

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Siet'ové aplikácie a správa sietí Klient POP3 s podporou TLS

Obsah

1	Úvod	2
2	POP3 protokol	2
2.1	Nešifrovaná komunikácia	2
2.2	Šifrovaná komunikácia	2
3	Návrh a implementácia	2
3.1	Parsovanie argumentov	2
3.2	Spracovanie autentizačného súboru	3
3.3	Nadviazanie spojenia so serverom	3
3.3.1	Nezabezpečené spojenie	3
3.3.2	Zabezpečené spojenie	3
3.3.3	Nezabezpečené spojenie a príkaz STLS	3
3.4	Autentizácia užívateľa	3
3.5	Počet správ na serveri	4
3.6	Ukladanie správ	4
3.6.1	Overenie úložiska mailov	4
3.6.2	Prijímanie správ	4
3.7	Mazanie správ	4
3.8	Ukončenie spojenia	4
3.9	Výsledný výpis a uvoľnenie zdrojov	4
3.10	Pomocné funkcie	4
3.10.1	Posielanie príkazov	5
3.10.2	Prijímanie správ	5
4	Základné informácie o programe	5
4.1	Povinné parametre	5
4.2	Kombinácia parametrov	5
4.2.1	Parameter -S a -T	5
4.2.2	Parameter -c a -C	6
4.2.3	Parameter -d a -n	6
5	Testovanie skriptu	6
5.1	Testovanie nezabezpečeného pripojenia	6
5.2	Testovanie zabezpečeného pripojenia	7
5.3	Testovanie STLS	9
5.4	Testovanie správnosti uloženia mailu s bodkou na začiatku riadku	10
5.5	Testovanie situácií, ktoré majú ukončiť program	10
5.5.1	Nesprávne prihlasovacie údaje	10
5.5.2	Zadanie neexistujúcej zložky pre ukladanie mailov	11
5.5.3	Kombinácia -S a -T parametra	11
5.5.4	Neplatnosť certifikátov	11
5.5.5	Pripojenie na zlý port	12
6	Záver	12

1 Úvod

Cieľom tohto projektu bolo vytvoriť mailového klienta, ktorý umožňuje čítanie elektronickej pošty cez POP3 protokol pričom klient zvláda šifrovanú aj nešifrovanú komunikáciu. Po spustení programu sa z uvedeného servera stiahnu maily a uložia sa do zadaného adresára.

2 POP3 protokol

Jedná sa o najčastejšie používaný protokol pre prijímanie elektronickej pošty. Pre komunikáciu prostredníctvom POP3 protokolu sa využívajú štandardne porty 110 a 995. Port 110 slúži pre nešifrovanú komunikáciu, zatiaľ čo port 995 sa používa pre zabezpečenú komunikáciu. Jedná sa o výmenu správ medzi serverom a klientom[2].

2.1 Nešifrovaná komunikácia

Pri nešifrovanej komunikácii server štandardne počúva na porte 110[1]. Keď klient chce vstúpiť do mailovej schránky tak ako prvé sa nadviaže TCP spojenie so serverom. Pokiaľ toto spojenie prebehlo úspešne, tak server odpovedá uvítaciu správou a od tohoto momentu komunikácia prechádza do "Authorization state". V tomto stave je potrebné aby sa klient overil pomocou príkazov USER a PASS / APOP. Po úspešnom prihlásení prechádza komunikácia do "Transaction state". V tomto stave je možné pracovať s mailami: zistiť ich počet a veľkosť, čítať maily alebo označiť maily na vymazanie. Keď klient zadá príkaz QUIT komunikácia prejde do "Update state" kedy vymaže označené maily a zruší sa TCP spojenie[4].

2.2 Šifrovaná komunikácia

Šifrovaná komunikácia je veľmi podobná ako nešifrovaná s tým rozdielom, že server počúva na porte 995[1] a po nadviazaní TCP spojenia dochádza k overeniu (výmene) certifikátov medzi klientom a serverom, ktoré budú použité pre šifrovanie. Následujúce stavy a príkazy sú rovnaké ako pri nešifrovanej komunikácii.

