

Network Applications and network Administration (ISA)

# Project documentation – POP3 client with TLS support

Samuel Líška (xliska20)

Table of Contents

[Project documentation – POP3 client with TLS support 1](#_Toc87774523)

[1. Introduction 3](#_Toc87774524)

[2. Usage 3](#_Toc87774525)

[3. Files 3](#_Toc87774526)

[Globals.h 3](#_Toc87774527)

[Handler.h 3](#_Toc87774528)

[Popcl.c 4](#_Toc87774529)

[Makefile 4](#_Toc87774530)

[4. Implementation 4](#_Toc87774531)

[Establishing connection and login 4](#_Toc87774532)

[Retrieving mail messages 4](#_Toc87774533)

[Deleting mail messages 4](#_Toc87774534)

[Closing connection 4](#_Toc87774535)

[Parsing arguments 5](#_Toc87774536)

[5. Sources 5](#_Toc87774537)

# Introduction

This solution is able to read electronic mail using POP3 protocol (RFC 1930) with POP3s extension (RFC 2595). Program uses basic USER/PASS authentication. Basic usage downlaods contents of the mail server to local device. This behaviour can be altered with program arguments which are described lower. This software is **POSSIXLY CORRECT** and requires openssl library installed (developed on openssl **OpenSSL 1.1.1j** and tested on **OpenSSL 1.1.1l – freebsd**)

# Usage

Run makefile with „make“ to compile program. Then run with:

**popcl <server> [-p <port>] [-T|-S [-c <certfile>] [-C <certadds>]] [-d] [-n] -a <auth\_fie> -o <out\_dir>**

Where:

**<server>** - is IP address or domain of source (required)

**-p** is port

**-T** enforces encrypted connection with pop3s or -S non-encrypted connection STLS

**-S** upgrades non-crypted connection to STLS crypted variant

**-c** defines file with certificates

**-C** defines folder with certificates for SSL/TLS

**-d** set program for deletion of messages

**-n** read only new messages

**-a** authetification file <auth\_file> credentials (required)

**-o** defines output file <out\_dir> (required) \n"

<auth\_file> login credentials must be exactly in this format:

*username = name*

*password = password*

After correctly running program, number of downloaded messages is printed on stdio and each mail is saved into *<out\_dir>*  into separated file in Internet **Message Format according** to **RFC 5332.**

# Files

### Globals.h

Header source file, which holds constants and global variables, such as buffes for varius use, pop3 constants and openssl related variables.

### Handler.h

Header source file that is responsible for C language reated problems. Various things are solved here such as hanlding strings, working with files and parsing arguments. Some memory work is done as well.

### Popcl.c

Main source file. Actually represents POP3 client. All communications with server if done within this file.

### Makefile

With command *make* compiles program to executable.

# Implementation

Software is developed with C language and is deployed in four source files, with modular approach.

### Establishing connection and login

Default non-crypted connection is establisthed with BIO socket using **BIO\_new\_connect()**  function and is check with **BIO\_do\_connect()**. When **-T** parameter is used, secured connection is established with creating new contex **SSL\_CTX\_NEW().** Afterwards if -c / -C parameters are used, default verify paths are set. Next ssl connect is created and proper mode set before verifying certificates. With **-S** argument default non-crypted connection is established and **STLS\r\n** is send to server. If TLS negotiation is successful context is set and certificates are set. After that **BIO\_push** and BIO\_get\_sslare used to upgrade communication to TLS.

After successfully establishing connection login credentials are send to server.

### Retrieving mail messages

After sending **STAT** command to pop server number of messages if recieved. That is used cycle through messages with **RETR „x“** command. Responses from RETR are multilined, which can result (and most likely will) in buffer overflow, since **BUFF\_SIZE** is set to 1024. That is why **BIO** is being constantly read until **CRLF.CRLF** pattern is found, which according to RFC indicates that response from the POP server is ended and **.CRLF** is not part of message. If line begins with decimal code 046, which is actually „ **.**“ followed by CRLF is **byte-stuffed.** Client checks if line begins with termination octet. If so and if octets other than CRLF follow, first octet of the line (termination octet – „ **.** “) is stripped away.

When reading responses from each mail, **Message-ID** is found from within the mail, which is used to name output file that mail is being saved to. **Message-ID** of each mail is then saved into **downloaded.txt** file.

**-n argument :** Each Message-ID is retrieved before saving mail, and if it´s not present in **downloaded.txt** file, it is considered to be new mail. This approach downloads only new messages, but is possibly not fastests, since whole message must be read anyway to find out Message-ID of mail.

## Deleting mail messages

When -d is used **DELE “x”** command is send to the pop3 server. This argument deletes mails from server not from local computer! When combined with **-n,** only new messages are deleted.

## Closing connection

Afterward, pop3 connection is closed by **“QUIT”**, while all memory is freed. This includes all alocations, bio sockets, ctx etc.

## Parsing arguments

Argument parsing is done by hand. No external function (such as getopt) is used, since these would most likely cause problems later. **argParse()** function worsk similiary and is defined in **handler.h.** First argument without “ **- „** is considered to be **source** (IP addres or domain) while others are ignored.

# Sources

RFCs - <https://datatracker.ietf.org/doc/html/rfc1939> , <https://datatracker.ietf.org/doc/html/rfc2595>

OpenSSL API - <http://www.ibm.com/developerworks/library/l-openssl/> , <https://www.openssl.org/>

ISA Lectures, IPK Lectures, UNIX manual page(man)