FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
SLOVENSKÁ TECHNICKÁ UNIVERZITA
Ilkovičova 2, 842 16 Bratislava 4

2020/2021

Počítačové a komunikačné siete

Zadanie č.1

Cvičiaci: Ing. Miroslav Bahleda, PhD.
Čas cvičení: Štvrtok 18:00 – 19:50

Vypracovala: Monika Zjavková
AIS ID: 105345

Obsah

1. Zadanie

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.

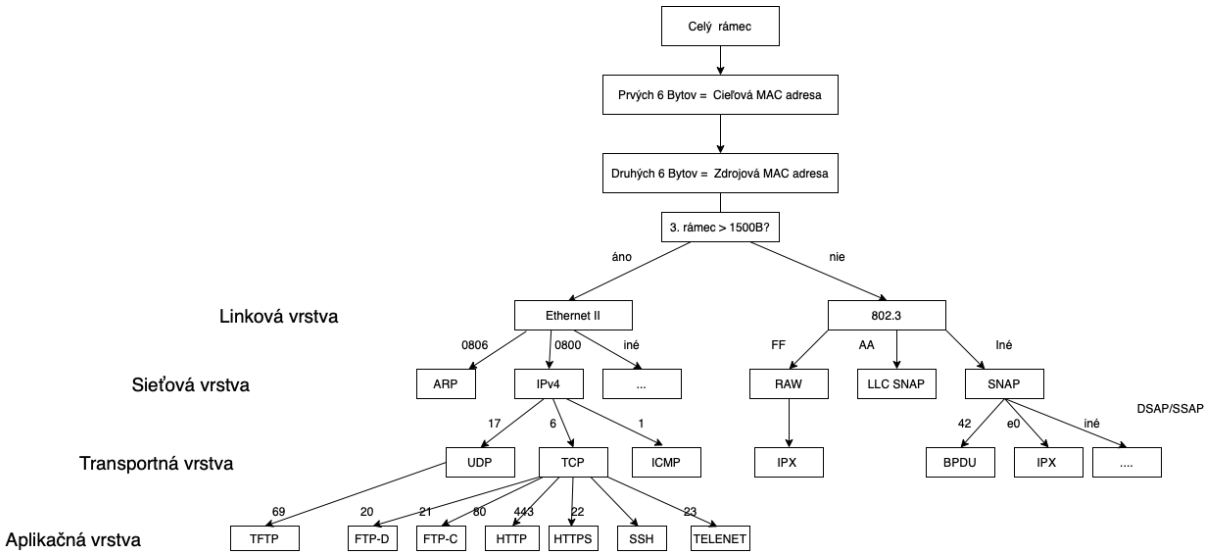
5) Program musí byť organizovaný tak, aby čísla protokolov v rámci Ethernet II (pole Ethertype), IEEE 802.3 (polia DSAP a SSAP), v IP pakete (pole Protocol), ako aj čísla portov v transportných protokoloch boli programom **načítané z jedného alebo viacerých externých textových súborov.** Pre známe protokoly a porty (minimálne protokoly v bodoch 1) a 4) budú uvedené aj ich názvy. Program bude schopný uviesť k rámcu názov vnoreného protokolu po doplnení názvu k číslu protokolu, resp. portu do externého súboru. Za externý súbor sa nepovažuje súbor knižnice, ktorá je vložená do programu.

6) V procese analýzy rámcov pri identifikovaní jednotlivých polí rámca ako aj polí hlavičiek vnorených protokolov nie je povolené použiť funkcie poskytované použitým programovacím jazykom alebo knižnicou. **Celý rámec je potrebné spracovať postupne po bajtoch.**

7) Program musí byť organizovaný tak, aby bolo možné jednoducho rozširovať jeho funkčnosť výpisu rámcov pri doimplementovaní jednoduchej funkčnosti na cvičení.

8) Študent musí byť schopný preložiť a spustiť program v miestnosti, v ktorej má cvičenia. V prípade dištančnej výučby musí byť študent schopný prezentovať podľa pokynov cvičiaceho program online, napr. cez Webex, Meet, etc.

2. Blokový návrh riešenia



3. Navrhnutý algoritmus analyzovania

Algoritmus analyzovania rámcov funguje po vrstvách. Vstup sa číta po rámoch a následne sa spracováva najskôr vo funkcii main, kde získa zdrojové a cieľové MAC adresy a určí protokol pre linkovú vrstvu podľa veľkosti 4 bajtov za MAC adresami.

V prípade 802.3 sa určí, s akým protokolom sa pracuje podľa podmienok na sieťovej vrstve a následne z načítaných dát z externého súboru sa určí vnorený protokol na transportnej vrstve.

Analyzovanie prebieha podobne pre Ethernet II, kde sa vnorené protokoly určujú len pre IPv4. V tom prípade sa určujú protokoly aj na Aplikačnej vrstve podľa čísiel portov uložených v externom súbore.

Následne je celý výstup zapísaný vo výstupnom súbore

Pri spracovávaní údajov pre základné údaje, sa ukladajú zoznamy všetkých IP adries odosielajúcich uzlov. Tento zoznam sa prechádza postupne a jedinečné výskyty IP adries sa vkladajú do slovníka s počtom výskytu. Ak sa nájde duplikát zvýši sa počet, na konci je teda možné nájsť maximum, čiže IP adresa s najviac odoslanými paketmi.

V bode 4 je možné riešiť analýzu TCP komunikácie rovnakým spôsobom, preto je to vykonávané v jednej funkcii. Pri prechádzaní rámcov s hľadaným protokolom sa funkcia pozerá na parameter flags a hľadá začiatok komunikácie – SYN, následne skontroluje, či začiatok komunikácie prebehol úspešne a v ďalšom cykle sa hľadá koniec, pri čom sa pridáva komunikácia prebehnutá na určených portoch. Keď nájde FIN alebo RST, skontroluje koniec komunikácie a zaradí sa medzi úplne. V prípade, že nebola ukončená zapíše sa ako neúplná. Program skončí, keď nájde úplnú a neúplnú komunikáciu.

Analýza TFTP komunikácie prebieha po vyfiltrovaní protokolu UDP, kde nájde najbližší port 69 pre TFTP. Potom prechádza ďalšie komunikácie, či sa tam vyskytuje port z predchádzajúceho rámca. Ak sa nájde ďalší port 69 alebo s predchádzajúcim portom sa nezhoduje, komunikácia je pokladaná za ukončenú.

Pri ICMP komunikácii sú popárované rámce pri Echo a Echo Reply pomocou IP adries. Každý výpis obsahuje aj správu podľa kódu v políčku type.

Funkcia na vypísanie ARP dvojíc začína najskôr nájdením ARP protokolu typu Request a následne sa k nemu hľadá v cykle Reply, ak sa nájde Request s rovnakými požiadavkami ako prvý, pridá sa do komunikácie. Komunikácia sa po ukončení vypíše a vymaže zo zoznamu ARP rámcov. Všetky komunikácie, ktoré tam ostanú a nemajú pár, sú vypísané na konci.

4. Príklad štruktúry externých súborov

4.1. LLC

```
00 Null SAP
02 LLC Sublayer Management / Individual
03 LC Sublayer Management / Group
06 IP (DoD Internet Protocol)
0e PROWAY
0f NetBIOS
42 BPDU
4e MMS
5e ISI IP
7e X_25 PLP
8e PROWAY
aa SNAP
e0 IPX
f4 LAN Management
fe ISO Network Layer Protocols
```

4.2. Ethertype

```
0200 XEROX PUP
P201 PUPP Addr Trrans
0800 IPv4
0801 X.75 Internet
0805 X.25 Level 3
0806 ARP
8035 Reverse ARP
809B Appletalk
80F3 AppleTalk AARP
8100 IEE 802.1Q VLAN-tagged frames
8137 Novell IPX
86DD IPv6
880B PPP
8847 MPLS
8848 MLPS with upstream-assinged label
8863 PPPoE Discovery Stage
8864 PPPoE Session Stage
```

4.3. TCP

```
7 echo
19 chargen
20 FTP dátove
21 FTP riadiace
22 SSH
23 TELENET
25 smtp
53 domain
79 fingerr
80 HTTP
110 pop3
111 sunrpp
119 nntp
139 netbios-ssn
143 imapp
179 bgp
389 ldap
443 HTTPS (ssl)
445 microsoft-ds
1080 socks
```

4.4. IPv4

```
1 ICMP
2 IGMP
6 TCP
9 IGRP
17 UDP
47 GRE
50 ESP
51 AH
57 SKIP
88 EIGRP
89 OSPF
115 L2TP
```

5. Používateľské rozhranie a implementačné prostredie

Prostredie použité na implementáciu je PyCharm. Program je spúšťaný cez main funkciu nachádzajúcu sa na konci súboru. Súbory na analýzu komunikácie sa nachádzajú v súbore s názvom Vzorky, odtiaľto sa otvára aktuálne napísaný v maine.

```
def main():
    n = 1

    # Súbor s pcap súbormi
    pcap = scapy.rdpcap("vzorky/eth-2.pcap")
```

Po spustení sa vypíšu všetky rámce s očíslovaním podľa požiadaviek pre body 1.-2. do externého súboru s názvom „výstup.txt“. Následne vypýta od používateľa z konzoly vstup pre vypísanie ďalších úloh. Jednotlivé čísla, podľa ktorých je spravená ďalšia funkcionalita programu sú vždy vypísané v konzole.

Program pýta vstup, pokiaľ nie je ukončený enterom.