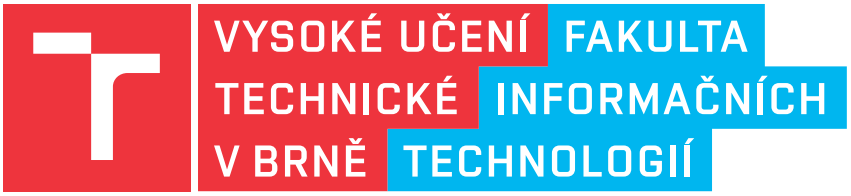

Klient POP3 s podporou TLS
ISA
Síťové aplikace a správa sítí

30. října 2021 Jakub Komárek (xkomar33)

Obsah

1 Úvod

Cílem práce bylo implementovat klienta pro stahování zpráv z POP3 serveru s podporou šífrování. Tento dokument popisuje implementaci tohoto programu a jeho použití.

2 Protokol POP3

Jedná se o internetový protokol sloužící pro stahování emailů z poštovního serveru [3]. Standartně probíhá komunikace na portu 110 TCP, v šifrované verzi protokolu poté na 995 TCP [5]. Pro komunikaci využívá několik klíčových příkazů, které klient zasílá na server, kde jsou zpracovány a následně server vrací příslušnou odpověď. Kompletní výčet příkazů je k nalezení v dokumentaci [3].

3 Návrh a implementace

Klient je implementován jako třída `popClient`. Hlavní chod programu je řízen z funkce main, kde se případně zpracovávají výjimky. Na začátku činnosti programu se objekt klienta zkonstruuje za pomocí vstupních argumentů. Nesprávná kombinace argumentů vyvolává výjimku `invalid_argument`. Po správném zparsování argumentů se inicializuje knihovna `openSSL` a získají se přihlašovací data ze zadaného souboru. Hlavní běh programu se skládá ze tří částí - navázání spojení, stahování a vypsání výsledků. Poté dojde k dealokaci zdrojů a korektní ukončení. Během činnosti programu reaguje objekt na neočekávanou událost vyvoláním výjimky `runtime_error` s jejím konkrétním popisem.

3.1 Navázání spojení

Podle zadaných parametrů se zvolí přístup navázání spojení. Tuto činnost má na starost metoda `estConnection`. Ta poté volá jednotlivé procedury pro navázání spojení. Po skončení této metody může program začít posílat jednotlivé požadavky na server. Při implementaci této části jsem masivně využíval knihovnu `openssl/bio` a vycházel jsem z ukázkových příkladů dokumentaci této knihovny [2] a manuálu [4].

3.2 Stahování

Před stahováním se program dotáže serveru na počet zpráv dostupný ve schránce. Poté cyklicky stáhne a zpracuje každou jednu zprávu.

Samotné stahování jednoho emailu probíhá cyklicky přes interní buffer dokud se nepřijme poslední segment zakončený ukončující sekvencí. Poté je email zpracován do formátu RFC 5322 [1] a následně uložen do předem specifikované složky. Jméno souboru je zvoleno podle položky `Messege-ID` v emailu. `Messege-ID` by mělo být pro každou zprávu jedinečné, proto by nemělo docházet k přepsání. Pokud z nějakého důvodu se `Messege-id` v emailu nenachází, je emailu přiděleno náhodné nekonfliktní číslo (takové číslo, které se ve výstupním adresáři nenachází), pod kterém se zpráva uloží. V tomto případě pak program nemůže zjistit novost zprávy. Každý uložený email má koncovku `.eml`. Pokud email nelze uložit (nedostatečná oprávnění/jiná chyba), je tato skutečnost oznámena na standardní chybový výstup a program se pokusí pokračovat.

3.2.1 Ukládání pouze nových zpráv

Novost zprávy program zjišťuje za pomocí `Messege-ID`, které by se mělo v emailu nacházet. Pokud je tedy ve výstupním adresáři email s tímto `Messege-ID`, zpráva není uložena znovu. Tento přístup tedy běh programu o moc nezrychluje a proto je doporučeno občasně schránku vyprázdnit. Rovněž je problém, pokud se `Messege-ID` v emailu nenachází (v tom případě je zpráva stažena znovu).

