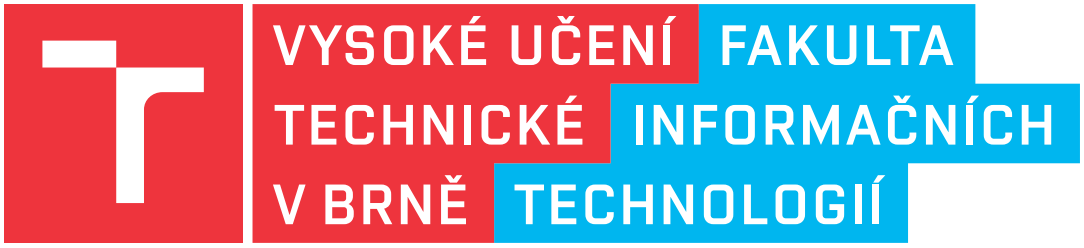

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Síťové aplikace a správa sítí

Dokumentace k projektu

Pavlo Kornieiev (xkorni03)

28. října 2024

Obsah

1 Úvod

Program imapcl je navržen pro čtení elektronické pošty pomocí protokolu IMAP4rev1 (RFC 3501). Tento program po spuštění stáhne všechny zprávy uložené na zadaném IMAP serveru a uloží je jako samostatné soubory do zadaného adresáře. Na standardní výstup následně vypíše počet úspěšně stažených zpráv. Program umožňuje upravit svou funkcionalitu prostřednictvím dodatečných parametrů, které specifikují chování aplikace.

2 Návrh a implementace

Imapcl je nástroj napsaný v jazyce C pro stahování emailových zpráv ze vzdáleného IMAP serveru pomocí protokolu IMAP4rev1. Cílem je umožnit uživateli pohodlně stáhnout e-maily do místního adresáře, a to buď jako celé zprávy, nebo pouze jejich hlavičky, s možností stahovat pouze nové nebo všechny zprávy. Program podporuje připojení přes SSL/TLS, aby bylo zajištěno bezpečné přenosy dat.

2.1 Struktura programu

Program je rozdělen na několik hlavních částí:

1. Pro čtení konfiguračních parametrů a autentizačních dat program používá struktury `ImapConfig` a `AuthData` pro ukládání konfiguračních informací, jako je adresa serveru, port, cesta k certifikátům, požadavek na připojení přes SSL/TLS, složka, ze které se mají e-maily stáhnout, a další nastavení. Při spuštění programu se načtou potřebné parametry příkazového řádku, které se poté uloží do struktury `ImapConfig`. Soubor s autentizačními údaji (uživatelské jméno a heslo) je načten pomocí funkce `read_auth_file`.

```
typedef struct {
    char *server;
    int port;
    int use_tls;
    char *certfile;
    char *certaddr;
    int new_only;
    int headers_only;
    char *auth_file;
    char *mailbox;
    char *out_dir;
} ImapConfig;


typedef struct {
    char username[256];
    char password[256];
} AuthData;
```

2. Funkce `connect_to_server` naváže připojení k serveru. Program podporuje standardní i zabezpečené připojení (SSL/TLS), přičemž pro SSL/TLS připojení využívá knihovnu OpenSSL. Pokud je vyžadováno SSL, program načte certifikát a provede ověření certifikátu serveru.

3. Po navázání připojení se program přihlásí k serveru pomocí IMAP příkazu LOGIN a autentizačních údajů. Následně použije příkaz SELECT pro výběr složky (ve výchozím nastavení INBOX). Program využívá funkce `send_imap_command` pro odesílání příkazů serveru a `read_imap_response` pro čtení odpovědí. Každý příkaz je odeslán s unikátním tagem pro identifikaci odpovědi serveru.

4. Funkce `search_and_fetch_emails` umožňuje vyhledat požadované zprávy (nové nebo všechny) a stáhnout jejich obsah. Pro vyhledání zpráv je použit příkaz SEARCH, který vrací seznam ID emailů. Následně je každé ID zpracováno a zpráva je stažena příkazem FETCH. Uživatel si může vybrat, zda chce stáhnout jen hlavičky nebo celé zprávy.