3 Návrh a implementácia

Program popcl je štrukturovaný a rozdelený do viacerých funkcií. Pracuje so štandardnými C/C++ knižnicami a ďalej s knižnicami -lssl a -lcrypto. V main funkcii dochádza len k volaniu ďalších podfunkcií. Projekt som si dopredu rozdelil na jednotlivé podproblémy, ktoré som následne začal chronologicky riešiť.

3.1 Parsovanie argumentov

Ako prvé bolo potrebné riešiť parsovanie argumentov programu. Z mainu je volaná funkcia `arg_checker`, ktorá sa stará o toto spracovanie parametrov. Používa pomocnú funkciu `is_there`, ktorá vracia pozíciu na ktorej sa parameter nachádza a taktiež ho nahradí prázdny reťazcom. Pokiaľ funkcia parameter nenájde tak vracia hodnotu -1. Funkcia `arg_checker` si pomocou nej overí všetky možné prepínače, ktoré môže užívateľ zadať. Všetky prepínače sú uchovávané ako globálne premenné aby s nimi bolo možné pracovať vo všetkých funkciách.

3.2 Spracovanie autentizačného súboru

Ako ďalší krok bolo potrebné získať prihlasovacie meno a heslo zo súboru, ktorý bol zadaný užívateľom. K tomuto slúži funkcia `open_auth_file`, ktorá jednoducho overí či súbor existuje a následne ho otvorí. Pomocou funkcie `fscanf` sú následne načítavané jednotlivé reťazce a kontrolované podľa formátu v akom majú byť.

3.3 Nadviazanie spojenia so serverom

3.3.1 Nezabezpečené spojenie

Pokiaľ užívateľ nezadal parameter `-T` alebo `-S`, tak v tom prípade chceme nadviazať nezabezpečené spojenie so serverom. Na to slúži funkcia `start_unencrypted_connection`. Pokiaľ nám nebol zadaný port ako parameter tak potom použijeme štandardný port 110. Pripojenie je robené pomocou BIO knižnice, kde najskôr zavoláme funkciu `BIO_new_connect`, ktorá nám vytvorí spojenie a následne pomocou funkcie `BIO_do_connect` overíme či pripojenie bolo úspešné[3].

3.3.2 Zabezpečené spojenie

Ak užívateľ zadal parameter `-T`, tak nadviažeme zabezpečené spojenie so serverom. Z mainu je zavolaná funkcia `start_crypted_connection` v ktorej sa ako prvej nastaví štandardný port 995 ak nebol pri spustení zadaný nejaký iný. Ako prvé si vytvoríme `SSL_CTX` štruktúru, ktorú použijeme pre vytvorenie SSL spojenia pomocou funkcie `SSL_CTX_new`. Ďalším krokom je načítanie certifikátov. Pokiaľ užívateľ zadá parameter `-c` tak potom sa certifikát načíta pomocou funkcie `SSL_CTX_load_verify_locations`, pokiaľ zadá parameter `-C` tak sa overí zložka s certifikátmi pomocou rovnakej funkcie. Pokiaľ ani jeden z týchto parametrov zadaný nebol, tak sa použije funkcia `SSL_CTX_set_default_verify_paths`. Následne je potrebné vytvoriť spojenie so serverom pomocou funkcií `BIO_new_ssl_connect` a `BIO_do_connect`. Posledným krokom je overenie platnosti certifikátu pomocou funkcie `SSL_get_verify_result`. Pokiaľ toto všetko prebehlo úspešne tak máme vytvorené zabezpečené spojenie so serverom[3].

3.3.3 Nezabezpečené spojenie a príkaz STLS

Ak užívateľ zadal parameter `-S`, tak v tom prípade najskôr nadviažeme nezabezpečené spojenie na porte 110 (ak nebol zadaný iný port užívateľom) a následne odošleme na server príkaz STLS. Pokiaľ server STLS podporuje, tak v tom prípade vracia `+OK` a môžeme stávajúce spojenie vylepšiť na zabezpečené tak, že do bio štruktúry vložíme ssl pomocou funkcie `BIO_push()` [5] s tým, že je ešte pred tým potrebné pridať certifikáty pomocou CTX rovnako ako pri zabezpečenom spojení. Toto všetko spraví funkcia `start_stls_connection` volaná z mainu.

