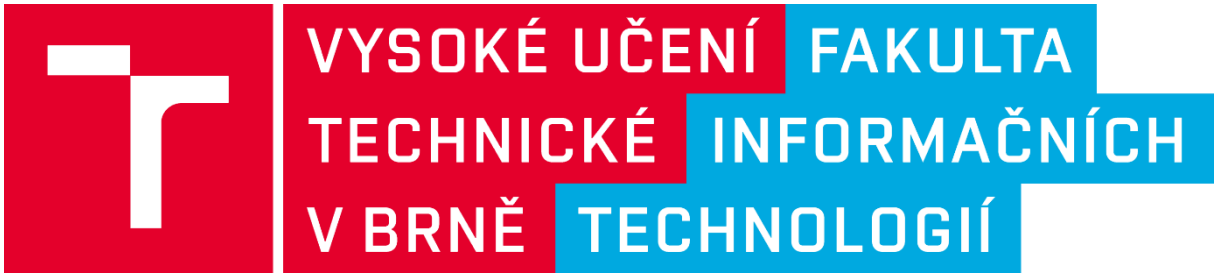
VYSOKÉ UČENÍ TECHNICKÉ V BRNE
FAKULTA INFORMAČNÝCH TECHNOLÓGIÍ



ISA - Síťové aplikace a správa sítí

Programování síťové služby – Klient POP3 s podporou TLS

Matej Horník (xhorni20)

19.10.2021

Obsah

1. Úvod

Cieľom projektu bolo implementovať klienta POP3 s podporou TLS, ktorý na základe požiadavku od klienta stiahne emaily zo servera do zadaného priečinka alebo vymaže emaily na zadanom serveri pomocou POP3 protokolu definovaného v RFC1939[1]. Klient taktiež dokáže komunikovať zo serverom cez šifrované spojenie pomocou TLS[3] v prípade ak uživateľ použije daný argument . K projektu taktiež bolo potrebné vypracovať projektovú dokumentáciu. Tutoriál od IBM – programovanie s OPENSSL API bol skvelý začiatok na pochopenie danej problematiky[4].

2. Implementácia

2.1. Základné informácie

Klient POP3 s podporou TLS bol implementovaný pomocou knižnice **openssl.** Program bol implementovaný v jazku C++ a prekladaný pomocou g++ v štandarde C++17. Na preloženie program bol použitý **Makefile.** Po prevedení príkazu *make* sa program preloží na výslednú binárku s názvom popcl. Celý program je implementovaný v súbore popcl.cpp.

2.2. Spustenie programu

Program podporuje následné parameter(názov servera musí byť prvý argument):

-a <auth_file>: **povinný argument**, súbor s prihlasovacími údajmi na zadaný server

-o <out_dir>: **povinný argument**, priečinok do ktorého sa stiahnu emaily zo servera

-p <port>: špecifikuje číslo portu na serveri

-d: zo serveru sa vymažú emaily

-n: zo serveru sa stiahnu len nové emaily, ktoré sa nenachádzajú v zadanom priečinku

-T{-c -C}: zapne sa šifrovanie celej komunikácie(pop3s)

-S{-c -C}: so serverom naviaže šifrované spojenie po pripojení a príkaze STLS

-c <certfile>: definuje zadaný súbor s certifikáty, ktorý sa použije pre overenie platnosti certifikátu SSL/TLS predloženého serverom. Použitie len s -S/-T.

-C <certaddr>: definuje priečinok v ktorom sa majú vyhladávať certifikáty, ktoré sa použijú pre overenie platnosti certifikátu SSL/TLS predloženého serverom. Použitie len s -S/-T.

Pri použití argumentu -d zároveň s -n sa zo servera vymažú emaily a žiadne emaily sa nestiahnu.

Použitie programu:

./popcl <server> [-p <port>] [-T|-S [-c <certfile>] [-C <certaddr>]] [-d] [-n] -a <auth_file> -o <out_dir>

Formát autetizačného súboru:

username = meno

password = heslo

2.3. Popis samotného programu

Pri spustení programu sa najprv skontrolujú a spracujú vstupné argumenty pomocou funkcie parse_arguments(). Na spracovanie argumentov nebola použitá žiadna knižnica. Potrebné konfiguračné údaje sa uložia do konfiguračnej štruktúry. Pri spracovávaní argumentov sa taktiež skontroluje existencia súborov či priečinkov. Taktiež sa kontroluje formát autentifikačného súboru.

