SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

ak. rok 2021/22, zimný semester

Analyzátor sieťovej komunikácie

Cvičiaci:
Ing. Peter Kaňuch

Vypracoval:
Roman Bitarovský

Obsah

Zadanie

Znenie zadania prikladám v pôvodnom znení v priloženom PDF súbore a sem uvádzam iba základný opis z tohto zadania.

Navrhnite a implementujte programový analyzátor Ethernet siete, ktorý analyzuje komunikácie v sieti zaznamenané v .pcap súbore a poskytuje nasledujúce informácie o komunikáciách. Vypracované zadanie musí spĺňať nasledujúce body:
1) **Výpis všetkých rámcov v hexadecimálnom tvare** postupne tak, ako boli zaznamenané v súbore.

Pre každý rámec uveďte:
a) Poradové číslo rámca v analyzovanom súbore.

b) Dĺžku rámca v bajtoch poskytnutú pcap API, ako aj dĺžku tohto rámca prenášaného po médiu.

c) Typ rámca – Ethernet II, IEEE 802.3 (IEEE 802.3 s LLC, IEEE 802.3 s LLC a SNAP, IEEE 802.3 – Raw).

d) Zdrojovú a cieľovú fyzickú (MAC) adresu uzlov, medzi ktorými je rámec prenášaný.

Vo výpise jednotlivé **bajty rámca usporiadajte po 16 alebo 32 v jednom riadku**. Pre prehľadnosť výpisu je vhodné použiť neproporcionálny (monospace) font.
2) Pre rámce typu **Ethernet II a IEEE 802.3 vypíšte vnorený protokol**. Študent musí vedieť vysvetliť, aké informácie sú uvedené v jednotlivých rámcoch Ethernet II, t.j. vnáranie protokolov ako aj ozrejmiť dĺžky týchto rámcov.

3) Analýzu cez vrstvy vykonajte pre rámce Ethernet II a protokoly rodiny TCP/IPv4:

Na konci výpisu z bodu 1) uveďte pre IPv4 pakety:
a) Zoznam IP adries všetkých odosielajúcich uzlov,

b) IP adresu uzla, ktorý sumárne odoslal (bez ohľadu na prijímateľa) najväčší počet paketov a koľko paketov odoslal (berte do úvahy iba IPv4 pakety).

IP adresy a počet odoslaných / prijatých paketov sa musia zhodovať s IP adresami vo výpise Wireshark -> Statistics -> IPv4 Statistics -> Source and Destination Addresses.
4) V danom súbore analyzujte komunikácie pre zadané protokoly:

a) HTTP

b) HTTPS

c) TELNET

d) SSH

e) FTP riadiace

f) FTP dátové

g) TFTP, **uveďte všetky rámce komunikácie**, nielen prvý rámec na UDP port 69

h) ICMP, uveďte aj typ ICMP správy (pole Type v hlavičke ICMP), napr. Echo request, Echo reply, Time exceeded, a pod.

