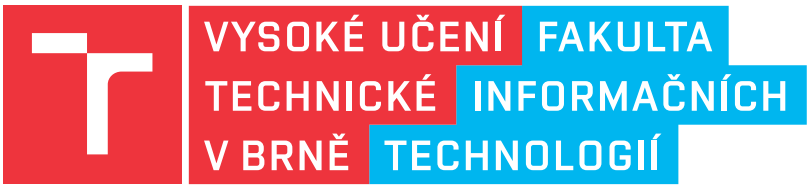

Documentation

ISA IMAP Client with TLS Support

Author:
Lilit Movsesian xmovse00

November 16, 2024

Contents

1 Overview

The *imapcl* program allows reading electronic mail using the IMAP4rev1 protocol (RFC 3501). The program downloads messages stored on the server and saves them in the RFC 5322 format in a specified directory (each message separately) and outputs the number of downloaded messages to standard output. Additional parameters can modify its functionality.

2 IMAP and IMAPS

The Internet Message Access Protocol (IMAP) is a protocol that allows clients to retrieve emails from a mail server. While IMAP is unencrypted, IMAPS(IMAP over SSL/TLS) provides a secure channel by using the Secure Sockets Layer (SSL) or Transport Layer Security (TLS). It typically operates on port 143 for unencrypted connections and on port 993 for encrypted connection. [3]

3 Relevant RFCs

3.1 RFC 3501

RFC 3501, or IMAP4rev1, specifies the protocol's commands and responses. All requests sent to the server must be in the format "UID REQUEST" where the "UID" is an alphanumeric string unique for each message.

The commands used in the implemented IMAP client `imapcl` are following:

- UID LOGIN username password
- UID SELECT mailbox
- UID SEARCH UNSEEN – Response contains a list of the UIDs of unread emails
- UID SEARCH ALL – Response contains a list of the UIDs of emails
- UID FETCH emailID BODY.PEEK[] – Sends the content of the email (header and body) in response
- UID FETCH emailID BODY.PEEK[HEADER] – Sends the content of the email header in response
- LOGOUT

The server responds in the format "UID RESPONSE OK/NO/BAD":

- OK – request was executed successfully
- NO – request was unsuccessful
- BAD – request was unsuccessful due to syntax error or another issue [1]

3.2 RFC 5322

RFC 5322 specifies the syntax for text messages that are sent using electronic mail. The messages contain:

- Message Header with Fields:
  - `Date`
  - `From`
  - `To`
  - `Subject`
  - `Message-ID` – a unique identifier for each message
  - `Reply-to`
  - `Content-type`
  - `Other Fields`
- Message Body – contains the actual content of the email, is separated from the header by an empty line, may include plain text, HTML, or any other MIME type [2]

4 Required Libraries

The program relies on the following libraries:

OpenSSL

- Version: 1.1.1 or higher
- Components:
  - libssl: Provides support for SSL and TLS protocols.
  - libcrypto: Provides cryptographic functions for secure data handling.
- Linking: Both libraries are linked in the Makefile with the following flags:

  -lssl -lcrypto

5 Compilation

The program comprises a *Makefile* and the source file *imapcl.c*. To compile the program, run:

```
make
```

To clean up the generated files, use:

```
make clean
```

6 Usage

The application can be started with the following command:

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

6.1 Parameters

- **server (mandatory)**: The name of the server (IP address or domain name) of the desired resource.
- **-p port (optional)**: Specifies the port number on the server.
- **-T**: Enables encryption (IMAPS). If this parameter is not provided, the unencrypted version of the protocol will be used.
- **-c certfile (optional)**: The certificate file used to verify the validity of the SSL/TLS certificate presented by the server.
- **-C certaddr (optional)**: Specifies the directory where certificates should be searched for. The default value is `/etc/ssl/certs`.
- **-n**: Processes only new messages.
- **-h**: Downloads only the headers of the messages.
- **-a auth_file (mandatory)**: Refers to the authentication file, which is formatted as follows:

  username = name
  password = secret

- **-b mailbox (optional)**: Specifies the name of the mailbox to work with on the server. The default value is **INBOX**.
- **-o out_dir (mandatory)**: Specifies the output directory where the program should save the downloaded messages.