Po spracovaní argumentov sa nastaví **openssl** knižnica, ktorá je použitá ako aj pre šifrovanú aj nešifrovanú komunikáciu. V prípade ak bol program spustení s argumentom -T alebo -S tak sa načítajú certifikáty a vytvoria sa potrebné dátové štruktúry pre šifrovanú komunikáciu. Následne sa volá funkcia **pop3sesion()** ktorá riadi komunikáciu medzi klientom a serverom. Na základe vstupných argumentov sa na server pripája zabezpečene alebo nezabezpečene. Po

úspešnom pripojení na server sa program pokúsi o autorizáciu užívateľa pomocou príkazov *USER* a *PASS*. Posielane príkazov na server je implementované vo funkci **send_command()**, ktorá v prípade chyby vyhodí výnimku *MyException* s popisom chyby. Čítanie správ od servera je implementované vo funkcii **get_response()**, ktorá v prípade chyby vyhodí výnimku *MyException* s popisom chyby alebo vráti odpoveď v reťazci. Pre prácu so socketmi je použitá knižnica *openssl*, konkrétne modul *bio.h.*

Následne na základe požiadavku od uživateľa sa buď stiahnu všetky emaily, len nové alebo sa vymažú emaily zo servera. Pri sťahovaní emailov sa použije príkaz *STAT*, ktorý zistí počet emailov na serveri. Na získanie emailu zo servera sa použije príkaz *RETR* s číslom emailu ktorý chceme stiahnuť. Stiahnutie emailu spracováva funkcia **download_email()** ktorá v prípade úspešného stiahnutia emailu vráti daný email v reťazci. Po dokončení sťahovania všetkých emailov sa štandardný výstup vypíše koľko emailov bolo stiahnutých. Sťahovanie nových emailov je spravené na základe zadaného výstupného priečinka s emailmi. To znamená že ak sa daný email ešte nenachádza v priečinku tak sa považuje za nový email. Nevýhoda tohto prístupe je že uživateľ si môže sťahovať emaily do rôznych priečinkov. Po dokončení sťahovania sa vypíše informácia na výstup o počte nových stiahnutých emailov. Ako názov súbora pre stiahnutý email sa používa unikátny kód, ktorý sa volá *Message-Id*. Na vymazanie emailov zo servera sa používa príkaz *DELE* a príslušne číslo emailu na zmazanie. Po úspešnom mazaní sa vypíše informácia koľko emailov bolo vymazaných. Pre ukončenie komunikácie so serverom sa zašle serveru príkaz *QUIT*, ktorý oznamuje koniec komunikácie. Po skončení sa uvoľnia všetky alokované dátové štruktúry openssl.

3. POP3

Pojem POP3 označuje Post Office Protocol – Version 3, čo je internetový protokol aplikačnej vrstvy používaný na správu emailov. Používa sa komunikácia klient – server. Klient po pripojení posiela príkazy serveru, server ich spracuje a poskytne odpoveď. Protokol popisuje presný formát správ a odpovedí. Odpoveď musí začínať s +OK v prípade úspešnej odpovedi a -ERR pri neúspešnej odpovedi. Každá správa končí \r\n (CRLF). Pri zasielaní emailov správa končí ".\r\n". Spávy prichádzajú vo formáte IMF – internet message format, ktorý je definovaný v RFC5322[2].

4. Testovanie

Pri lokálnom testovaní bola použitá aplikácia hMailServer, na ktorej sa bolo možné otestovať vlastné certifikáty. Taktiež pri testovaní boli použité emailové schránky *centrum.sk* a *seznam.cz*.

5. Zdroje

[1] J. Myers, Carnegie Mellon, M. Rose.: Post Office Protocol - Version 3. 1996, [Online] URL https://datatracker.ietf.org/doc/html/rfc1939

[2] P. Resnick, Ed.: Internet Message Format. 2008, [Online] URL https://datatracker.ietf.org/doc/html/rfc5322

[3] C. Newman.: Using TLS with IMAP, POP3 and ACAP. 1999, [Online] URL https://datatracker.ietf.org/doc/html/rfc2595

[4] Kenneth Ballard.: Secure programming with the OpenSSL API. 2004, [Online] URL https://developer.ibm.com/tutorials/l-openssl/