5. Program kontroluje, zda cílový výstupní adresář existuje, a pokud ne, vytvoří jej pomocí funkce create_output_directory. Každá zpráva je poté uložena jako samostatný soubor v tomto adresáři. Ukládání je realizováno v `fetch_and_save_email`, která identifikuje novou zprávu podle Message-ID a ověřuje, zda už byla dříve stažena.

6. Na konci programu je spojení ukončeno, SSL kontext uvolněn, a v případě úspěšného stahování je vypsán počet stažených zpráv.

2.2 Stav aplikace

Soubor `uids_map` slouží jako databáze pro sledování stažených e-mailů. Účelem tohoto souboru je uchovávat seznam Message-ID (s příznakem -H pro stahování pouze hlaviček), aby se zabránilo opakovanému stahování už stažených e-mailů. `uids_map` je vytvářen v každém výstupním adresáři (out_dir), což znamená, že každý projekt nebo relace stahování e-mailů má svůj vlastní záznam o již stažených e-mailech. To umožňuje uživatelům snadno spravovat stahování v různých kontextech a zamezuje konfliktům mezi různými relacemi.

Každý e-mail má unikátní hlavičku Message-ID, která slouží k jeho odlišení od ostatních zpráv. Tento Message-ID se načítá z hlavičky e-mailu a ukládá se do `uids_map` jako označení, že byl daný e-mail stažen.

Program kontroluje, zda je Message-ID (nebo jeho varianta pro stahování hlavičky označená příponou -H) již uložen v souboru `uids_map`. Pokud je identifikátor v souboru nalezen, e-mail se přeskočí, čímž se zabrání duplicitnímu stahování. Pokud nalezen není, program identifikuje e-mail jako nový a po jeho stažení uloží Message-ID do `uids_map`.

Program ukládá e-maily a jejich odpovídající hlavičky (s označením -H) do různých souborů, což znamená, že i když mají stejný Message-ID, nedochází k přepisování. Tímto způsobem je možné mít jak kompletní e-maily, tak jejich hlavičky staženy současně, což poskytuje uživateli flexibilitu při práci s oběma formáty zpráv.

Díky uchovávání záznamu v `uids_map` lze snadno stahovat pouze nové e-maily a vyhnout se opětovnému stahování již stažených zpráv.

2.3 SSL/TLS

V projektu je použito zabezpečení SSL a TLS pro šifrování komunikace mezi klientem a serverem. Tento mechanismus zabezpečuje, že data, která jsou přenášena mezi e-mailovým klientem a IMAP serverem, jsou chráněna.

1. Program začíná inicializací knihovny OpenSSL pomocí funkcí jako SSL_library_init(), OpenSSL_add_all_algorithms() a SSL_load_error_strings(). To umožňuje práci s SSL/TLS a zpracování kryptografických algoritmů.

```
if (use_ssl) {
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ...
```

2. Pomocí SSL_CTX_new(TLS_client_method()) se vytváří nový SSL kontext, který se používá pro nastavení a řízení SSL/TLS relace. Tento kontext se následně používá k vytvoření SSL objektu.

```
*ssl_ctx = SSL_CTX_new(TLS_client_method());
if (!*ssl_ctx) {
    fprintf(stderr, "SSL_CTX_new failed.\n");
    close(sock);
    return ERROR_SOCKET_CREATION;
}
```

3. Program zajišťuje načtení certifikátů z daného souboru nebo adresáře pomocí SSL_CTX_load_verify_locations(). Tím se ověřuje důvěryhodnost serverového certifikátu a tím se zajišťuje, že komunikace probíhá s legitimním serverem.

```
if (certfile || certaddr) {
    if (!SSL_CTX_load_verify_locations(*ssl_ctx,
    (certfile ? certfile : NULL),
    (certaddr ? certaddr : NULL))) {
        fprintf(stderr, "Could not load certificates from file '%s'
        or directory '%s'\n", certfile ? certfile :
        "none", certaddr ? certaddr : "none");
        SSL_CTX_free(*ssl_ctx);
        close(sock);
        return ERROR_SOCKET_CREATION;
    }
}
```

4. Po úspěšném připojení k serveru se volá SSL_connect(), která zahajuje SSL/TLS handshake, během kterého se navazuje zabezpečené spojení. V případě, že handshake selže, program zobrazí chybu a ukončí proces.

