035908 10.0.0.141      147.229.176.14  TCP        54 34238 → 993 [RST] Seq=744 Win=0 Len=0
```

2. **Test Connection to Correct and Incorrect Addresses:**
   2.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt*
   Output: ***Staženo 9 zpráv ze schránky INBOX.***

   2.2. *Input: ./imapcl fit.vutbr.cz -o maildir -a credentials.txt*
   Output: ***Invalid address/Domain name not supported***

3. **Test Port Parameters:**
   3.1. *Input: ./imapcl eva.fit.vutbr.cz -p 143 -o maildir -a credentials.txt*
   Output: ***Staženo 9 zpráv ze schránky INBOX.***

   3.2. *Input: ./imapcl eva.fit.vutbr.cz -p 14 -o maildir -a credentials.txt*
   Output: ***Connection Failed***

3.3. *Input: ./imapcl eva.fit.vutbr.cz -p asdf -o maildir -a credentials.txt*
Output: ***Error: port must contain only numbers.***

4. **Test Authentication with Valid and Invalid Credentials:**

4.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt*
Output: ***Staženo 9 zpráv ze schránky INBOX.***

4.2. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a wrong_credentials.txt*
Output: ***Server error: a001 NO [AUTHENTICATIONFAILED] Authentication failed.***

5. **Download Different Types of Messages:**

5.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -n*
Output: ***Žádné nové zprávy ve schránce INBOX.*** - pokud není k dispozici

Stažena 1 nová zpráva ze schránky INBOX. - pokud je k dispozici

5.2. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -h*
Output: ***Staženy hlavičky 6 zpráv ze schránky INBOX.***

5.3. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -h -n*
Output: ***Žádné nové zprávy ve schránce INBOX.*** - pokud není k dispozici

Stažena hlavička 1 nové zprávy ze schránky INBOX. - pokud je k dispozici

6. **Test Working with Certificates:**

6.1. *Input:* ./imapcl eva.fit.vutbr.cz -T -C Sertificate/ -o maildir -a credentials.txt
Output: ***Staženo 9 zpráv ze schránky INBOX***

6.2. *Input:* ./imapcl eva.fit.vutbr.cz -T -C Wrong_Sertificate/ -o maildir -a credentials.txt
Output: ***Certificate directory is empty or contains no valid certificates***

6.3. *Input:* ./imapcl eva.fit.vutbr.cz -T -c cacert.pem -o maildir -a credentials.txt
Output: ***Staženo 9 zpráv ze schránky INBOX.***

6.4. *Input:* ./imapcl eva.fit.vutbr.cz -T -c wrong.pem -o maildir -a credentials.txt
Output: ***Error loading certificate***

7. **Test Saving Messages:**

7.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt*

Output: ***Staženo 9 zpráv ze schránky INBOX.***

7.2. *Input: ./imapcl eva.fit.vutbr.cz -o wrong_maildir -a credentials.txt*

Output: ***Error: output directory does not exist: wrong_maildir***

8. **Test Mailbox Parameters:**

8.1. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -b Sent*

Output: ***Staženo 67 zpráv ze schránky Sent.***

8.2. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -b Trash*

Output: ***Staženo 1568 zpráv ze schránky Trash.***

8.3. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -b Error*

Output: ***Server error: a002 NO Mailbox doesn't exist: Error (0.001 + 0.000 secs).***

9. **Testing Invalid Command-Line Formats:**

9.1. *Input: ./imapcl -o maildir -a credentials.txt*

Output: ***Error: server IP or domain name is required***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.2. *Input: ./imapcl eva.fit.vutbr.cz -o maildir*

Output: ***Error: credentials file is required (-a auth_file)***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.3. *Input: ./imapcl eva.fit.vutbr.cz -a credentials.txt*

Output: ***Error: output directory is required (-o out_dir)***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.4. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -z*

Output: ***./imapcl: invalid option -- 'z'***

Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.5. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -b*

Output: ***./imapcl: option requires an argument -- 'b'***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.6. *Input: ./imapcl eva.fit.vutbr.cz extra_argument -o maildir -a credentials.txt*

Output: ***Error: unexpected argument(s): extra_argument***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

9.7. *Input: ./imapcl eva.fit.vutbr.cz -o maildir -a credentials.txt -C* Sertificate/

Output: ***Error: -C and -c options require -T to be specified.***
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir

8. References

1. RFC 3501 - IMAP4rev1
2. RFC 5322 - Internet Message Format
3. Dokumentace OpenSSL
4. Standardní knihovny C/C++