MTAA Zadanie 1

Branislav Hozza

FIIT STU

28.02.2022

# Zadanie

## Hlavná́ myšlienka zadania:

Na vašom počítači (alebo virtuálnom počítači) sprevádzkujte SIP Proxy, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami.

## Doplňujúce informácie k zadaniu:

Na implementáciu vašej SIP Proxy si môžete zvoliť akýkoľvek programovací jazyk a použiť akúkoľvek SIP knižnicu, ktorá pre daný programovací jazyk existuje. Vo výsledku však musíte spúšťať “váš kód”, v ktorom sú zakomponované knižnice, ktoré poskytujú funkcionalitu SIP Proxy. To znamená, že nemôžete zobrať existujúcu SIP Proxy ako napr. Asterisk, kde len skompilujete alebo priamo spustíte cudziu binárku... Hovor musí byť realizovaný medzi dvomi fyzickými zariadeniami v rámci LAN siete.

# Implementácia

Na implementáciu tohto zadania som použil programovací jazyk Python a základ implementácie som implementoval z tejto knižnice na Git Hube:

<https://github.com/tirfil/PySipFullProxy>

## Spojazdnenie knižnice

Následne som extrahoval hlavnú logiku obalenú v if \_\_name\_\_ == „\_\_main\_\_“ do separátneho súboru main.py a inicializoval som globálne premenné v knižnici potrebné na správny beh programu ako je napríklad register používateľov a začiatok hlavičky Via

Graphical user interface, text

Description automatically generated

Po úprave starej syntaxe tejto knižnice na moderný Python, som otestoval funkčnosť kódu pričom väčšina vecí fungovala ako mala.

## Denník hovorov

Následne som upravoval výpisy do denníka hovorov pretože pôvodná knižnica vypisovala príliš veľa zbytočných informácii. Na túto implementáciu som si musel dopísať vlastné funkcie ako napríklad getCallID() na získanie unikátneho označenia hovoru:

Text

Description automatically generated

alebo pickedUp() na zistenie či sa jedná o zodvihnutie hovoru:

Text

Description automatically generated

Následne výpisy v mojom proxy.log vyzerali nasledovne:



## Zmena stavových kódov

Okrem vypisovania do logu som sa snažil aj zmenu stavových kódov, respektíve o zmenu hlášky pri stavovom kóde. Čo sa týka odpovede na žiadosti tak to bolo ľahké, stačilo v pôvodnom kóde pozmeniť text pri metóde sendResponse():

Text

Description automatically generated

Ale čo sa týka stavových kódov, ktoré prichádzali ako odpovede od klienta, bolo ich potrebné zachytávať a následne pomeniť (táto logika nebolo pôvodne implementovaná v knižnici):

Text

Description automatically generated

A následne spracovanie:

Text

Description automatically generated

# Záver

Okrem správnej implementácie a refaktorizácii pôvodnej knižnice som spracoval aj PCAP súbory, ktoré sú priložene v repozitári a dokazujú funkčnosť môjho riešenia.

Fungoval mi normálny hovor, konferenčný hovor, video hovor, presmerovanie a logovanie s prepisom status kódov.

Pomocou tohto zadania som získal mnohé znalosti o SIP komunikácii.

# Odkaz na môj repozitár

<https://github.com/brano-hozza/xhozza-proxy-py>