3.4 Autentizácia užívateľa

Ďalším krokom je autentizácia užívateľa. Zavolá sa funkcia `authenticate_user`, ktorá jednoducho zašle na server 2 príkazy. Jedným z nich je príkaz `USER`, pomocou ktorého pošleme serveru informáciu o prihlasovacom mene užívateľa za ktorým následuje príkaz `PASS`, pomocou ktorého zašleme heslo užívateľa.

3.5 Počet správ na serveri

Pre ďalšiu prácu potrebujeme zistiť koľko správ sa nachádza na serveri. Pomocou funkcie `get_number_of_messages` zašleme na server príkaz `STAT`, ktorý nás informuje o tomto počte, takže vieme s kolikatými správami budeme pracovať.

3.6 Ukladanie správ

3.6.1 Overenie úložiska mailov

Pred tým ako začneme prechádzať a ukladať maily, je potrebné overiť, či užívateľom zadaný súbor pre ukladanie mailov je validný. Pomocou funkcie `validating_outdir` overím či táto zložka existuje. Pokiaľ zadal neexistujúcu zložku tak program končí s chybou.

3.6.2 Prijímanie správ

Pre prijímanie správ je vytvorená obsiahlejšia funkcia `retrieving_messages`, ktorá obsahuje jeden for cyklus, ktorý prebehne toľko krát, koľko je správ na serveri. V ňom sa najskôr pošle príkaz `RETR` pre získanie danej správy, z ktorej si odrežeme Message-ID s ktorým bude neskôr pomenovaný súbor s daným mailom. Toto message ID je unikátne pre každý mail, takže pomocou neho je možné aj kontrolovať, ktoré správy sú nové. Vždy pri spracovaní mailu si do špeciálneho súboru vložíme jeho Message ID a keď bol zadaný parameter `-n` tak je kontrolované Message ID daného mailu so všetkými Message IDs v danom súbore. Pokiaľ sa tam už nachádza tak potom mail nieje nový a už bol prevzatý, pokiaľ tam nieje, v tom prípade vieme, že mail je nový. Takto je riešený parameter `-n`. Následne sa vezme string v ktorom je uložený obsah mailu a zavolá sa funkcia `create_file`, ktorá vytvorí nový súbor kde vloží obsah mailu vo formáte Internet Message Format[6].

3.7 Mazanie správ

Pokiaľ bol zadaný parameter `-d` pre vymazanie správ na serveri, tak sa zavolá funkcia `delete_messages`, ktorá jednoducho pomocou príkazu `DELE` vo for cykle odstráni všetky maily na serveri.

3.8 Ukončenie spojenia

Pre ukončenie spojenia so serverom je vytvorená funkcia `close_connection`, ktorá jednoducho zašle príkaz `QUIT`.

3.9 Výsledný výpis a uvoľnenie zdrojov

Poslednou vecou je výpis informácie o počte stiahnutých správ, poprípade počte nových stiahnutých správ alebo vymazaných správ a následuje uvoľnenie zdrojov pomocou funkcií `SSL_CTX_free` a `BIO_free_all`.

3.10 Pomocné funkcie

Program používa dve pomocné funkcie pre prijímanie správ zo servera a jednu pomocnú funkciu pre zasielanie príkazov na server.

3.10.1 Posielanie príkazov

Aby sme sa vyhli duplicitnému kódu tak je vytvorená funkcia pre zasielanie správ na server `write_to_server`, ktorá jednoducho zašle serveru príkaz pomocou funkcie `BIO_write`.

3.10.2 Prijímanie správ

Pre prijímanie správ zo servera sú implementované dve funkcie. Funkcia `read_from_server`, ktorá sa volá vždy keď potrebujeme prečítať správu od servera, ktorá není tzv. "multi-line". To znamená, že je ukončená CRLF znakmi. Pomocou while cyklu prijímame správy až pokiaľ nenatrafíme na ukončovacie znaky CRLF. Následne porovnáme či server zaslal +OK alebo -ERR. Ďalšiou pomocnou funkciou pre prijímanie správ je `read_retr`, ktorá je veľmi podobná, rozdiel je v tom, že sa volá vždy po zaslanom príkaze RETR, pretože ukončovací znak je v tomto prípade CRLF.CRLF, keďže sa jedná o tzv. "multi-line" správu.

