

VYSOKÉ UČENÍ TECHNICKÉ  
V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Sieťové aplikácie a správa sieti  
Klient POP3 s podporou TLS

9. listopadu 2021

Jozef Makiš

## Obsah

<b>1</b>	<b>Post Office Protocol - Version 3</b>	<b>2</b>
<b>2</b>	<b>SSL/TLS s protokolom POP3</b>	<b>2</b>
<b>3</b>	<b>Implementácia</b>	<b>3</b>
3.1	parse_args . . . . .	3
3.2	read_auth_file . . . . .	3
3.3	Pripojenie k serveru. . . . .	3
3.4	download_mails . . . . .	4
3.5	delete_messages . . . . .	5
3.6	Stiahnutie čisto nových správ a pomenovanie sprav bez predmetu.	5
<b>4</b>	<b>Spustenie programu.</b>	<b>5</b>
4.1	Príklady spustenia. . . . .	6
4.2	Ukážky komunikácie vo wiresharku. . . . .	8

## 1 Post Office Protocol - Version 3

POP3 je internetový protokol bežiaci na aplikačnej vrstve. Klient vďaka TCP/IP spojeniu získava elektronickú poštu zo servera s poštou. Protokol podporuje hlavné funkcie ako je sťahovanie elektronickej pošty prípadne jej mazanie zo servera. Využitie protokolu spočíva pri dočasnom pripojení na internet, kedy si užívateľ stiahne elektronickú poštu a neskôršie môže s ňou manipulovať už mimo pripojenia na internet. POP3 server čaká komunikáciu z portu 110. [3]

Po úspešnom nešifrovanom pripojení na server protokol umožňuje prejsť na šifrované spojenie cez príkaz STLS. Šifrovaná komunikácia môže prebiehať za pomoci protokolu POP3S, ktorý sa vie zabezpečené pripojiť na server cez protokoly SSL/TLS na porte 995.

Jedna z nevýhod je že klient prijme aj nevyžiadané správy a nemá možnosť filtrovania nevyžiadaných správ.

## 2 SSL/TLS s protokolom POP3

TLS poskytuje zabezpečenie pre aplikácie na aplikačnej vrstve napríklad pred odpočúvaním komunikácie. Prejsť na komunikáciu pod TLS je potrebné okamžite po nadviazaní komunikácie so serverom. Po úspešnom a overenom nadviazaní TLS prechodu je možné prejsť k overeniu a prihláseniu klienta. [2]

Klient so serverom dohodnú spojenie vďaka handshaku, počas ktorého sa dohodnú na parametroch zabezpečenej komunikácie. Pri prechode klient poskytne zoznam podporovaných šifier a transformačných funkcií. Server vyberie najsilnejšiu šifru a oznámi svoje rozhodnutie klientovi, tak isto odošle svoju identifikáciu z digitálneho certifikátu. Následne klient má možnosť overiť si certifikát pred tým než začne komunikovať. Server vygeneruje kľúč. Klient, ktorý vygeneruje kľúč pomocou verejného kľúča servera pošle výsledok serveru. Server pomocou súkromného kľúča ho dokáže dešifrovať, vďaka čomu sa zabezpečí že ku kľúčom majú prístup len klient a server. [4]

Hlavná zložka obsahujúca v certifikáte sú certifikačné authority. Sú to organizácie, ktoré majú nespochybniteľnú autoritu a majú všeobecne známe verejné kľúče. Certifikát môže obsahovať nasledujúce informácie. [1]

- Doménu na ktorej je certifikát nainštalovaný.
- Názov vlastníka.
- Mesto, alebo krajina registrácie.
- Informácie o certifikačnej autorite.
- Dôveryhodné a nedôveryhodné certifikáty.

- Sériové číslo SSL.

### 3 Implementácia

Program je rozdelený do funkcií, ktoré postupne spracovávajú vstup od užívateľa. Tok programu začína pri spracovaní argumentov príkazovej riadky, následne prihlásením užívateľa na základe súboru s týmito údajmi, s rôznym prístupom k zabezpečeniu komunikácie až po stiahnutie alebo mazanie správ zo servera. Základná hodnota portu cez ktorý sa klient pripája na server je 110.

#### 3.1 parse\_args

Spracovanie argumentov príkazovej riadky prebieha v tejto funkcii. Spracovanie prebieha nezávisle na poradí argumentov. Spracovanie prebieha v cykle cez obsah zadanych argumentov. Po zadaní argumentov, pri ktorých je potrebná cesta k súborom alebo zložkám sa spracuje ako cesta nasledujúca sekvencia znakov. V prípade že nebola sekvencia zadana a nasledujú iné argumenty program končí chybou. V prípade že boli zadané dve rôzne sekvencie, ktoré nepatria k parametrom programu dochádza k chybe. Prvá sekvencia znakov mimo argumentov s cestami k súborom je braná ako názov servera s elektronickou poštou.