5. Program také provádí ověření certifikátu pomocí SSL_get_verify_result(), aby se zajistilo, že certifikát serveru je platný a důvěryhodný. Pokud dojde k chybě při ověřování, je spojení ukončeno.

3 Testování

Aplikace byla manuálně testována na serveru pobox.sk, kde jsem vytvořil účet pro testování. Během testování jsem ověřil fungování SSL/TLS s výchozími certifikáty z adresáře /etc/ssl/certs.

Testoval jsem také různé funkce aplikace, včetně: Stahování hlaviček e-mailů pomocí přepínače -h; Stahování nepřečtených e-mailů pomocí přepínače -n; Správu stavu uids_map, která zajišťuje, že již stažené e-maily nejsou staženy znovu, a to při různých konfiguracích a parametrech.

Navíc jsem testoval aplikaci s různými výstupními adresáři, abych ověřil, že soubory jsou správně ukládány do specifikovaných umístění a že nedochází k přepsání existujících souborů. Tyto testy prokázaly, že aplikace správně zvládá více instancí a různá nastavení bez ztráty dat.

Další testy proběhly na serveru Merlin, kde jsem ověřil, že příkaz make funguje a všechny soubory jsou správně zkompilovány a spuštěny. Tento krok zajistil, že aplikace je připravena pro nasazení a bez problémů běží v cílovém prostředí.

4 Použití

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

4.1 Pořadí parametrů

Pořadí parametrů je libovolné.

4.2 Popis parametrů

- **server**: (Povinný) Název serveru (IP adresa nebo doménové jméno) požadovaného zdroje.
- **-p port**: (Volitelný) Číslo portu na serveru. Zvolte vhodnou výchozí hodnotu v závislosti na specifikaci parametru -T a číslech portů registrovaných organizací IANA.
- **-T**: Zapíná šifrování (imaps). Pokud není tento parametr uveden, použije se nešifrovaná varianta protokolu.
- **-c certfile**: (Volitelný) Soubor s certifikáty, který se použije pro ověření platnosti certifikátu SSL/TLS předloženého serverem.
- **-C certaddr**: (Volitelný) Určuje adresář, ve kterém se mají vyhledávat certifikáty, které se použijí pro ověření platnosti certifikátu SSL/TLS předloženého serverem. Výchozí hodnota je `/etc/ssl/certs`.
- **-n**: Při použití tohoto parametru se bude pracovat (číst) pouze s novými zprávami.
- **-h**: Při použití tohoto parametru se budou stahovat pouze hlavičky zpráv.
- **-a auth_file**: (Povinný) Odkazuje na soubor s autentizací (příkaz LOGIN). Obsah konfiguračního souboru `auth_file` je zobrazený níže.
- **-b MAILBOX**: (Volitelný) Specifikuje název schránky, se kterou se bude na serveru pracovat. Výchozí hodnota je `INBOX`.
- **-o out_dir**: (Povinný) Specifikuje výstupní adresář, do kterého má program stažené zprávy uložit.

4.3 Příklad použití

Aplikace se kompiluje pomocí nástroje `make`. Kompilace probíhá na základě souboru `Makefile`, který obsahuje instrukce pro přeložení zdrojového kódu. Hlavní soubor, který je třeba zkompilovat, je `imapcl.c`. Po úspěšné kompilaci bude vytvořen spustitelný soubor `imapcl`.

./imapcl imap.pobox.sk -p 993 -T -a auth_file -o maildir

Tento příklad se připojí k serveru `imap.pobox.sk` přes port 993 s TLS, použije autentizační údaje ze souboru `auth_file` a uloží stažené e-maily do specifikovaného výstupního adresáře maildir.

5 Použitá literatura

- INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. https://datatracker.ietf.org/doc/html/rfc3501
- Internet Message Format. https://datatracker.ietf.org/doc/html/rfc5322
- Secure programming with the OpenSSL API. https://developer.ibm.com/tutorials/l-openssl/
- Service Name and Transport Protocol Port Number Registry. https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml