VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

počítačové komunikace a sítě 2021/2022

2. projekt do predmetu IPK Varianta ZETA: Sniffer paketov

Pavel Kratochvíl - xkrato61

18.4.2022

Obsah

1 Úvod

Tento dokument popisuje princíp implementácie, fungovania nástroja v rámci predmetu IPK pre odchytávanie a analýzu paketov.

2 Úvod do problematiky

2.1 Čo je paket?

Paket (angl. packet - balíček) je základná jednotka dát prenášaná počas komunikácie v TCP/IP sieti. Paket má zo špecifikácie stanovenú štruktúru a obmedzenú dĺžku. Delením správ komunikácie do množstva paketov vieme docieliť konzistentnú a spoľahlivú komunikáciu medzi dvomi zariadeniami.

Na prenos paketu nie je potrebná žiadna dopredu zahájená komunikácia medzi dvomi zariadeniami. Vďaka tomu môžme vyslať do siete paket s určeným cieľom a ak existuje cesta k príjemcovi a komunikácia prebehla v poriadku, paket dorazí do cieľa bez ďalších potrebných akcií z našej strany.[1]

2.2 Štruktúra paketu

Paket je reťazec bitov rozdelených do dvoch hlavných častí:

- Riadiace dáta(metadáta)
- Užívateľské dáta(angl. payload)

Riadiace metadáta umožňujú správny pohyb paketu medzi zariadeniami. Takýto pohyb ale môže byť značne komplexný a zahŕňajúci množstvo rôznych zariadení. Preto bol vyvynutý abstraktný na vrstvách založený opis návrhu štruktúry komunikačných a počítačových sieťových protokolov s názvom OSI[2] (Open Systems Interconnection Reference Model). Tento model rozčleňuje pohyb dát do siedmych vrstviev, z ktorých každá vrstva poskytuje služby vrstve nad ňou a využíva vrstvy pod ňou.[3]

Aby každá vrstva mala potrebné informácie pre operácie s paketom s ňou spojené, pri odosielaní paketu každá vrstva zaobalí paket. Zaobalením paketu rozumieme pripojenie ďalších metadát k jeho užívateľskému obsahu.

2.3 Zaobalenie paketu(packet encapsulation)

Prvá vrstva(najvrchnejšia) obalu, ktorou sa `ipk-sniffer` zaoberá je 2. spojová vrstva OSI. Môj program umožňuje ďalše rozbalovanie iba `Ethernet` paketov. Pod ňou nájdeme 3. sieťovú vrstvu. V nej môžme ďalej analyzovať `IP`, `ICMP` a `ARP` rámce. Pod nimi nasleduje 4. transportná vrstva, ktorá je obmedzená na `TCP` a `UDP` pakety.

3 Implementácia

Pri implementácii projektu som používal mnoho štandardných knižníc a tiež aplikačné programové rozhranie pcap, ktoré mi v mnohom uľahčilo prácu. Keďže sa ale o danú oblasť zaujímam, rozhodol som sa nepoužiť niektoré vstavané funkcie zamerané na filtrovanie odchytených paketov(napr. `pcap-filter`), čo sa odrazilo aj na rozsiahlosti môjho riešenia.

Kvôli nepoužitiu vstavaného filtra a k nemu patriacich funkcií pre iterovanie v získaných paketoch, som zvolil nekonečnú slučku odchytávania paketov `pcap_loop()`, ktorú zastaví buď dosiahnutý počet odchytených paketov špecifikovaný užívateľom(-n X), chyba počas behu programu(napr. chyba alokácie pamäte) alebo zastavenie za behu užívateľom(napr. ctrl+c).

Po nájdení nového paketu nekonečnou slučkou sa spúšťa funkcia `got_packet()`, ktorá podľa hlavičiek a obsahu rozhodne, ktorá ďalšia funkcia bude otvárať následujúcu vrstvu zaobalenia.

Na ukladanie informácií získaných o konkrétnom pakete využívam štruktúru v globálnej premennej `packet_storage`. Na uľahčenie práce s reťazcami využívam vlastnú implementáciu dynamického reťazca v súbore `dynamic_string.c`.

Pri implementácii som sa opieral hlavne o dokumentáciu, zdrojové kódy jednotlivých knižníc a referenčný návod ku knižnici `pcap`[4].

4 Rozšírenia

V mojom riešení som implementoval aj jedno malé rozšírenie, ktoré užívateľovi umožňuje filtrovanie paketov podľa verzie IP protokolu. Použitie filtra je možné vstupným argumentom `--ipv4` alebo `--ipv6`. Vďaka manuálnej implementácii triedenia paketov v mojom riešení by bola implementácia ďalších filtrov podľa akejkoľvek zachytenej informácie triviálne.