#### 3.2 read\_auth\_file

Po úspešnom spracovaní argumentov program prejde k načítaniu prihlasovacích údajov na server. Program prehľadáva súbor po riadkoch, ale jediné čo ho zaujíma je či obsah riadku začína so slovami *username =* alebo *password =*. Za rovná sa musí nasledovať medzera po ktorej nasleduje reťazec čísel a znakov, v ktorých sa môže vyskytovať bodka, alebo podčiarkovník. Po zadaní sekvencií sa na riadkoch nesmie nič iné vyskytovať, inak to povedie k chybe. Takisto súbor na riadkoch nesmie nič iné obsahovať ako sekvenciu pre užívateľské meno, alebo heslo.

#### 3.3 Pripojenie k serveru.

Po inicializácii potrebných funkcií, potrebných pre využívanie knižnice OpenSSL program vetví pripojenie na server podľa zadanych argumentov, ktoré určujú zabezpečenie.

- *secured\_connect\_to\_server* je pripojenie na server za pomoci protokolu POP3S cez port 955. Program vytvorí ukazateľ, ktorý obsahuje informácie SSL protokolu, ktorý sa využíva k zabezpečeniu pripojenia na server cez BIO knižnicu.

Po vytvorení kontextovej štruktúry sa prejde k načítaniu certifikátov či už zadaných užívateľom, alebo základným certifikátom z OpenSSL súboru. Zadaním serveru sa overí odpoveď, či šifrované pripojenie prebehlo úspešne a spojenie bolo nadviazané. Po úspešnom nadviazaní šifrovanej komunikácie klient bezpečne pokračuje k prihlasovaniu užívateľa na server a následnom stiahnutí, alebo mazaní správ.

- *upgraded\_connection\_to\_server* je nezabezpečené pripojenie na server. Po pripojení na server sa pomocou komandu STLS klient okamžite pokúsi nadviazať zabezpečené spojenie. Po overení si odpovede servera na daný komand klient prejde na podobnú komunikáciu ako pri zabezpečenom spojení. Prechod je v programe možný vďaka tomu že sa alokuje nový ukazateľ s kontextom ako pri zabezpečenom pripojení popísanom vyššie. Vytvorí sa BIO socket v mode klienta, ktorý sa spojí so socketom, ktorý je nezabezpečený. Vďaka čomu je možné získať BIO SSL ukazateľ a pokračovať tak v zabezpečenej komunikácii.
- *connect\_to\_server* je nezabezpečená varianta bežiaci na porte 110. Pripojí sa na server pomocou jeho názvu a základného portu. Pripojenie sa overí, aby sa mohlo pokračovať ďalej v komunikácii.

OpenSSL dokumentácia odporúča použiť funkciu *c\_rehash* k overeniu formátu kľúčov. Tá počas implementácie nebola vložená do programu, pretože riešenie by nemuselo spoľahlivo bežať na referenčných strojoch melin a eva.

### 3.4 download mails

Pred tým ako sa začne postupne sťahovať elektronická pošta sa pomocou príkazu STAT overí počet správ. Táto informácia sa získa pomocou regulárneho výrazu kedy sa očakáva odpoveď vo formáte *+OK počet\_mailov veľkosť\_mailov*. V priestore na veľkosť celkovej elektronickej pošty môžu byť rôzne znaky, ktoré vyjadrujú aj jednotku číselného údaju. Ak sa údaj nepodaril získať program končí chybou.

Stiahnuté správy sú pomenované podľa ich predmetu.

Následne prebieha stiahnutie každého mailu samostatne. Obsah elektronickej pošty sa získava z odpovede servera vyvolanej príkazom RETR, ktorého parametrom je číselná hodnota daného mailu. Odpoveď servera sa ukladá do buffera pokiaľ sa neobjaví koniec odpovede reťazcom *\r \n. \r \n*. S uloženou správou sa prevádzajú nasledovné úpravy.

Prvou je odstránenie tzv. „byte-stuffingu“, ktorý na začiatku riadkov pridá extra bodku, ktorá pomáha rozoznať koniec odpovede.

Následne prebehne overenie odpovede servera že správa bola úspešne stiahnutá a jej odstránenie z odpovede, ktorá by bola spojená s obsahom mailu.

Dôležitou súčasťou je získanie ID z hlavičky mailu. ID slúži na overenie či daná správa už bola stiahnutá pri zadanom parametre pre stiahnutie iba nových správ.