7 Error Handling

In case of an error, the program terminates with a unified error code and prints error description to stderr.

8 Output

- The standard output will display the total number of downloaded messages or "No message to download" text.
- In the specified directory, the new files will be created, corresponding to the downloaded messages.

9 Message storage

The downloaded messages are stored in suitably named files in the directory specified by the -o parameter, with each message in a separate file.

9.1 Log Files

There are two log files maintained in the specified directory. One log, *log_h.txt*, is used for message headers, while the other log, *log.txt*, contains the full message bodies. After the message fetching, each message's ID is extracted from the response. Before downloading a message, the program checks whether its ID already exists in the appropriate log file. If the message ID is found in the log, the message is not downloaded again, ensuring that duplicates are avoided. If the message ID is not present, the message will be downloaded, and its ID will be appended to the log file to track the downloaded messages.

9.2 File Naming

The same email can be downloaded either as just the header or as the full message, and both versions will be saved. The filename for the message contains its index position in the log file, with the header files named with an *_h* suffix (e.g., *1_h.txt*) and the full message files without the suffix (e.g., *1.txt*).

10 Connection Types

The program supports two types of connections to an IMAP server: unencrypted (IMAP) and encrypted (IMAPS).

10.1 Unencrypted Connection (IMAP)

The connection is established using standard TCP sockets. The *connect_to_imap* function is responsible for creating the socket, resolving the server address, and attempting to connect. If the connection fails, an error message is printed to stderr, and the program exits with a failure code.

10.2 Encrypted Connection (IMAPS)

This connection is established using SSL/TLS to ensure secure communication. The *connect_to_imaps* function initializes the OpenSSL library, creates a new SSL context, and binds it to the socket. If any step fails, an error message is printed to stderr, and the program exits with a failure code.

Relevant Headers:

openssl/ssl.h for SSL/TLS functions.
openssl/err.h for error handling in OpenSSL.

11 Testing

The program was manually tested on both merlin.fit.vutbr.cz server and on Windows with WSL(Windows Subsystem for Linux) using personal email accounts on eva.fit.vutbr.cz and imap.pobox.sk. This testing involved:

- Connecting to both encrypted (IMAPS) and unencrypted (IMAP) servers.
- Fetching message headers and full message bodies.
- Verifying that the program correctly handled duplicate message downloads.
- Checking the log files (*log.txt* and *log_h.txt*) for accurate tracking of downloaded message IDs.
- Checking the downloaded messages.
- Verifying that the program correctly handled only new messages if the `-n` parameter is specified.
- Checking the functionality with both default and user-specified mailboxes.

11.1 Test Cases

```
> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir -b imap -T -h
4 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir -b imap -T -h
0 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir -b imap -T
4 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir -b imap -T
0 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir2 -b imap
4 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir2 -b imap
0 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir2 -b imap -h
4 messages were downloaded from the 'imap' mailbox.

> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -o ./dir2 -b imap -h
0 messages were downloaded from the 'imap' mailbox.
```

```
> ./imapcl eva.fit.vutbr.cz -a ./auth2.txt -b imap -h
Missing required argument -a auth_file or -o out_dir.

> ./imapcl eva.fit.vutbr.cz -o ./dir2 -b imap -h
Missing required argument -a auth_file or -o out_dir.

> ./imapcl -a ./auth2.txt -o ./dir2 -b imap -h
Missing server or IP address argument.

> ./imapcl imap.pobox.sk -a ./auth.txt -o ./dir -n
1 messages were downloaded from the 'INBOX' mailbox.

> ./imapcl imap.pobox.sk -a ./auth.txt -o ./dir
2 messages were downloaded from the 'INBOX' mailbox.

> ./imapcl imap.pobox.sk -a ./auth.txt -o ./dir -n
0 messages were downloaded from the 'INBOX' mailbox.
```

References

[1] M. Crispin. *Internet Message Access Protocol - Version 4rev1*. 2003. Available at: https://tools.ietf.org/html/rfc3501.

[2] P. Resnick. *Internet Message Format*. 2008. Available at: https://tools.ietf.org/html/rfc5322.

[3] *Service Name and Transport Protocol Port Number Registry*. Last Updated: 2024-10-15. Available at: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.