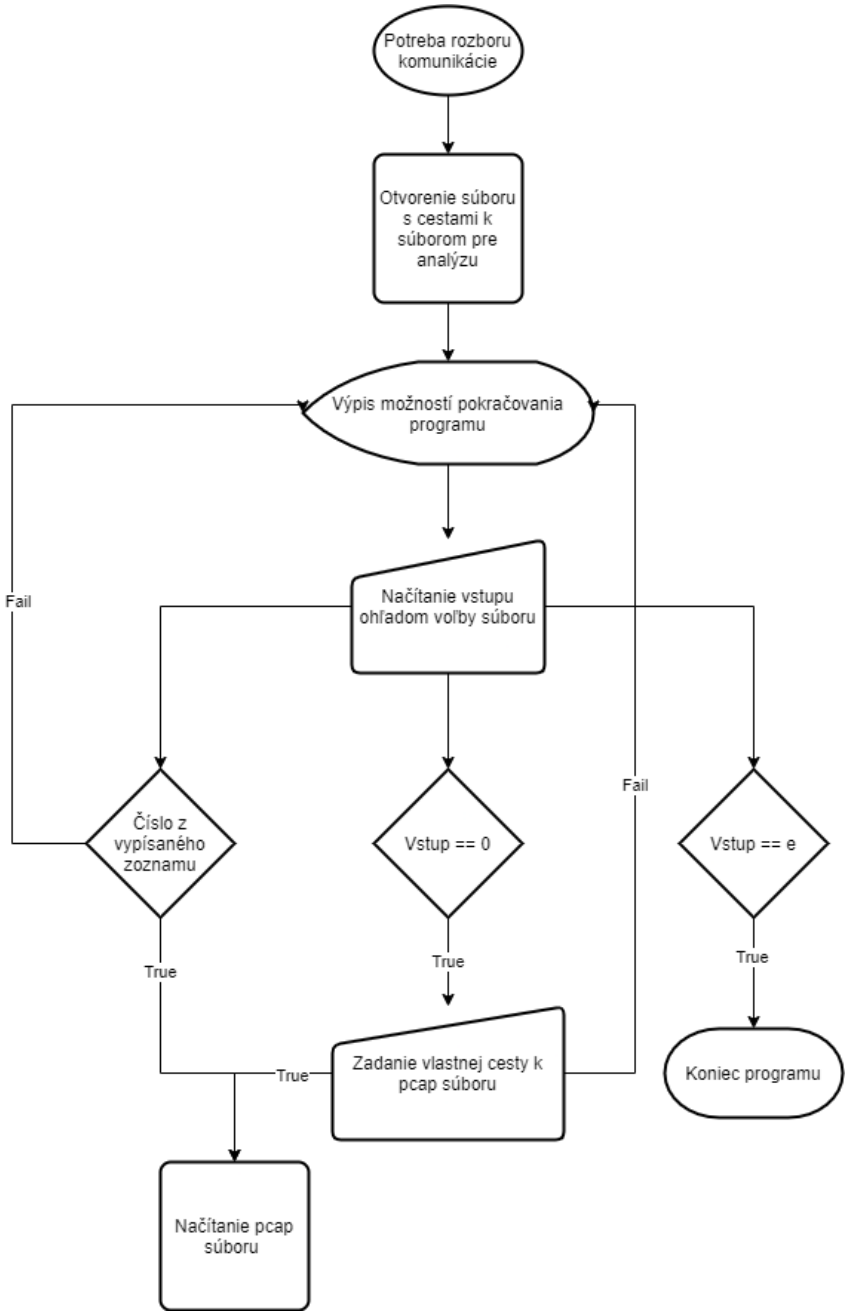
i) **Všetky** ARP dvojice (request – reply), uveďte aj IP adresu, ku ktorej sa hľadá MAC (fyzická) adresa a pri ARP-Reply uveďte konkrétny pár - IP adresa a nájdená MAC adresa. V prípade, že bolo poslaných viacero rámcov ARP-Request na rovnakú IP adresu, vypíšte všetky. Ak sú v súbore rámce ARP-Request bez korešpondujúceho ARP-Reply (alebo naopak ARP-Reply bez ARP-Request), vypíšte ich samostatne.

Vo všetkých výpisoch treba uviesť aj IP adresy a pri transportných protokoloch TCP a UDP aj porty komunikujúcich uzlov.
V prípadoch komunikácií so spojením vypíšte iba jednu kompletnú komunikáciu - obsahuje otvorenie (SYN) a ukončenie (FIN na oboch stranách alebo ukončenie FIN a RST alebo ukončenie iba s RST) spojenia a aj prvú nekompletnú komunikáciu, ktorá obsahuje iba otvorenie spojenia. Pri výpisoch vyznačte, ktorá komunikácia je kompletná.
Ak počet rámcov komunikácie niektorého z protokolov z bodu 4 je väčší ako 20, vypíšte iba 10 prvých a 10 posledných rámcov tejto komunikácie. **(Pozor: toto sa nevzťahuje na bod 1, program musí byť schopný vypísať všetky rámce zo súboru podľa bodu 1.)** Pri všetkých výpisoch musí byť poradové číslo rámca zhodné s číslom rámca v analyzovanom súbore.

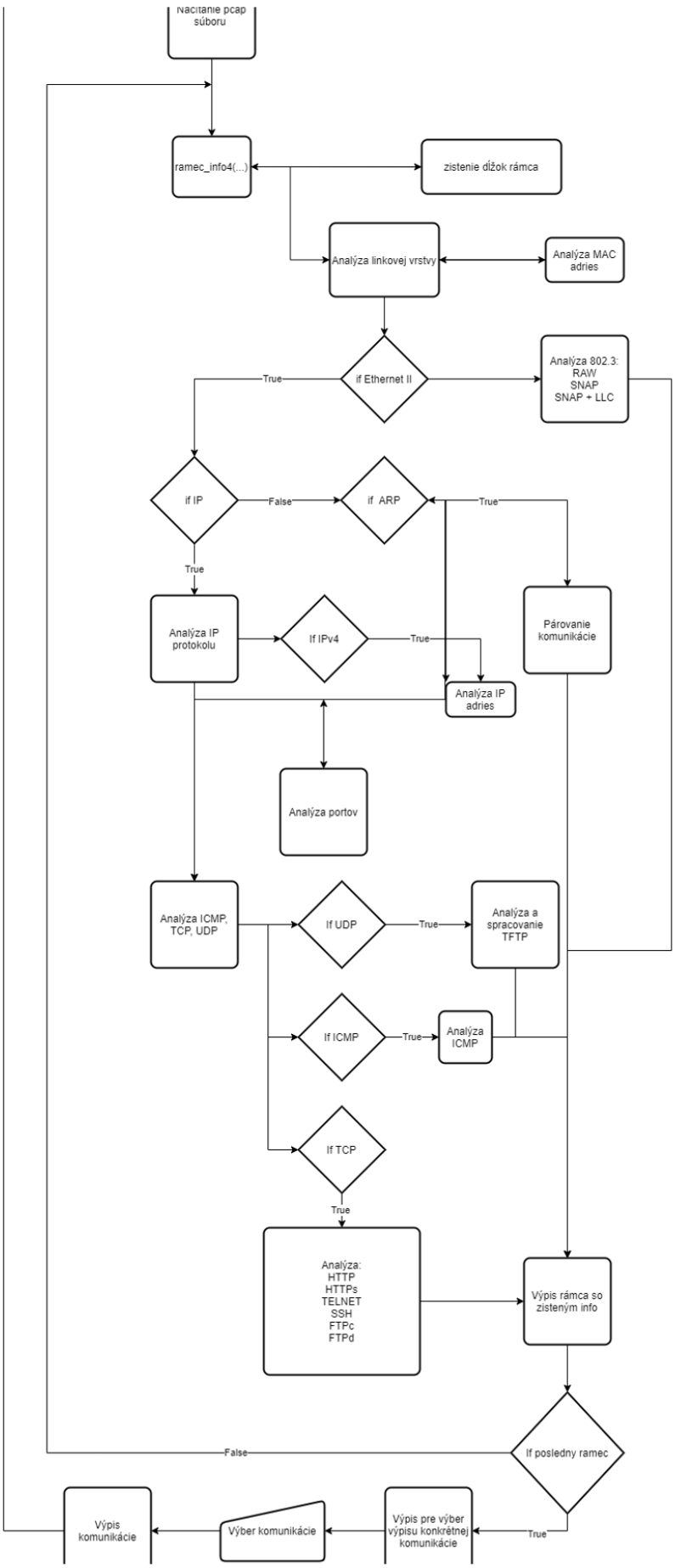
Blokový návrh fungovania riešenia

1. Riadiaca a používateľská časť

Tento diagram znázorňuje počiatočné voľby používateľa konkrétne a hlavne to aký spôsobom vieme nechať program analyzovať pcap súbor podľa našich požiadaviek. Rovnako tu je aj možnosť ukončiť program.



2. Analýza komunikácie



Blokový diagram pre analýzu komunikácie približne znázorňuje kroky (postup) pracovania programu pri analýze komunikácie smerom od linkovej vrstvy nahor.

3. Opis mechanizmu

Načítanie dát

Ak preskočíme riadiace náležitosti programu kt. som spomínal vyššie dostávame sa k prvému kroku pre analýzu a to je načítanie .pcap súboru. Nato používam funkciu rdpcap() z knižnice scapy. Následne si dáta prevedieme byte string pomocou funkcie raw() z kt. už vieme lepšie čítať dáta a po prevode ich porovnávať s hexadecimálnymi hodnotami alebo čítať hexadecimálne hodnoty.
Pomocou funkcie len() zmeriame dĺžku rámca. Následne zo 12teho alebo 14teho „bajtu“ čítame a porovnávaním zisťujeme typ rámca (Ethernet II, 802.3 RAW, 802.3 LLC + SNAP, 802.3 LLC).

MAC adresy

Keď už poznáme dĺžku rámca a typ rámca zisťujeme MAC adresy (zdrojovú a cieľovú). Zdrojovú zo 12:24 a Cieľovú z 0:12 avšak tento krát čítame z hexastringu teda na byte string sme aplikovali ešte .hex().

Vnorený protokol (tretia (sieťová) vrstva)

V určujeme či sa jedná o rodinu IP a teda protokoly IPv4, IPv6 alebo okrem „rodiny“ IP protokolov vieme určiť ak sa jedná o ARP komunikáciu.

- Ak nájdeme protokol z rodiny IP, konkrétne IPv4 tak určujeme aj IP adresy
- Ak sme určili ARP tak ďalej budeme určovať či sa jedná o request alebo reply

Určenie IP adries

```
245 def find_IP(raw_ramec):
246     source_ip = str(int(raw_ramec[26:27].hex(), 16)) + "." + str(int(raw_ramec[27:28].hex(), 16)) + "." + str(int(raw_ramec[28:29].hex(), 16)) + "." + str(int(raw_ramec[29:30].hex(), 16))
247     destination_ip = str(int(raw_ramec[30:31].hex(), 16)) + "." + str(int(raw_ramec[31:32].hex(), 16)) + "." + str(int(raw_ramec[32:33].hex(), 16)) + "." + str(int(raw_ramec[33:34].hex(), 1
248     return source_ip, destination_ip
```

Je čítanie konkrétnych bajtov a postupné vyskladanie do formátu IPv4 adresy.

ARP určenie request | reply

```
"operation": int(raw_ramec_hex[20*2:22*2]),
```

Ak je hodnota 1 tak to je request, ak je hodnota 2 tak sa jedná o reply

Určenie protokolu na 4 (transportnej) vrstve pre protokol IPv4 na tretej vrstve

Ak sme detekovali protokol IPv4 tak ďalej podľa slovníka protokolov hľadáme a snažíme sa určiť či sa jedná o jednu z nasledovných komunikácií:

- ICMP určujeme echo request, echo reply
- TCP v tejto komunikácií ďalej určujeme ďalšie protokoly podľa portov
- UDP ak je jeden z portov 69 tak sa jedná o TFTP kde ďalej budeme hľadať komunikáciu

Protokoly pri TCP komunikácii

Určujú sa zase podľa slovníka protokolov pre kategóriu TCP.