Po úspešnom stiahnutí správ dochádza k ukončeniu spojenia pomocou príkazu QUIT.

### 3.5 delete\_messages

Pomocou tejto funkcie budú zmazané všetky elektronická pošta na serveri. Údaje o počte správ získajú pomocou príkazu STAT. Po prebehnutí mazania sa ukončí úspešné spojenie pomocou príkazu QUIT.

### 3.6 Stiahnutie čisto nových správ a pomenovanie správ bez predmetu.

Pri parametri -n prebieha sťahovanie nových správ. Počas behu programu sa vždy stiahne každá jedna správa, ktorá sa uloží do „bufferu“. Pomocou regulárneho výrazu sa získajú message-id danej správy. Následne v prípade že existuje súbor s rovnakým názvom ako sťahovaná správa, súbor sa tvorí a vyhľadá sa jeho message-id. V prípade že message-id je rovnaké daná správa sa do súboru neukladá. Týmto spôsobom sa detekujú staré a nové správy.

Slabina tohto riešenia je keď správa neobsahuje message-id. Správy neobsahujúce message-id sa sťahujú stále.

Správy bez predmetu sa pomenovávajú podľa nasledovného formátu:  
empty\_subject\_<číslo správy>.

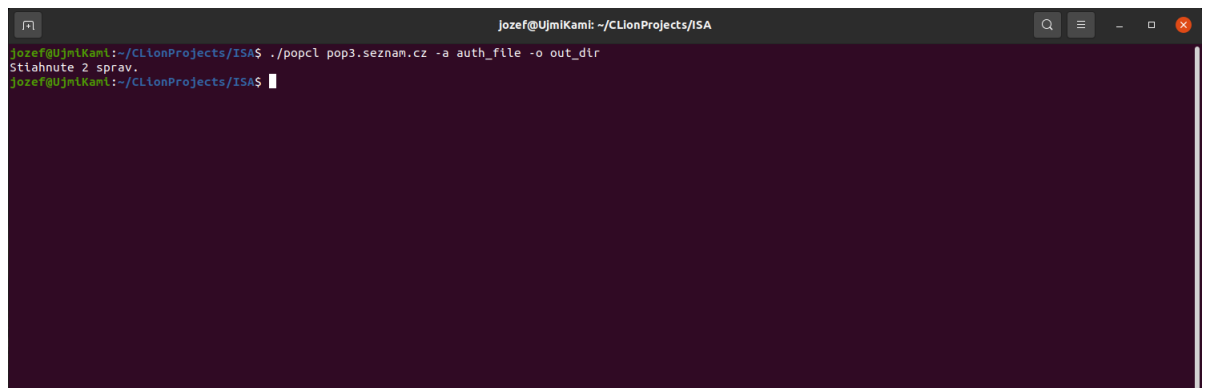
## 4 Spustenie programu.

Preloženie programu je možné za pomoci priloženého Makefile pomocou príkazu *make*. Program má implementované nasledujúce parametre.

- <názov servera> obsahujúci názov servera s elektronickou poštou.
- (-p <port >) voliteľný parameter portu pre server.
- (-d) voliteľný parameter, ktorý zmaže všetky správy na serveri.
- (-n) voliteľný parameter, ktorý stiahne iba nové správy na základe message-ID zo servera.
- (-a <autentifikačný súbor >) povinný parameter s obsahom užívateľského mena a hesla formátom viz *read\_auth\_file*.
- (-o <priečinko pre stiahnuté správy >) povinný parameter s cestou pre priečinko, ktorý bude obsahovať stiahnuté správy.

- (-S) nadviaže nešifrované spojenie, po ktorom prejde na šifrované za pomoci príazu STLS. Nesmie sa kombinovať s parametrom -T.
- (-T) nadviaže šifrované spojenie na porte 995 počas celého behu programu. Nesmie sa kombinovať s parametrom -S.
- (-c <súbor s certifikátom >) voliteľný parameter so súborom, ktorý obsahuje zvolený certifikát. Tento parameter môže byť zadaný samostatne a nezávisle na parametri priečinku s certifikátmi.
- (-C <priečinok s certifikátmi >) voliteľný parameter s priečinkom v ktorom sa nachádza daný certifikát. V prípade že nebol zvolený súbor s certifikátom dôjde k chybe !

#### 4.1 Príklady spustenia.

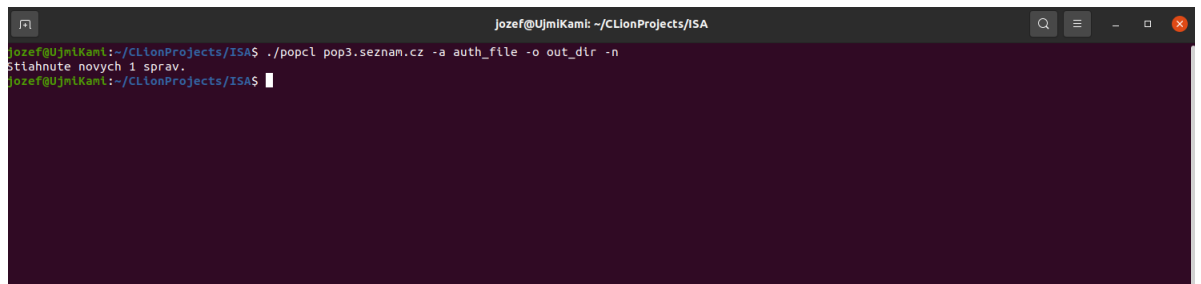


```

Jozef@UjmiKam: ~/CLionProjects/ISA
Jozef@UjmiKam:~/CLionProjects/ISA$ ./popcl pop3.seznam.cz -a auth_file -o out_dir
Stiahnute 2 sprav.
Jozef@UjmiKam:~/CLionProjects/ISA$

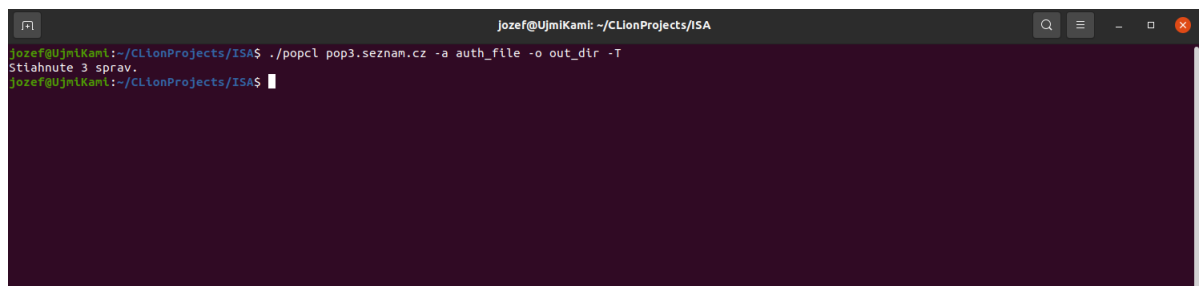
```

Obrázek 1: Nešifrované spustenie klienta.



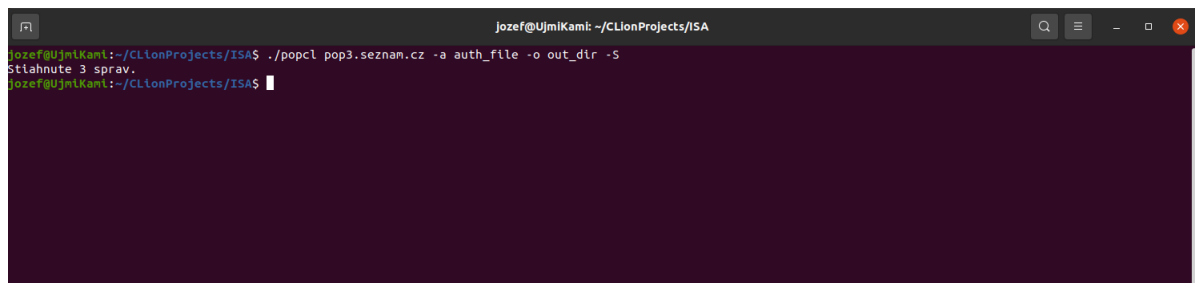
```
jozef@UjmiKamit: ~/CLionProjects/ISA
jozef@UjmiKamit:~/CLionProjects/ISA$ ./popcl pop3.seznam.cz -a auth_file -o out_dir -n
Stiahnute novych 1 sprav.
jozef@UjmiKamit:~/CLionProjects/ISA$
```

Obrázek 2: Stiahnutie nových správ.



```
jozef@UjmiKamit:~/CLionProjects/ISA$ ./popcl pop3.seznam.cz -a auth_file -o out_dir -T
Stiahnute 3 sprav.
jozef@UjmiKamit:~/CLionProjects/ISA$
```

Obrázek 3: Šifrovanie celej komunikácie.



```
jozef@UjmiKamit:~/CLionProjects/ISA$ ./popcl pop3.seznam.cz -a auth_file -o out_dir -S
Stiahnute 3 sprav.
jozef@UjmiKamit:~/CLionProjects/ISA$
```

Obrázek 4: Prechod zo šifrovanej na nešifrovanú komunikáciu.