4 Základné informácie o programe

Program je pomenovaný `popcl` a spúšťa sa ako:

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

kde:

<server> - IP adresa alebo názov servera

-p <port> - číslo portu na serveri

-T - zapne šifrovanie celej komunikácie (pop3s)

-S - nadviaže nešifrované spojenie a pomocou príkazu STLS sa zapne šifrovaná varianta protokolu

-c <certfile> - definuje súbor s certifikátmi použitých pre overenie platnosti certifikátu SSL/TLS predloženého serverom

-C <certaddr> - určuje adresár v ktorom sa majú vyhľadávať certifikáty, ktoré sa použijú pre overenie platnosti certifikátu SSL/TLS predloženého serverom

-d - zašle serveru príkaz pre vymazanie správ

-n - určuje, že sa bude pracovať iba s novými správami

-a <auth_file> - súbor, ktorý obsahuje údaje potrebné pre prihlásenie

-o <out_dir> - špecifikuje adresár, kde budú ukladané maily

-h - výpis nápovedy

4.1 Povinné parametre

Pre beh programu je potrebné zadať parameter <server>, parameter -a s prihlasovacími údajmi a parameter -o s adresárom, kde budú maily uložené.

4.2 Kombinácia parametrov

4.2.1 Parameter -S a -T

Nieje možné kombinovať parameter -S spolu s parametrom -T, pokiaľ k takémuto niečomu dôjde, tak program končí s chybou.

4.2.2 Parameter -c a -C

Nieje možné kombinovať parameter -c s parametrom -C, je potrebné zadať buď adresár s certifikátmi alebo konkrétny certifikát. Pokiaľ nieje zvolený ani jeden z parametrov tak sa použije funkcia `SSL_CTX_set_default_verify_paths` pre získanie úložiska s certifikátmi.

4.2.3 Parameter -d a -n

Pokiaľ sú zadané tieto dva parametre spolu, tak klient najskôr stiahne nové správy a následne vymaže všetky správy na serveri. Takže nedochádza k mazaniu len nových správ ale všetkých na serveri.

5 Testovanie skriptu






Program bol testovaný na viacerých serveroch, či už lokálne pomocou Hmailu ale taktiež aj pomocou služieb ako centrum.sk, seznam.cz alebo safe-mail.net. Program bol otestovaný taktiež aj na eve a merlinovi, kde fungoval podľa očakávania, až na to, že na merlinovi program pri zabezpečenom spojení spadne na prvej funkcii `SSL_CTX_new`, čo je pravdepodobne spôsobené staršou verziou SSL knižnice, ktorá sa na merlinovi nachádza.

5.1 Testovanie nezabezpečeného pripojenia

Na nasledujúcich obrázkoch si môžeme všimnúť malé demo, kde mám v mailovej schránke na začiatku celkovo 3 maily. Po ich stiahnutí sa pokúsim znova o stiahnutie ale s -n parametrom (takže chcem prijať iba nové správy). Keďže žiadne nové správy aktuálne nemám, tak program nestiahne žiadne maily. Keď si skúsim poslať nový mail a následne spustím program znova s -n parametrom tak môžeme vidieť, že bol stiahnutý 1 nový mail, takže -n parameter funguje tak ako má. Posledným skriptom je vymazanie správ na serveri, po ktorom naozaj zo servera sú správy vymazané. Maily sa nám uložili do zadaného adresára v príslušnom formáte.

```
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/
Stiahnute spravy: 3
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -n
Stiahnute nove spravy: 0
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -n
Stiahnute nove spravy: 1
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -n -d
Stiahnute nove spravy: 0 a vymazane: 4
```

Obrázek 1: Výstupy skriptu pri nezabezpečenom pripojení

/homes/eva/xz/xzauko00/ISA/Maildir/					
Name ^	Size	Changed	Rights	Own...	
		17. 10. 2021 11:...	rw-r-x...	xza...	
 <202110171...	2 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <202110171...	2 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <202110171...	2 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <202110171...	2 KB	17. 10. 2021 11:...	rw-r--...	xza...	