Hlavne protokoly kt. hľadáme sú:

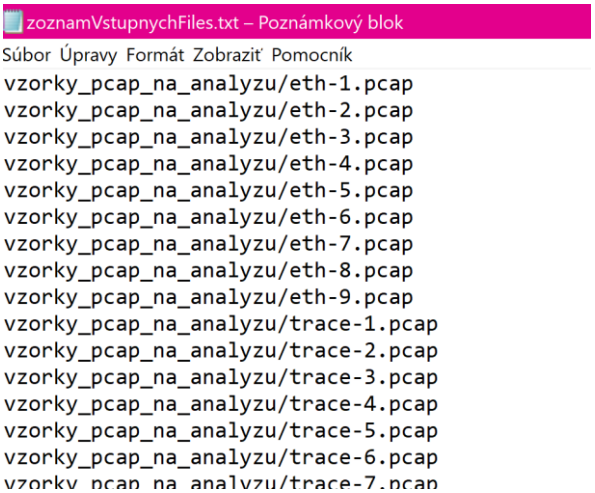
- HTTP
- HTTPs
- TELNET
- SSH
- FTP control
- FTP data

Tieto protokoly určujeme podľa všeobecne známych portov na kt. by podľa štandardov mali komunikovať.

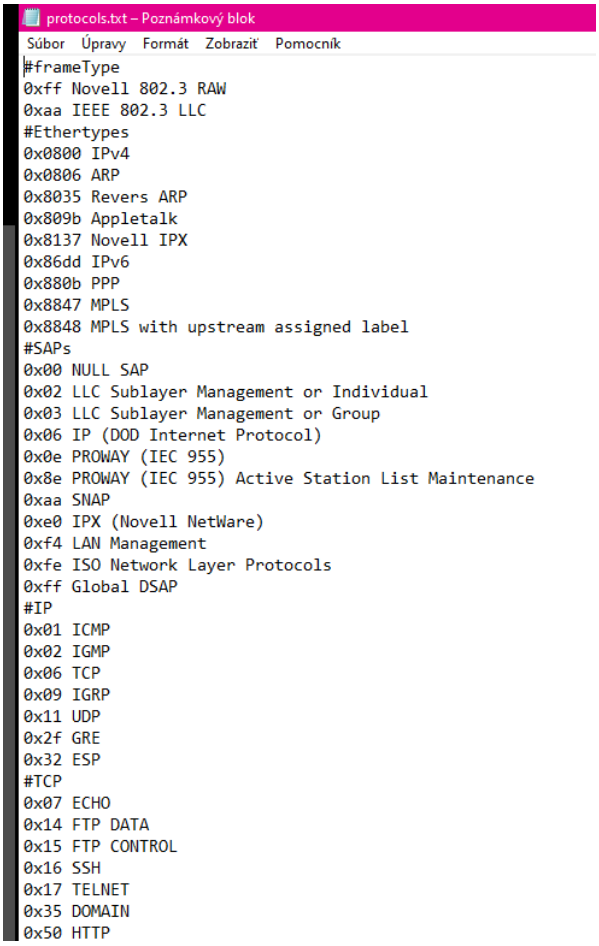
Príklad štruktúry externých súborov

Program používa dva resp. tri externé súbory.