3.2.2 Mazání zpráv

Mazání zprávy se provádí pouze v případě, že byla zpráva úspěšně uložena a je zapnutý příslušný přepínač.

4 Návod k použití

Program se spouští z příkazové řádky. Pro korektní spouštění je potřeba zadat první argument - adresu serveru (adresa IP/doménové jméno). Poté je nutné ještě v libovolném pořadí zadat výstupní složku (pokud složka neexistuje, pokusí se program ji vytvořit), do které se budou emaily ukládat. Další povinný parametr je autentizační soubor s přihlašovacími údaji (formát viz. 4.2). Pokud není zadán port, zvolí se implicitní - pro nešifrovanou komunikaci 110, pro šifrovanou 995 [6]. Pokud je zvolen parameter -S, výchozí port je nastaven na 110. Přepínače -T a -S nelze kombinovat. Při použití šifrovaní a nespecifikování zdroje certifikátů se vybere implicitní místo pro certifikáty. Po spuštění a dokončení činnosti vypíše počet stažených souborů. V případě chyby, vypíše příslušnou chybovou hlášku.

4.1 Syntaxe spuštění

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

příklad spuštění: $ *popcl seznam.cz -p 995 -T -n -o maildir -a login.txt*

- -h vypíše nápovědu.
- <server> IP adresa/doménové jméno serveru.
- -p <port> specifikuje číslo portu serveru.
- -T zapíná šifrování celé komunikace (pop3s).
- -S naváže nešifrované spojení se serverem a pomocí příkazu STLS přejde na šifrovanou variantu protokolu.
- - c <certfile> definuje soubor s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- -C <certaddr> určuje adresář, ve kterém se vyhledávají certifikáty, které se použijí pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- -d po stažení se zprávy, odstraní danou zprávu ze serveru.
- -n stažení pouze nových zpráv.
- -a <auth_file> umístění souboru s přihlašovacíma údaji].
- -o <out_dir> specifikuje výstupní adresář, do kterého program stažené zprávy ukládá.

4.2 Autentizační soubor

Autentizační soubor obsahuje 2 položky - `username` a `password`. Tyto položky je nutno oddělit novým řádkem. Těmito údaji se bude program přihlašovat na emailový server. K jednotlivým položkám se údaje přiřazují pomocí znaku =. Při nedodržení této konvence skončí program s chybou.

username = *váš*Login
password = *vaše*Heslo

5 Testování

Testování bylo prováděno manuálně. V průběhu vývoje byl využit lokální server hMail. Po dokončení jsem využil pro testy servery Seznamu. Z testování vyplynulo, že některé emaily nemají ve svém těle Messege-ID a implementace tomuto faktu byla přizpůsobena 3.2. Aplikace neměla problém přenést velké objemy dat a nebyl problém s poškozenými daty. Překlad a funkčnost byla v poslední verzi úspěšně otestována na obou referenčních strojích (merlin, eva).

6 Závěr

Podle mého vědomí by program měl bez omezení splňovat požadavky v zadání práce.

Odkazy

[1] *Internet Message Format*. URL: `https://datatracker.ietf.org/doc/html/rfc5322`. (navštíveno: 31.10.2021).

[2] *ManualOpenSSl*. URL: `https://www.openssl.org/docs/man1.1.0/man3/BIO_new_ssl_connect.html`. (navštíveno: 31.10.2021).

[3] *Post Office Protocol - Version 3*. URL: `https://datatracker.ietf.org/doc/html/rfc1939`. (navštíveno: 31.10.2021).

[4] *Secure programming with the OpenSSL API*. URL: `https://developer.ibm.com/tutorials/l-openssl/`. (navštíveno: 31.10.2021).

[5] *Service Name and Transport Protocol Port Number Registry*. URL: `http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`. (navštíveno: 31.10.2021).

[6] *Using TLS with IMAP, POP3 and ACAP*. URL: `https://datatracker.ietf.org/doc/html/rfc2595`. (navštíveno: 31.10.2021).