5 Výstup programu

5.1 Štruktúra výstupu

Výstupom môjho programu je n blokov, z ktorých každý odpovedá jednému zachytenému paketu. Blok môže obsahovať následujúce položky:

```
timestamp:
eth type:
protocol:
arp protocol:
arp operation:
src MAC:
dst MAC
src MAC (IPv6):
dst MAC (IPv6):
frame length:
src IP:
dst IP:
src port:
dst port:
ICMP type:

0x0000 28 d0 ea 5c 3d 25 d8 47  32 f5 d9 c5 08 00 45 00  (..\=%.G  2.....E.
```

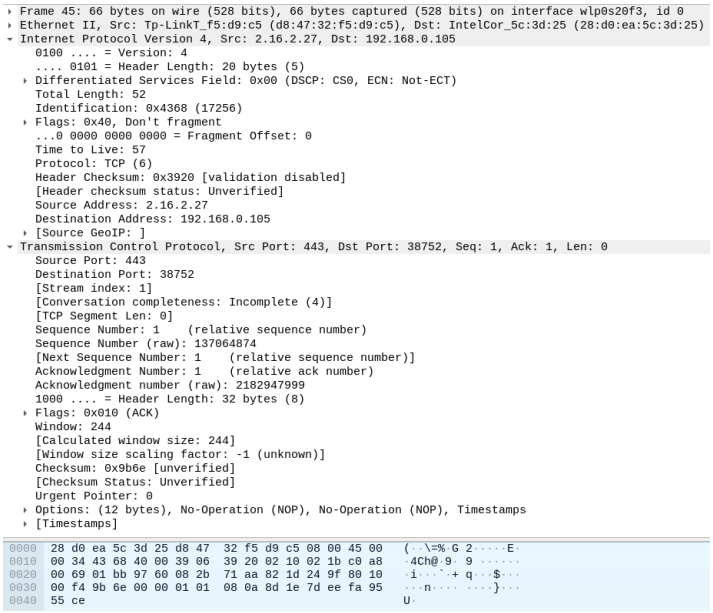
Ktoré položky sa zobrazia záleží na type nájdeného paketu a jeho hlavičkách(protokolu a typu obsahu). Napríklad pred IPv4 sa nam zobrazia iba polia `timestamp`, `eth type`, `protocol`, `src a dst MAC`, `frame length`, `src a dst IP`, `src a dost port` a `payload size`. Obsah celého rámca sa zobrazí v každom prípade.

Formát výpisu obsahu jedného riadka rámca je: offset(posun v rámci v hexadecimálnom formáte), 16 byte-ov dát, z ktorých každý je reprezentovaný dvojicou znakov v hexadecimálnej sústave a na konci je reprezentácia tých istých 16 byte-ov v ASCII znakoch. Netlačiteľné znaky sú nahradené bodkou.

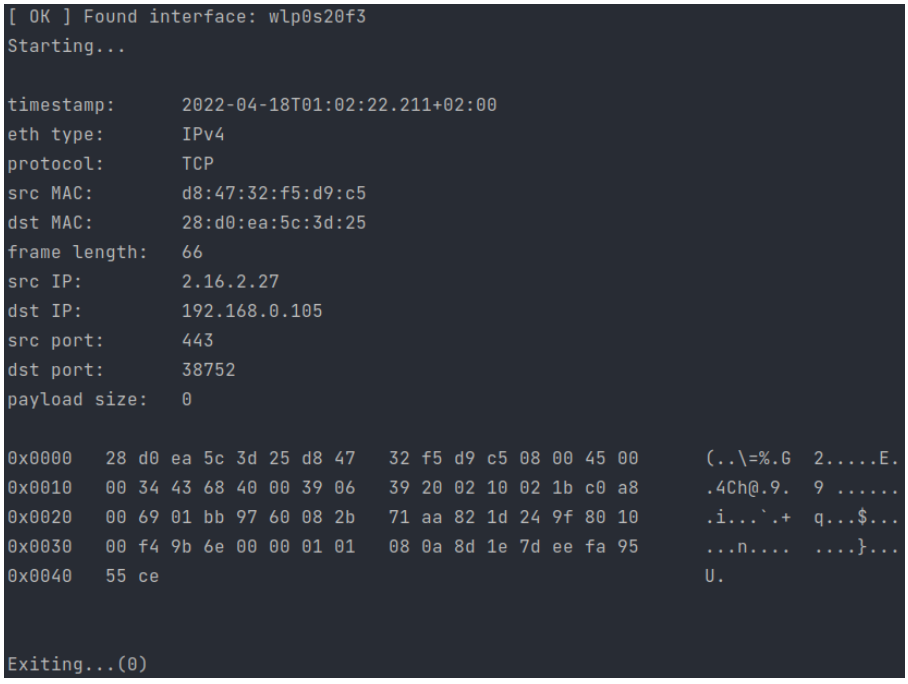
6 Testovanie

6.1 Porovnanie s referenčným nástrojom Wireshark

Pri testovanie som si vybral ako referenčný nástroj Wireshark[5](verzia 3.6.2). Porovnanie výstupov pre jeden zachytený paket(IPv4 TCP):



Obrázek 1: Referenčný výstup z nástroja Wireshark



Obrázek 2: Výstup z ipk-sniffer

Literatúra

[1] Kurose, J. F.; Ross, K. W.: *Computer Networking: A Top-Down Approach*. Boston, MA: Pearson, 7 vydání, 2016, ISBN 978-0-13-359414-0.

[2] Open Systems Interconnection – Connection-oriented Session protocol: Protocol specification. Nov 1995.
URL `https://www.itu.int/rec/T-REC-X.225-199511-I/en`

[3] Data Encapsulation and the TCP/IP Protocol Stack. 2010.
URL `https://docs.oracle.com/cd/E19455-01/806-0916/ipov-32/index.html`

[4] Cartens, T.: Programming with PCAP. 2002.
URL `https://www.tcpdump.org/pcap.html`

[5] Wireshark. [online], 2022.
URL `https://www.wireshark.org/`