Obrázek 2: Obsah adresára Maildir - 4 maily podľa očakávania

5.2 Testovanie zabezpečeného pripojenia

Demo je podobné ako na príklade vyššie, no tu sa používa zabezpečená komunikácia pomocou parametra -T so seznamovým serverom. Znova sú zvolené rovnaké kroky ako vo vyššom príklade.

```
eva ~/ISA> ./popcl -T -a seznam -o Maildir/ pop3.seznam.cz
Stiahnute spravy: 3
eva ~/ISA> ./popcl -T -a seznam -o Maildir/ pop3.seznam.cz -n
Stiahnute nove spravy: 0
eva ~/ISA> ./popcl -T -a seznam -o Maildir/ pop3.seznam.cz -n
Stiahnute nove spravy: 1
eva ~/ISA> ./popcl -T -a seznam -o Maildir/ pop3.seznam.cz -n -d
Stiahnute nove spravy: 0 a vymazane: 4
eva ~/ISA> |
```

Obrázek 3: Výstupy skriptu pri zabezpečenom pripojení

/homes/eva/xz/xzauko00/ISA/Maildir/					
Name ^	Size	Changed	Rights	Own...	
		17. 10. 2021 11:...	rw-r-x...	xza...	
 <Shd.6jA}d.2...	1 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <She.6jA}Z.4...	2 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <Shf.6jA}l.6r...	1 KB	17. 10. 2021 11:...	rw-r--...	xza...	
 <Shj.6jA}h.1E...	1 KB	17. 10. 2021 11:...	rw-r--...	xza...	

Obrázek 4: Obsah adresára Maildir - 4 maily podľa očakávania

```

1 Received: from unknown ([2001:67c:1220:c1a2:1140:41:4e0e:d081])
2   by email.seznam.cz (szn-ebox-5.0.80) with HTTP;
3   Sun, 17 Oct 2021 11:30:18 +0200 (CEST)
4 From: <svoradaslavo@seznam.cz>
5 To: <svoradaslavo@seznam.cz>
6 Subject: qfqf
7 Date: Sun, 17 Oct 2021 11:30:18 +0200 (CEST)
8 Message-Id: <Shj.6jA}h.1E}E{uJegE}.1XQ{qg@seznam.cz>
9 Mime-Version: 1.0 (szn-mime-2.1.14)
10 X-Mailer: szn-ebox-5.0.80
11 Content-Type: text/plain;
12   charset=utf-8
13 Content-Transfer-Encoding: quoted-printable
14
15 ssss
16

```



Obrázek 5: Obsah súbora s mailom

5.3 Testovanie STLS

Testovanie STLS aj s výpisom vymenených správ zo strany klienta a servera aby bolo vidno, že k STLS dochádza.

```
eva ~/ISA> ./popcl -a seznam -o Maildir/ pop3.seznam.cz -S
+OK Hello, this is Seznam POP3 server unknown.
STLS
+OK Begin TLS negotiation
USER [REDACTED]
+OK Enter your password please.
PASS [REDACTED]
+OK 1 489
STAT
+OK 1 489
RETR 1
QUIT
+OK Closing connection, see you later.
Stiahnute spravy: 1
```

Obrázek 6: Komunikácia pri STLS príkaze

/homes/eva/xz/xzauko00/ISA/Maildir/					
Name ^	Size	Changed	Rights	Own...	
		19. 10. 2021 19:...	rwxr-x...	xza...	
 <UDX.6jA}y....	1 KB	19. 10. 2021 19:...	rw-r--...	xza...	