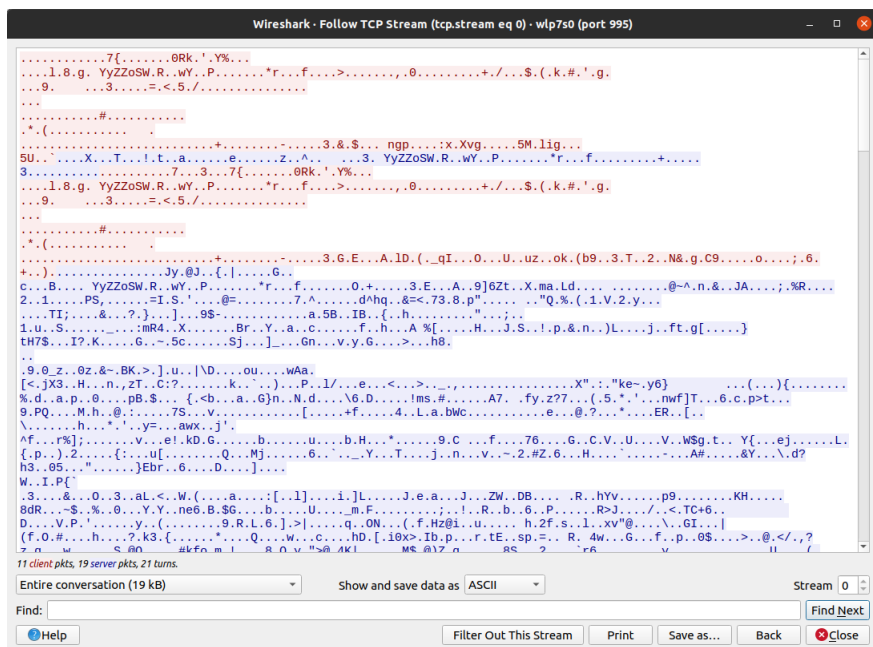
## 4.2 Ukážky komunikácie vo wiresharku.

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2021-11-07 19:00:02.303157834	192.168.1.19	77.75.78.46	TCP	74	53692 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 ...
2	2021-11-07 19:00:02.322941456	77.75.78.46	192.168.1.19	TCP	74	110 → 53692 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1412 S...
3	2021-11-07 19:00:02.322133452	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=231009602...
4	2021-11-07 19:00:02.338398214	77.75.78.46	192.168.1.19	POP	114	S: +OK Hello, this is Seznam POP3 server unknown.
5	2021-11-07 19:00:02.338463076	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=1 Ack=49 Win=64256 Len=0 TSval=23100960...
6	2021-11-07 19:00:02.353890846	77.75.78.46	192.168.1.19	POP	85	C: USER kamikadze420
7	2021-11-07 19:00:02.353890846	77.75.78.46	192.168.1.19	TCP	66	110 → 53692 [ACK] Seq=49 Ack=20 Win=29056 Len=0 TSval=2319668...
8	2021-11-07 19:00:02.366565698	77.75.78.46	192.168.1.19	POP	99	S: +OK Enter your password please.
9	2021-11-07 19:00:02.366608858	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=20 Ack=82 Win=64256 Len=0 TSval=2310096...
10	2021-11-07 19:00:02.366714400	192.168.1.19	77.75.78.46	POP	81	C: PASS isaproj1
11	2021-11-07 19:00:02.423841498	77.75.78.46	192.168.1.19	TCP	66	110 → 53692 [ACK] Seq=82 Ack=35 Win=29056 Len=0 TSval=2319668...
12	2021-11-07 19:00:03.504867690	77.75.78.46	192.168.1.19	POP	79	S: +OK 2 18107
13	2021-11-07 19:00:03.504897621	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=35 Ack=95 Win=64256 Len=0 TSval=2310097...
14	2021-11-07 19:00:03.504942110	192.168.1.19	77.75.78.46	POP	72	C: STAT
15	2021-11-07 19:00:03.520263161	77.75.78.46	192.168.1.19	TCP	66	110 → 53692 [ACK] Seq=95 Ack=41 Win=29056 Len=0 TSval=2319668...
16	2021-11-07 19:00:03.520645236	77.75.78.46	192.168.1.19	POP	79	S: +OK 2 18107
17	2021-11-07 19:00:03.521089550	192.168.1.19	77.75.78.46	POP	74	C: RETR 1
18	2021-11-07 19:00:03.538784026	77.75.78.46	192.168.1.19	POP	101	S: +OK Message follows (7260 bytes).
19	2021-11-07 19:00:03.540297666	77.75.78.46	192.168.1.19	POP/IMF	7329	from: <kamikadze420@seznam.cz>, subject: =?utf-8?q?Fwd=3A_V=C...
20	2021-11-07 19:00:03.540350302	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=49 Ack=7406 Win=58368 Len=0 TSval=23100...
21	2021-11-07 19:00:03.570706088	192.168.1.19	77.75.78.46	POP	74	C: RETR 2
22	2021-11-07 19:00:03.587988600	77.75.78.46	192.168.1.19	POP	101	S: +OK Message follows (2847 bytes).
23	2021-11-07 19:00:03.588938484	77.75.78.46	192.168.1.19	POP/IMF	2918	from: Jozef Makis <makisjozef28@gmail.com>, subject: .yikes, ...
24	2021-11-07 19:00:03.588985563	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=57 Ack=10293 Win=61952 Len=0 TSval=2310...
25	2021-11-07 19:00:03.600047656	192.168.1.19	77.75.78.46	POP	72	C: QUIT
26	2021-11-07 19:00:03.617663013	77.75.78.46	192.168.1.19	POP	106	S: +OK Closing connection, see you later.
27	2021-11-07 19:00:03.617878955	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [FIN, ACK] Seq=63 Ack=10333 Win=64128 Len=0 TSval...
28	2021-11-07 19:00:03.618035244	77.75.78.46	192.168.1.19	TCP	66	110 → 53692 [FIN, ACK] Seq=10333 Ack=63 Win=29056 Len=0 TSval...
29	2021-11-07 19:00:03.618066159	192.168.1.19	77.75.78.46	TCP	66	53692 → 110 [ACK] Seq=64 Ack=10334 Win=64128 Len=0 TSval=2310...
30	2021-11-07 19:00:03.633031741	77.75.78.46	192.168.1.19	TCP	66	110 → 53692 [ACK] Seq=10334 Ack=64 Win=29056 Len=0 TSval=2319...

