SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

ZADANIE 1 – ANALYZÁTOR SIEŤOVEJ KOMUNIKÁCIE

Dávid Kromka
Cvičenie: Piatok 8:00
21.10.2021

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

1. Obsah

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

1. Zadanie úlohy

Zadaním úlohy je vytvorenie analyzátora sieťovej komunikácie Ethernet II siete, ktorý analyzuje záznamy z .pcap súborov.

1.1. Výpis rámcov

Prvým bodom zadanie je vypísať poradové číslo rámca, jeho dĺžku poskytnutú pcap API a reálnu dĺžku rámca prenášaného po médiu. Je potrebné určiť typ rámca, fyzickú cieľovú a zdrojovú MAC adresu a vypísať rámec v hexadecimálnom tvare.

1.2. Vnorený protokol

Pre rámce typu Ethernet II a IEEE 802.3 je potrebné vypísať vnorený protokol.

1.3. Analýza cez vrstvy

Pre rámce typu Ethernet II a protokoly z rodiny TCP/IPv4 sa na konci výpisu bodu 1 vypíše zoznam IP adries všetkých vysielajúcich uzlov a IP adresa uzla, ktorý odoslal najväčší počet paketov a počet týchto paketov.

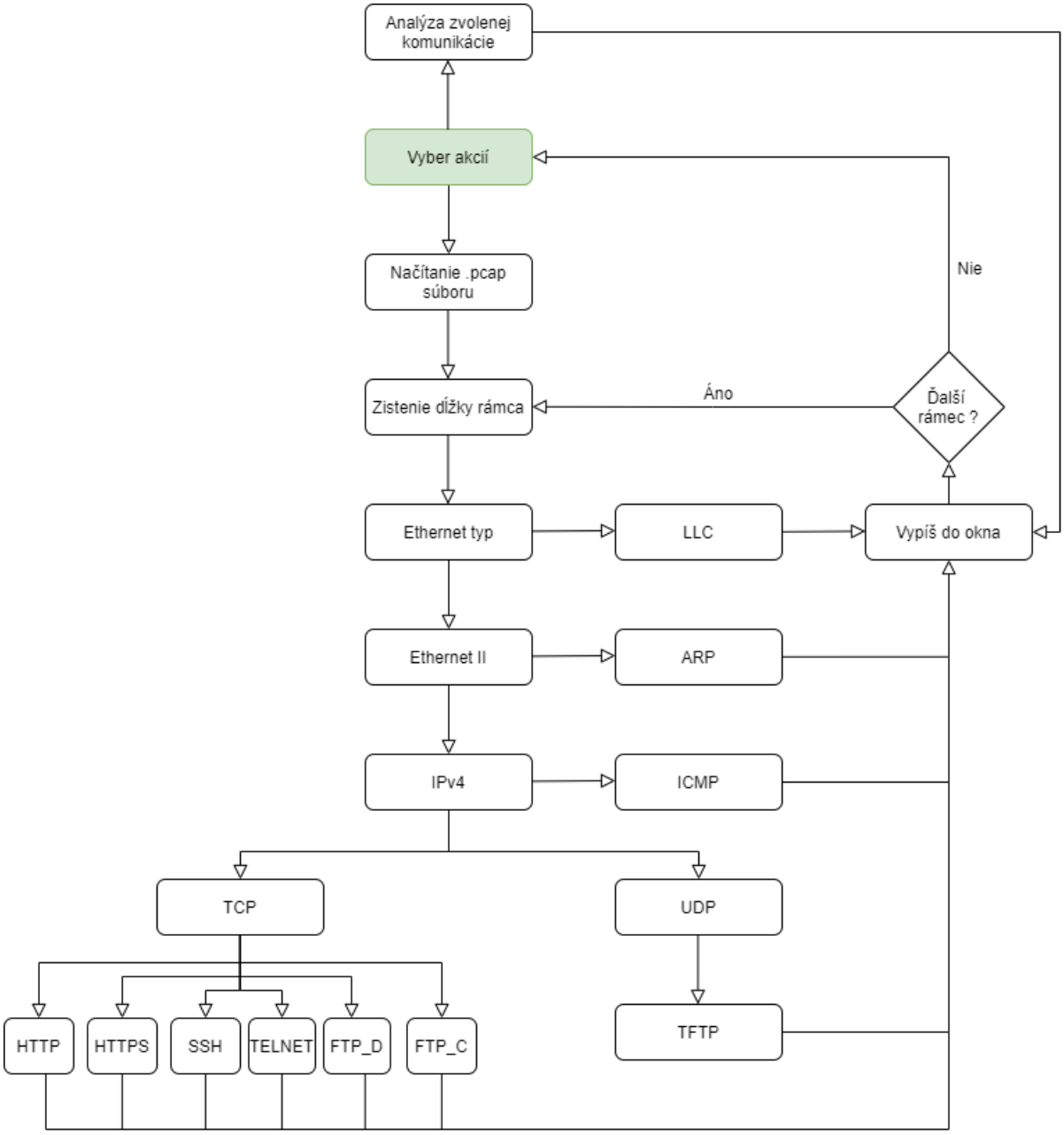
1.4. Analýza komunikácií

Úlohou je vypísať komunikácie pre protokoly HTTP, HTTPS, TELNET, SSH, FTP riadiace, FTP dátové, TFTP, ICMP a ARP. Pre ARP komunikáciu je potrebné vypísať všetky dvojice, ak pár neexistuje, vypísať tieto rámce osobitne. V prípade komunikácie so spojením sa vypíše jedna kompletná a jedna nekompletná komunikácia a ak komunikácia prekračuje dĺžku 20 rámcov, vypíše sa prvých a posledných 10.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

2. Blokový návrh fungovania riešenia

Diagram postupnosti analýzy rámcov, výpisu a analýzy komunikácii po zvolení akcii po spustení programu, začiatočným bodom je „Výber akcií“.



Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

3. Analýza protokolov na jednotlivých vrstvách

3.1. Sieť Ethernet – linková vrstva

Vo funkcii `get_type(result, frame)` sa kontorluje tretí rámec, hodnota 13. a 14. bytu. Ak je táto hodnota vyššia ako 1500 B, ide o Ethernet II a podľa tejto hodnoty sa hľadá vnorený protokol v externom textovom súbore. Ak je vnorený protokol typu IPv4 alebo ARP, pokračuje sa v analýze týchto protokolov vo funkcii `def ipv4(result, frame)` alebo `def arpc(result, frame)`. Ak je hodnota nižšia ako 1500 B, ide o dĺžku a sleduje sa 4. pole, 15. byte, ktorý predstavuje hodnotu DSAP. Prehľadáva sa externý súbor a ak ide o LLC + SNAP, hľadá sa vnorený protokol v poli ethertype, ak ide o LLC RAW, vnorený protokol je automaticky IPX alebo ide o LLC s inou hodnotou poľa DSAP.

```
def find(frame, n1, n2, sign):
    file.seek(0)
    f_hash = ''
    for line in file:
        if line.find('#') >= 0:
            f_hash = line[:-1]
        if line.find(frame[n1:n2].decode()) >= 0 and f_hash == sign:
            return ((line.split(':'))[1])[:-1]
    return 'Neznámy'
```

Kód pre vyhľadávanie protokolov a portov v externom súbore. Vstupom pre funkciu je rámec z ktorého sa hľadá, n1 a n2 sú začiatočný a koncový index z rámca, ktorý predstavuje text, ku ktorému hľadáme pomocou cyklu názov v súbore. Ak je hľadanie úspešné, je vrátený tento názov, inak je vrátený text „neznámy“. Sign predstavuje značku, v ktorej skupine v textovom súbore sa má názov hľadať.

3.2. Analýza cez vrstvy pre protokoly z rodiny TCP/IPv4

Pri protokole IPv4 sa sleduje zdrojová a cieľová IP adresa a zároveň sa do poľa ukladajú všetky zdrojové adresy a ich počet, kvôli ich výpisu na konci výpisu rámcov. Vnorený protokol sa vyhľadáva v externom súbore a ak ide o protokoly UDP, ICMP alebo TCP, pokračuje sa vo funkciách pre ich analýzu. Analýza protokolu IPv4 sa vykonáva vo funkcii `def ipv4(result, frame)`.

Analýza ICMP protokolu prebieha vo funkcii `icmp(result, frame)` a zisťuje sa Type z externého súboru.

Analýza protokolu ARP prebieha vo funkcii `def arpc(result, frame)`. Sleduje sa opcode, zdrojová a cieľová MAC a IP adresa.

```
def arpc(result, frame):
    op = frame[40:44]
    smac = frame[44:56]
    sip = frame[56:64]
    dmac = frame[64:76]
    dip = frame[76:84]
    result.extend([op, smac, sip, dmac, dip])
    result.append(frame)
    arp_list.append(result)
    gui.draw_arp(op, smac, sip, dmac, dip)
    gui.draw(result[:7], frame)
```

Funkcia na analýzu ARP protokolu. Vstupom je pole result, v ktorom sa nachádzajú výsledky analýzy na nižších vrstvách a do tohto poľa sa uložia aj výsledky analýzy na tejto vrstve.

