**Slovenská technická univerzita v Bratislave**

Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Zadanie 1: Analyzátor sieťovej komunikácie

Obsah

[**Zadanie** 3](#_Toc148139085)

[**Diagramy spracovávania** 4](#_Toc148139086)

[**Analyzovanie rámcov** 4](#_Toc148139087)

[**Zjednodušený priebeh programu** 5](#_Toc148139088)

[**Fungovanie riešenia** 5](#_Toc148139089)

[**Fungovanie úloh 1-3** 5](#_Toc148139090)

[**Fungovanie jednotlivých filtrov** 6](#_Toc148139091)

[**TCP komunikácia** 6](#_Toc148139092)

[**UDP komunikácia** 6](#_Toc148139093)

[**ARP komunikácia** 6](#_Toc148139094)

[**ICMP komunikácia + fragmentácia** 6](#_Toc148139095)

[**Štruktúra externého súboru** 6](#_Toc148139096)

[**Používateľské rozhranie** 7](#_Toc148139097)

[**Voľba implementačného prostredia** 8](#_Toc148139098)

# **Zadanie**

Navrhnite analyzátor sieťovej komunikácie, kde program dostane na vstupe .pcap súbor a druh filtru podľa ktorého bude program analyzovať pakety a komunikácie. Komunikácie následne rozdelí do kompletných a čiastočných podľa zadaného filtra.

# **Diagramy spracovávania**

## **Analyzovanie rámcov**

Obrázok, na ktorom je snímka obrazovky, diagram, text, pestrofarebnosť

Automaticky generovaný popis

## **Zjednodušený priebeh programu**

Obrázok, na ktorom je text, snímka obrazovky, diagram, dizajn

Automaticky generovaný popis

# **Fungovanie riešenia**

## **Fungovanie úloh 1-3**

Do funkcie sa posielajú rámce, ktoré sa následne spracujú a ich všetky hodnoty sa zapíšu do slovníka a potom do súboru yaml. Do funkcie sa ešte posiela číslo paketu a pri výpise ICMP komunikácie aj flag či je rámec fragmentovaný alebo nie. Úloha 3 sa vypisuje iba pri úlohe vypisovania rámcov.

## **Fungovanie jednotlivých filtrov**

Všetky filtre majú podobnú logiku, kde sa najprv roztriedia rámce s rovnakým protokolom do polí. Potom sa rámce rozdelia do jednotlivých komunikácií podľa zadania a následne sa zistí, či daná komunikácia patrí do kompletnej alebo nie a vypíšu sa.

### **TCP komunikácia**

Pri TCP sa delia rámce podľa IP adries a portov a následne sa najprv kontroluje, či je komunikácia začatá pomocou flagov v rámci a potom koniec. Začiatok môže nastať 2 spôsobmi. Prvý je „three-way handshake“ a druhý, že obe strany pošlú naraz príznak SYN.

Ukončenie môže nastať 3 spôsobmi a to príznakom RST, naraz si pošlú príznak FIN alebo jedna strana pošle FIN a druhá mu odpovie, dostane naspäť ACK a komunikáciu ukončí. Potom už iba vypíšeme jednotlivé komunikácie do kompletných a prvú nekompletnú

### **UDP komunikácia**

Pri UDP protokole sme pracovali s TFTP komunikáciou na porte 69. Keďže ide o connectionless protokol tak sme brali kompletnú komunikáciu, takú kde je posledný prijatý datagram s príznakom ACK. Ostatné berieme ako nekompletné.

### **ARP komunikácia**

Rámce rozdeľujeme po pároch podľa IP adries a „opcode“. Ako kompletné považujeme pár REQUEST – REPLY. Do 2 nekompletných vypíšeme všetky REQUESTY a REPLY zvlášť bez páru.

### **ICMP komunikácia + fragmentácia**

Pri tomto filtri sme najprv rozdelili rámce do dvojíc REQUEST – REPLY podľa IP adries, ID a SEQ čísel. Potom sme kontrolovali, či sú komunikácie kompletné alebo nie. V prípade, kde sa odpovie na REQUEST TIME EXCEED je komunikácia považovaná za nekompletnú. Ďalej sa kontroluje, či je rámec fragmentovaný. Po vytvorení komunikácií ešte pozrieme komunikácie s rovnakými ID a IP adresami a spojíme ich. Následne ich vypisujeme a rámce bez odpovede alebo s neplatnou odpoveďou vypisujeme ako nekompletné.

# **Štruktúra externého súboru**

V externom súbore mam uložené protokoly a flagy s ich decimálnymi hodnotami, ktoré program načíta do slovníkov. Súbor je vo formáte .txt a má nasledovnú formu, kde „LLC:“, “IPPROTOCOL:“,... riadky slúžia ako rozdeľovač (na obrázku je iba časť súboru):

Obrázok, na ktorom je text, snímka obrazovky, písmo, dizajn

Automaticky generovaný popis

# **Používateľské rozhranie**

Používateľ spúšťa program cez menu, ktoré najprv musí dostať cestu k súboru a potom typ úlohy akú má vykonať. Pri zadaní „1“ sa program ešte opýta, či má vypísať aj úlohu 3 na výpis IP adries.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

# **Voľba implementačného prostredia**

Zvolil som si IDE Visual Studio Code, kvôli rozsiahlej možnosti upravovania prostredia s rozšíreniami. Zadanie je písane v jazyku Python s použitím knižníc ako binascii, os, scapy a yaml.