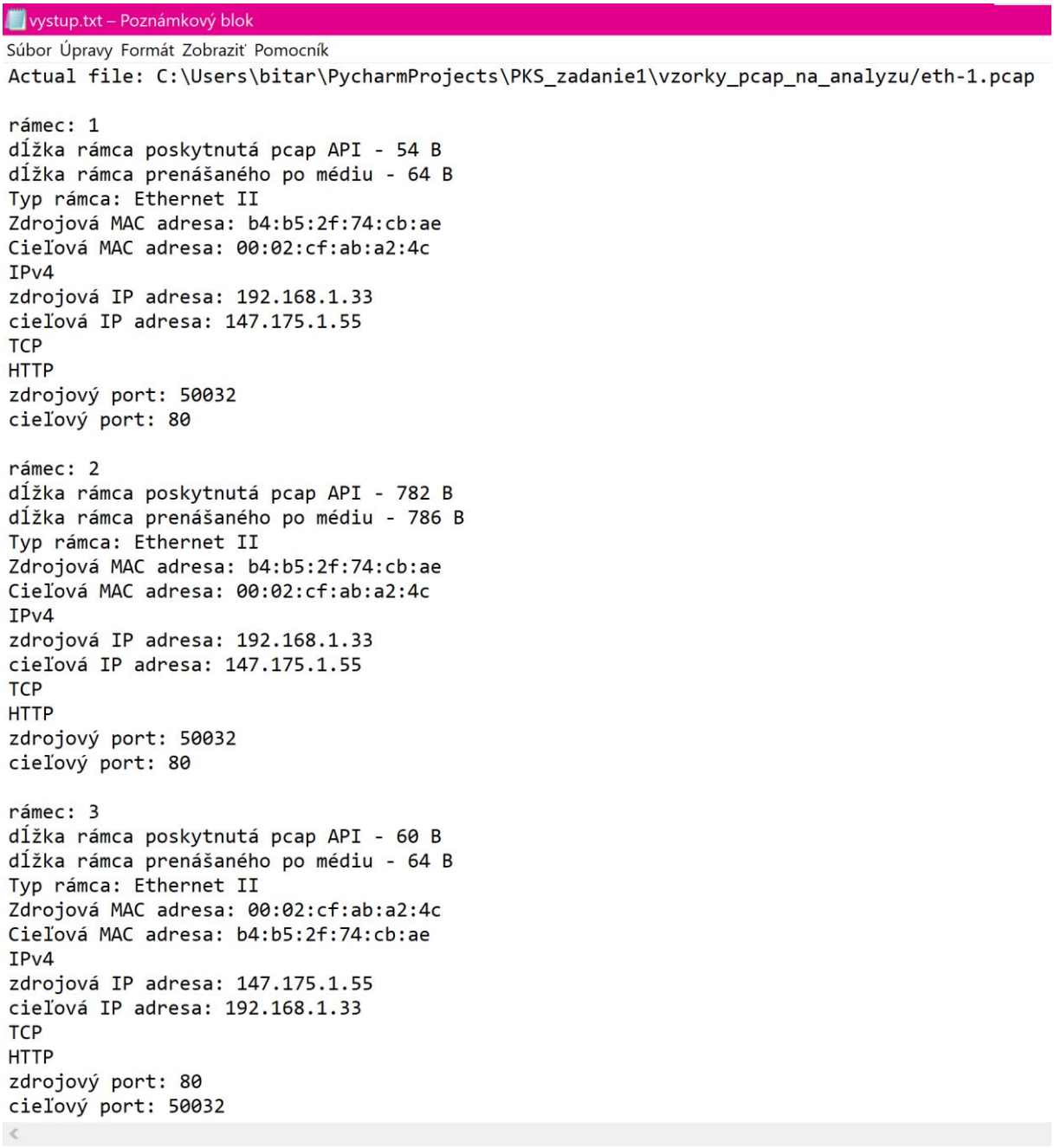
Prvý „*zoznamVstupnychFiles.txt*“ kde sú uložené relatívne cesty k pcap súborom, kt. sú v programu čítané a na základe kt. je následne vypásané menu kde používateľ môže vyberať, kt. súbor chce nechať analyzovať.



Druhý súbor, kt. je programom využívaný je „*protocols.txt*“. z toho súboru je vytvorený slovník podľa kategórií kde každý záznam obsahuje hexa hodnotu a názov príslušného protokolu a na základe týchto dát sa potom určujú protokoly v analyzovanej komunikácii.



Posledným, tretím súborom, kt. program využíva je „*vystup.txt*“, kt. ako názov napovedá slúži na výpis výstupu.