Obrázek 7: Obsah adresára Maildir

5.4 Testovanie správnosti uloženia mailu s bodkou na začiatku riadku

Bodka je zduplikovaná pokiaľ sa v maily nachádza na začiatku riadku, tým pádom bolo potrebné tento problém vyriešiť a nadbytočné bodky odstrániť.

```
1 DomainKey-Signature: a=rsa-sha1; q=dns; c=noaws;  
2 s=N1-0105; d=Safe-mail.net;  
3 b=VWgogKlMdHgbRsALxAQP+50G6i40kEJscM/+Je/sYi8/YfulGaMa001yuAmWxG4w  
4 m785SfhYaS2zUDuQEYxWcH+64wDupRb32yuKW34yCbNf1ak9S96vRWPzUJnmOSLk  
5 68F/9T2UvRn3eIJNjW09h4IyjOmyT1EUX/uHDCzfyBA=;  
6 Received: from pc ([147.229.217.10]) by Safe-mail.net with https  
7 Subject: aa  
8 Date: Sun, 17 Oct 2021 05:40:46 -0400  
9 From: testserver  
10 To: testserver@Safe-mail.net  
11 X-SMType: Regular  
12 X-SMRef: N10-CGTUj88bfp  
13 Message-Id: <N10-CGTUj88bfp@Safe-mail.net>  
14 MIME-Version: 1.0  
15 Content-Type: text/plain; charset=us-ascii  
16 Content-Transfer-Encoding: 7bit  
17 X-SMSignature: L/+uwz7iHFuSEnEi0/x5DPiQ2Ti0QxY/CCi452ViQoCjowzWQVH+UU5JWK+WQHnx  
18 +lwpLER/vlBkVUrmnCIRzgMme06pkGWWQDiaTKsLOTwb0kc1YZ+k4p3ew+KfAnqS  
19 CaR9NA47si8xYHvp+oSagPPw6ZvNldmBBU88YbzMuu8=  
20  
21 Test s bodkou na zaciatku riadku.  
22 .  
23 .  
24 .  
25
```

Obrázek 8: Obsah súboru s mailom s bodkami na začiatku riadku

5.5 Testovanie situácií, ktoré majú ukončiť program

5.5.1 Nesprávne prihlasovacie údaje

Pri zadaní zlých prihlasovacích údajov, nám server vracia ERR a vypisujeme následovnú chybovú hlášku a ukončujeme program.

```
eva ~/ISA> ./popcl pop3.centrum.sk -a seznam -o Maildir/  
Server returned -ERR during reading response.
```

Obrázek 9: Nesprávne prihlasovacie údaje

5.5.2 Zadanie neexistujúcej zložky pre ukladanie mailov

```
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o /neexistuje  
Specified outdir doesnt exist.
```

Obrázek 10: Neexistujúca zložka

5.5.3 Kombinácia -S a -T parametra

```
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -T -S  
Wrong parameters. Param T cannot be combined with param S.
```

Obrázek 11: Zadanie -S a -T parametrov súčasne

5.5.4 Neplatnosť certifikátov

```
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -T -c centrum  
Error while loading certificates.  
eva ~/ISA> ./popcl pop3.centrum.sk -a centrum -o Maildir/ -T -C Maildir/  
Error while loading certificates.
```

Obrázek 12: Zlyhanie overenia certifikátov

5.5.5 Pripojenie na zlý port

```
eva ~/ISA> time ./popcl pop3.centrum.sk -a centrum -o Maildir/ -p 1234
Error while connecting to server.

real    1m15,013s
user    0m0,002s
sys     0m0,007s
```

Obrázek 13: Pripojenie na zlý port

6 Záver

Projekt bol pre mňa veľmi prínosný a naučil som sa veľa o danej problematike ohľadom POP3 protokolu. Zistil som vďaka nemu ako prebieha internetová komunikácia prostredníctvom daného protokolu a taktiež som sa dozvedel veľa vecí ohľadom nadviazania zabezpečeného spojenia.

Literatura

- [1] Service Name and Transport Protocol Port Number Registry. [online], posledná úprava 12. Októbra 2021. Dostupné z: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>
- [2] Awati, R.: POP3 (Post Office Protocol 3). [online], publikované Október 2021. Dostupné z: <https://whatis.techtarget.com/definition/POP3-Post-Office-Protocol-3>
- [3] Ballard, K.: Secure programming with the OpenSSL API). [online], publikované 22. júl 2004. Dostupné z: <https://developer.ibm.com/tutorials/l-openssl/>
- [4] J. Myers, M. R.: Post Office Protocol - Version 3. [online], publikované Máj 1996. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc1939/>
- [5] Prikryl, M.: OpenSSL: Promote insecure BIO to secure one. [online], 6. Marca 2018. Dostupné z: <https://stackoverflow.com/questions/49132242/openssl-promote-insecure-bio-to-secure-one>
- [6] Resnick, P.: Internet Message Format. [online], publikované Október 2008. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc5322>