Na TCP protokole sa sledujú flagy, zdrojová a cieľová MAC adresa, ktoré sa hľadajú v externom súbore. Ak je vnorený protokol http, HTTPS, SSH, TELNET, FTP-data alebo FTP-control, pokračuje sa v jednotlivých funkciách pre tieto protokoly. TCP protokol sa analyzuje vo funkcii `def tcp(result, frame)`.

Pri protokole UDP sa vo funkcii `def udp(result, frame)` sleduje cieľová a zdrojová MAC adresa, ktoré sú hľadané v externom súbore. Ak je vnorený protokol TFTP, pokračuje sa vo funkcii pre tento protokol.

Vo funkciách pre protokoly, pri ktorých sa analyzuje komunikácia sa jednotlivé rámce ukladajú do konkrétneho poľa pre daný protokol, aby pri analýze komunikácii nebolo potrebné prechádzať všetkými rámcami znova.

4. Analýza komunikácií

4.1. Analýza komunikácie rámcov pod TCP protokolom

Komunikácia medzi rámcami HTTP, HTTPS, TELNET, SSH, FTP riadiace, FTP dátové zdieľajú rovnaké funkcie na analýzu komunikácie `com_start(prot_list, index)` na nájdenie začiatku komunikácie 3-way handshake, `def get_com(start, prot_list)` na nájdenie rámcov patriacich do komunikácia a určenie konca komunikácie a či je komunikácia kompletná alebo nekompletná. Poslednou funkciou je `def communication(prot_list)`, ktorá slúži na volanie funkcie na hľadanie komunikácii so začiatkom na rozdielnych indexoch v poli rámcov, funkcia sa ukončí, keď je nájdená kompletná a nekompletná komunikácia alebo sú prehľadané všetky komunikácie.

4.2. TFTP a ICMP komunikáciaa

TFTP komunikácia je analyzovaná vo funkcii `tftp_com()` a funguje na princípe prehľadania poľa TFTP rámcov a rozdelenia týchto rámcov do komunikácie, do ktorej patria. Na podobnom princípe funguje analýza ICM komunikácie vo funkcii `icmp_com()`.

4.3. ARP komunikácia

Pri ARP komunikácii sa vypisujú dvojice, prípadne ak je viac requestov na jeden reply, sú vypísané všetky. Za komunikáciami sú vypísané zvyšné rámce, bez páru. Párovanie funguje na porovnávaní cieľových a zdrojových MAC a IP adries. Ako prvý sa nájde reply a k nemu sa následne nájde request.

5. Štruktúra externých súborov

V programe je použitý iba jeden externý textový súbor .txt, v ktorom sú uložené čísla v hexadecimálnom tvare vyjadrujúce protokoly alebo porty a ich názvy, oddelené dvojbodkou. Jednotlivé skupiny protokolov a portov sú oddelené znakom # a názvom protokolu na nižšej vrstve, pod ktorý patria.

