
# IPK
# Projekt 2
# Varianta Zeta: Sniffer packetov

Rebeka Černianska, xcerni13

# Obsah

## Úvod

Cieľom tohto projektu bolo implementovať sieťový analyzátor, ktorý zachytáva packety na určitom sieťovom rozhraní. Program by mal zachytávať UDP, TCP a ICMP packety, prípadne ARP rámce. Toto riešenie nespĺňa celé zadanie, pričom program spracuje argumenty, no nevypíše obsah packetov, iba TCP hlavičku.

## Implementácia

Implementácia programu bola zvolená v jazyku C. Využié knižnice boli zo zadania, pcap, ďalej netinet, a potom iné podporné knižnice ktoré netvoria základ riešenia, ale umožňujú prácu s rôznymi dátovými typmi a časom.

### Spracovanie argumetov

Prvá činnosť, ktorú program vykonáva je načítanie a spracovanie argumentov. Toto sa vykonáva pomocou funkcie getopt_long(), ktorá umožňuje prácu s dlhými verziami argumentov. Kontrolujú sa požadované parametre pri argumentoch, prípadne chybné zadané argumenty, alebo ich parametre ukončujú beh programu. Táto časť je dôležitá pre ďalší beh programu vzhľadom na to, že argumenty určujú aké packety sa budú filtrovať.

### Nastavenie filtrov

Napriek tomu že spracovanie týchto packetov implementované nie je, spracovanie týchto filtrov a ich nastavenie sa v programe vykonáva. Podľa zadaných argumentov program rozhodne, ako sa nastaví filter. Pokiaľ nie je nastavené zariadenie, na ktorom sa budú hľadať packety, vypíšu sa všetky dostupné zariadenia pomocou funkcie pcap_findalldevs(). Okrem jednotlivých typov packetov, nastavuje prípadný port, a je spustená tzv. Sniffing session pomocou funkcie pcap_open_live(). Ďalšie krokz sú kompilácia, nastavenie filtra.

### Načítavanie packetov

Hlavná slučka programu je tvorená funkciou pcap_loop(), ktorá ako parameter dostane počet packetov ktoré ma načítať, ako bolo zadané v parametroch. Pre každý packet je zavolaná funkcia my_callback(), ktorá sprostredkuje spracovanie jednotlivých packetov.

### Spracovanie packetov

Vo funkcii my_callback() sa ako prvé nájde a vypíše aktuálny čas pomocou knižnice time.h. Následne sa pomocou knižnice netdb.h vypíše zdrojová a cieľová IP adresa a čislo portu.

## Testovanie

Program bol testovaný pomocou programu Wireshark, kde výstupy programu ipk-sniffer a Wireshark boli porovnané. Toto testovanie pomáhalo dolaďovať implementačné detaily a pomohlo odhaliť nejaké nedostatky.

**Zdroje**


- [1] CARSTENS, Tim. Programming with pcap, 2002, [cit. 25.5. 2021]

  URL: https://www.tcpdump.org/pcap.html

- [2] How to format time, 2018, [cit. 25.5. 2021]

  URL: https://stackoverflow.com/questions/48771851/im-trying-to-build-an-rfc3339-timestamp-in-c-how-do-i-get-the-timezone-offset