Obrázek 5: Ukážka nešifrovanej komunikácie vo wiresharku.

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-11-07 19:07:36.210499207	192.168.1.19	77.75.78.46	TCP	74	49554 → 995 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 ...
2	2021-11-07 19:07:36.226116591	77.75.78.46	192.168.1.19	TCP	74	995 → 49554 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1412 S...
3	2021-11-07 19:07:36.226146336	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=231054992...
4	2021-11-07 19:07:36.226177136	192.168.1.19	77.75.78.46	TLSv1.3	349	Client Hello
5	2021-11-07 19:07:36.241662013	77.75.78.46	192.168.1.19	TCP	66	995 → 49554 [ACK] Seq=1 Ack=284 Win=39080 Len=0 TSval=2319713...
6	2021-11-07 19:07:36.242125956	77.75.78.46	192.168.1.19	TLSv1.3	165	Hello Retry Request, Change Cipher Spec
7	2021-11-07 19:07:36.242137953	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=284 Ack=100 Win=64256 Len=0 TSval=23105...
8	2021-11-07 19:07:36.252069639	192.168.1.19	77.75.78.46	TLSv1.3	388	Change Cipher Spec, Client Hello
9	2021-11-07 19:07:36.268211750	77.75.78.46	192.168.1.19	TLSv1.3	1466	Server Hello, Application Data
10	2021-11-07 19:07:36.268278660	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=606 Ack=1500 Win=64128 Len=0 TSval=2310...
11	2021-11-07 19:07:36.269057247	77.75.78.46	192.168.1.19	TCP	2762	995 → 49554 [PSH, ACK] Seq=1500 Ack=606 Win=31104 Len=2696 TS...
12	2021-11-07 19:07:36.269093037	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=606 Ack=4196 Win=63104 Len=0 TSval=2310...
13	2021-11-07 19:07:36.280432713	77.75.78.46	192.168.1.19	TLSv1.3	1079	Application Data, Application Data, Application Data
14	2021-11-07 19:07:36.280477073	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=606 Ack=5209 Win=64128 Len=0 TSval=2310...
15	2021-11-07 19:07:36.293741618	192.168.1.19	77.75.78.46	TLSv1.3	140	Application Data
16	2021-11-07 19:07:36.311124305	77.75.78.46	192.168.1.19	TLSv1.3	305	Application Data
17	2021-11-07 19:07:36.311174000	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=680 Ack=5448 Win=64128 Len=0 TSval=2310...
18	2021-11-07 19:07:36.311574626	77.75.78.46	192.168.1.19	TLSv1.3	305	Application Data
19	2021-11-07 19:07:36.311603722	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=680 Ack=5687 Win=64128 Len=0 TSval=2310...
20	2021-11-07 19:07:36.311906318	77.75.78.46	192.168.1.19	TLSv1.3	136	Application Data
21	2021-11-07 19:07:36.311929229	192.168.1.19	77.75.78.46	TCP	66	49554 → 995 [ACK] Seq=680 Ack=5757 Win=64128 Len=0 TSval=2310...
22	2021-11-07 19:07:36.321099448	192.168.1.19	77.75.78.46	TLSv1.3	107	Application Data
23	2021-11-07 19:07:36.327344263	77.75.78.46	192.168.1.19	TLSv1.3	121	Application Data
24	2021-11-07 19:07:36.327530198	192.168.1.19	77.75.78.46	TLSv1.3	103	Application Data
25	2021-11-07 19:07:36.383324906	77.75.78.46	192.168.1.19	TCP	66	995 → 49554 [ACK] Seq=5812 Ack=758 Win=31104 Len=0 TSval=2319...
26	2021-11-07 19:07:37.373899703	77.75.78.46	192.168.1.19	TLSv1.3	101	Application Data
27	2021-11-07 19:07:37.374079610	192.168.1.19	77.75.78.46	TLSv1.3	94	Application Data
28	2021-11-07 19:07:37.390033041	77.75.78.46	192.168.1.19	TCP	66	995 → 49554 [ACK] Seq=5847 Ack=786 Win=31104 Len=0 TSval=2319...
29	2021-11-07 19:07:37.390033281	77.75.78.46	192.168.1.19	TLSv1.3	101	Application Data
30	2021-11-07 19:07:37.391020205	192.168.1.19	77.75.78.46	TLSv1.3	96	Application Data
31	2021-11-07 19:07:37.435261041	77.75.78.46	192.168.1.19	TLSv1.3	123	Application Data