vystup.txt – Poznámkový blok

Súbor Úpravy Formát Zobraziť Pomocník

```
Actual file: C:\Users\bitar\PycharmProjects\PKS_zadanie1\vzorky_pcap_na_analyzu/eth-1.pcap

rámec: 1
dĺžka rámca poskytnutá pcap API - 54 B
dĺžka rámca prenášaného po médiu - 64 B
Typ rámca: Ethernet II
Zdrojová MAC adresa: b4:b5:2f:74:cb:ae
Cieľová MAC adresa: 00:02:cf:ab:a2:4c
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP
HTTP
zdrojový port: 50032
cieľový port: 80

rámec: 2
dĺžka rámca poskytnutá pcap API - 782 B
dĺžka rámca prenášaného po médiu - 786 B
Typ rámca: Ethernet II
Zdrojová MAC adresa: b4:b5:2f:74:cb:ae
Cieľová MAC adresa: 00:02:cf:ab:a2:4c
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP
HTTP
zdrojový port: 50032
cieľový port: 80

rámec: 3
dĺžka rámca poskytnutá pcap API - 60 B
dĺžka rámca prenášaného po médiu - 64 B
Typ rámca: Ethernet II
Zdrojová MAC adresa: 00:02:cf:ab:a2:4c
Cieľová MAC adresa: b4:b5:2f:74:cb:ae
IPv4
zdrojová IP adresa: 147.175.1.55
cieľová IP adresa: 192.168.1.33
TCP
HTTP
zdrojový port: 80
cieľový port: 50032
```

Opis používateľského rozhrania

Ako používateľské rozhranie sme zvolili konzolu. Nakoľko sa v tomto prostredí jednoducho realizuje vstup aj výstup od a pre používateľa.

Ako prvé sa nám vypíše začiatok programu v podobe ** *PyCharm starting..* ** a následne možnosti pre výber súboru určeného na analýzu.

```
** PyCharm starting.. **
PCAP_FILES_LIST is: C:\Users\bitar\PycharmProjects\PKS_zadanie1\zoznamVstupnychFiles.txt
1: vzorky_pcap_na_analyzu/eth-1.pcap
2: vzorky_pcap_na_analyzu/eth-2.pcap
3: vzorky_pcap_na_analyzu/eth-3.pcap
4: vzorky_pcap_na_analyzu/eth-4.pcap
```

```
36: vzorky_pcap_na_analyzu/trace-27.pcap
37: vzorky_pcap_na_analyzu/trace_ip_nad_20_B.pcap
Pre ukoncenie programu napis: e
Pre vlastnu cestu k suboru zadaj: 0
Pre vyber cisla suboru od 1 do 37(vratane)
```

Za výpiskom súborov pre analýzu nasleduje výpisok aké ma používateľ možnosti pre voľbu akou sa bude program ďalej smerovať.

- Ukončenie programu
- Zadanie vlastnej cesty k súboru pre analýzu
- Vybrať číslo pre analýzu súboru z ponúkaných súborov

Po voľbe súboru a jeho spracovaniu a analýze programom ma používateľ možnosť vybrať výpis konkrétnej komunikácie alebo určitého typu komunikácie.

```
Pre vyber cisla suboru od 1 do 37(vratane)
1
Aké rámce si praješ vypísať?
1. ARP
2. ICMP
3. HTTP
4. HTTPS
5. TELNET
6. SSH
7. FTP CONTROL
8. FTP DATA
9. TFTP
e žiadne
```

Po spracovaní všetkých rámcom je používateľovi poskytnutá možnosť nechať vypísať určitú komunikáciu alebo všetky rámce/packety určitého protokolu.

Po tomto spracovaní program zase vypíše prvé menu a ideme odznova.

Dôležitá vlastnosť programu je, že **výpisky** rámcov alebo komunikáciách vcelku sú **presmerované do externého súboru**, už spomínaného „*vystup.txt*“.

Výpisky, kt. sú určené pre používateľa a týkajú sa priadenia programu sa však vypisujú do Pycharm konzoly.

Implementačné prostredie

Pre implementovanie zadania je použitý programovací jazyk Python, ver. 3.9 s kt. som pracoval vo vývojovom prostredí PyCharm.

Program je funkčný a spustiteľný za predpokladu dodržania všetkých špecifikačných potrieb pre spustenie python kódu (nainštalovaný jazyk python, dostupnosť knižníc..) a samozrejme prítomnosť potrebných externých súborov.