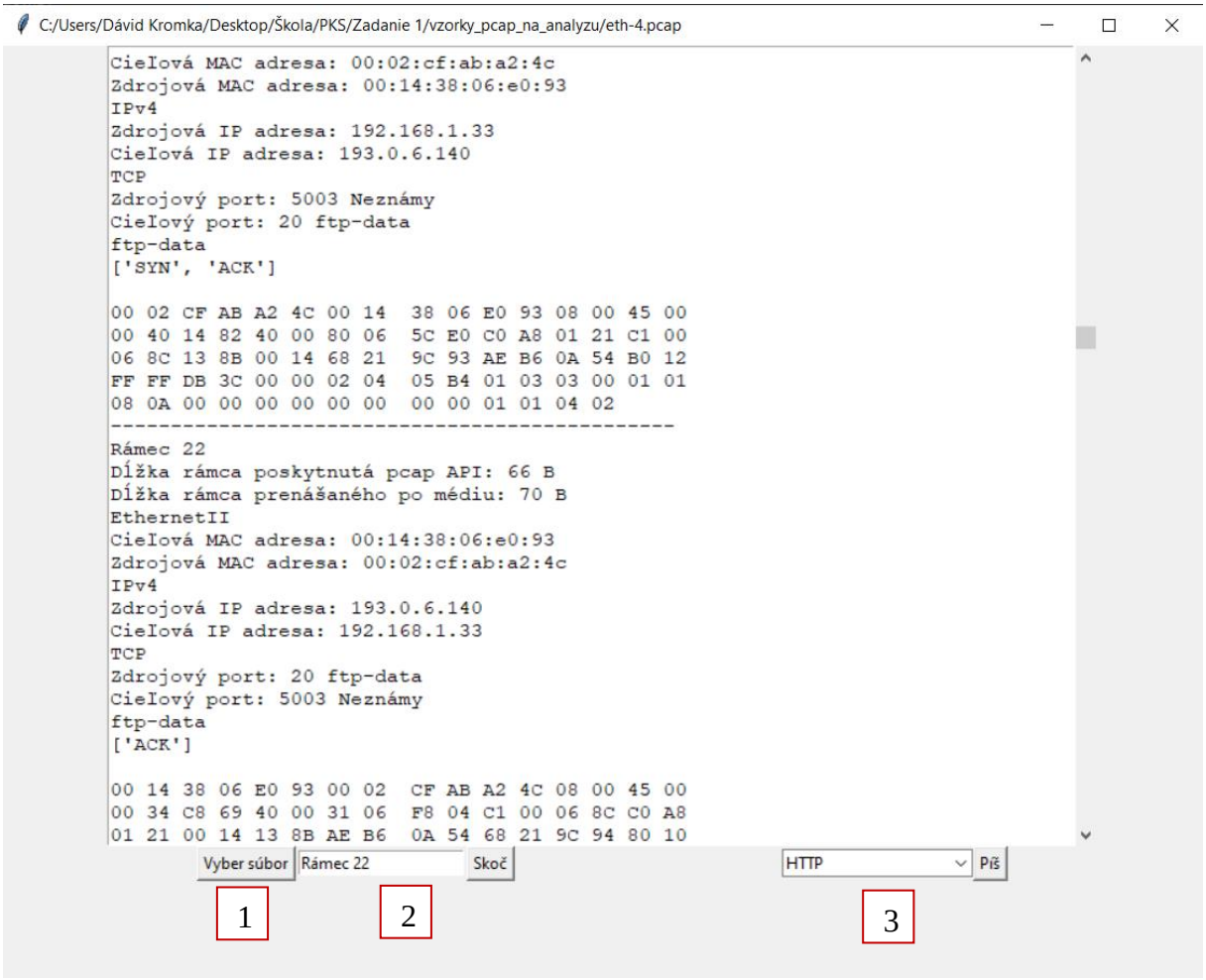
Úryvok zo súboru:

```
#ip
01:ICMP
02:IGMP
06:TCP
09:IGRP
11:UDP
2f:GRE
32:ESP
33:AH
39:SKIP
58:EIGRP
59:OSPF
73:L2TP
#udp
0007:echo
0013:chargen
0025:time
0035:domain
0043:bootps (DHCP)
0044:bootpc (DHCP)
0045:tftp
0089:netbios-ns
008a:netbios-dgm
00a1:snmp
```

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

6. Používateľské rozhranie

Používateľské rozhranie je realizované pomocou knižnice Tkinter a skladá sa z okna na výpis rámcov a komunikácií, tlačidla na výber .pcap súboru, vstupného poľa a tlačidla na skok na zadaný rámec a comboboxu a tlačidla na analýzu zvolenej komunikácie.



1. Prvým krokom pri spustení programu je výber .pcap súboru pomocou tlačidla „Vyber súbor“, následne sa vo vypisovacom okne zobrazia analyzované rámce.
2. Pomocou tlačidla „Skoč“ a vstupného poľa vľavo od tlačidla sa vieme posunúť vo výpise na žiadaný rámec po zadaní textu „Rámec “ nasledovaný číslom rámca.
3. Pomocou tlačidla „Píš“ a comboboxu vľavo, vieme vybrať typ komunikácie, ktorú chceme analyzovať a vypísať. Analýza komunikácie sa vypíše na konci výpisu analýzy rámcov.

7. Implementačné prostredie

Na realizáciu zadania je využitý jazyk Python 3.9.5 a k nemu zodpovedajúce implementačné prostredie Pycharm. Použité knižnice sú Scapy v2.4.5, ktorá je použitá na otváranie .pcap súborov a Tkinter, pomocou ktorej je vytvorené používateľské grafické rozhranie.