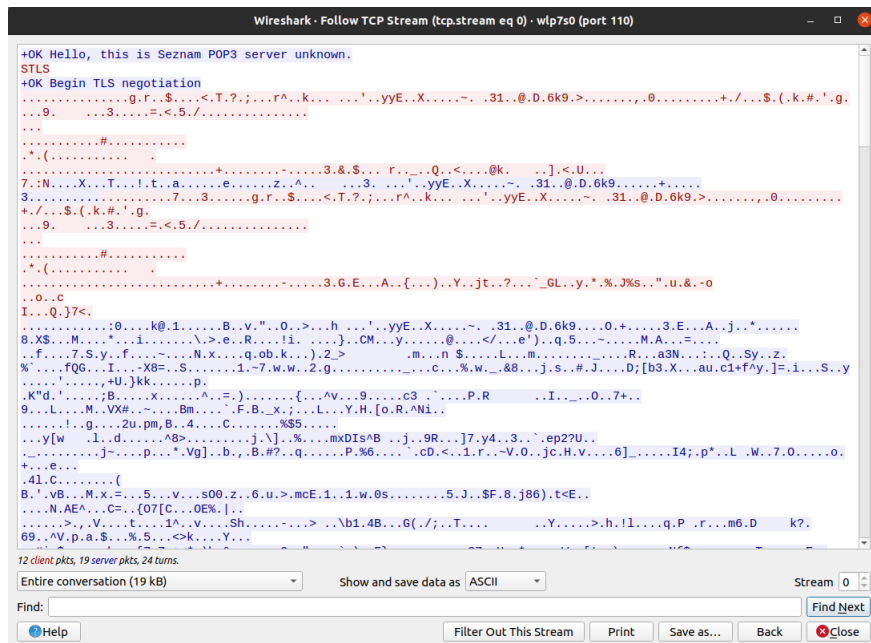
Obrázek 6: Ukážka kompletne šifrovanej komunikácie vo wiresharku.



Obrázek 7: TCP stream kompletne šifrovanej komunikácie vo wiresharku.

1	2021-11-07 19:10:11,100675174	192.168.1.19	77.75.78.46	TCP	74 54010 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2	2021-11-07 19:10:11,117626237	77.75.78.46	192.168.1.19	TCP	74 110 → 54010 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1412 S
3	2021-11-07 19:10:11,117658121	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=231070481..
4	2021-11-07 19:10:11,135257205	77.75.78.46	192.168.1.19	POP	114 S: +OK Hello, this is Seznam POP3 server unknown.
5	2021-11-07 19:10:11,135291843	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=1 Ack=49 Win=64256 Len=0 TSval=23107048..
6	2021-11-07 19:10:11,135350270	192.168.1.19	77.75.78.46	POP	72 C: STLS
7	2021-11-07 19:10:11,153997662	77.75.78.46	192.168.1.19	TCP	66 110 → 54010 [ACK] Seq=49 Ack=7 Win=29056 Len=0 TSval=24046086..
8	2021-11-07 19:10:11,154314577	77.75.78.46	192.168.1.19	POP	93 S: +OK Begin TLS negotiation
9	2021-11-07 19:10:11,154328846	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=7 Ack=76 Win=64256 Len=0 TSval=23107048..
10	2021-11-07 19:10:11,154610335	192.168.1.19	77.75.78.46	TLSv1.3	349 Client Hello
11	2021-11-07 19:10:11,169931981	77.75.78.46	192.168.1.19	TLSv1.3	165 Hello Retry Request, Change Cipher Spec
12	2021-11-07 19:10:11,169969269	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=290 Ack=175 Win=64256 Len=0 TSval=23107..
13	2021-11-07 19:10:11,170250280	192.168.1.19	77.75.78.46	TLSv1.3	388 Change Cipher Spec, Client Hello
14	2021-11-07 19:10:11,187115452	77.75.78.46	192.168.1.19	TLSv1.3	1466 Server Hello, Application Data
15	2021-11-07 19:10:11,187980321	77.75.78.46	192.168.1.19	TCP	2762 110 → 54010 [PSH, ACK] Seq=1575 Ack=612 Win=31104 Len=2696 TS..
16	2021-11-07 19:10:11,188038937	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=612 Ack=4271 Win=64128 Len=0 TSval=2310..
17	2021-11-07 19:10:11,198175826	77.75.78.46	192.168.1.19	TLSv1.3	1079 Application Data, Application Data, Application Data
18	2021-11-07 19:10:11,199407518	192.168.1.19	77.75.78.46	TLSv1.3	140 Application Data
19	2021-11-07 19:10:11,214769795	77.75.78.46	192.168.1.19	TLSv1.3	305 Application Data
20	2021-11-07 19:10:11,214797736	192.168.1.19	77.75.78.46	TLSv1.3	107 Application Data
21	2021-11-07 19:10:11,215226673	77.75.78.46	192.168.1.19	TLSv1.3	305 Application Data
22	2021-11-07 19:10:11,232617330	77.75.78.46	192.168.1.19	TLSv1.3	121 Application Data
23	2021-11-07 19:10:11,232690754	192.168.1.19	77.75.78.46	TCP	66 54010 → 110 [ACK] Seq=727 Ack=5817 Win=64128 Len=0 TSval=2310..
24	2021-11-07 19:10:11,232734874	192.168.1.19	77.75.78.46	TLSv1.3	103 Application Data
25	2021-11-07 19:10:11,289934371	77.75.78.46	192.168.1.19	TCP	66 110 → 54010 [ACK] Seq=5817 Ack=764 Win=31104 Len=0 TSval=2404..
26	2021-11-07 19:10:12,281454267	77.75.78.46	192.168.1.19	TLSv1.3	101 Application Data
27	2021-11-07 19:10:12,281536362	192.168.1.19	77.75.78.46	TLSv1.3	94 Application Data
28	2021-11-07 19:10:12,297214482	77.75.78.46	192.168.1.19	TCP	66 110 → 54010 [ACK] Seq=5852 Ack=792 Win=31104 Len=0 TSval=2404..
29	2021-11-07 19:10:12,297616275	77.75.78.46	192.168.1.19	TLSv1.3	101 Application Data
30	2021-11-07 19:10:12,298669604	192.168.1.19	77.75.78.46	TLSv1.3	96 Application Data
31	2021-11-07 19:10:12,316391647	77.75.78.46	192.168.1.19	TLSv1.3	123 Application Data

Obrázek 8: Ukážka prechodu na šifrovanu komunikáciu vo wiresharku.



Obrázek 9: TCP stream prechodu na šifrovanú komunikáciu vo wiresharku.

## **Použitá literatúra**

- [1] Gogetssl: What is SSL/TLS certificate. [online], [citované 6. 11. 2021].  
Dostupné z: <https://www.gogetssl.com/wiki/ssl-basics/what-is-ssl-tls/>
- [2] Newman, C.: Using TLS with IMAP, POP3 and ACAP. RFC 2595, June 1999, [citované 6. 11. 2021]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc2595>
- [3] Wikipedia: Post Office Protocol. [online], October 2021, [citované 6. 11. 2021]. Dostupné z: [https://en.wikipedia.org/wiki/Post\\_Office\\_Protocol](https://en.wikipedia.org/wiki/Post_Office_Protocol)
- [4] Wikipedia: Transport Layer Security. [online], január 2021, [citované 6. 11. 2021]. Dostupné z: [https://sk.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://sk.wikipedia.org/wiki/Transport_Layer_Security)