IPK - 2. projekt

Packet Sniffer

9.5.2021 Martin Žovinec

Obsah

Zadání

Cílem projektu bylo navrhnout a implementovat síťový analyzátor v C/C++/C#, který bude schopný na určitém síťovém rozhraním zachytávat a filtrovat pakety

Implementace

Jazyk

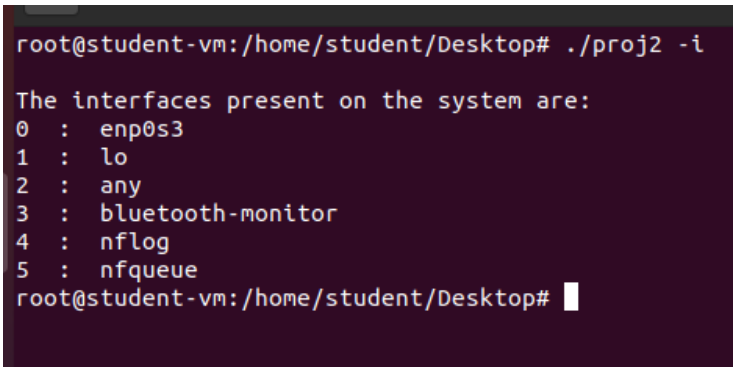
Projekt byl implementován v jazyce C.

Vstupní argumenty

Vstupní argumenty jsou zpracovány s pomocí funkce `getopt_long()`. U jednoduchých argumentů `-t/--tcp`, `-u/--udp`, `--icmp a--arp`, kterými uživatel volí typy paketů, které chce analyzovat, se pouze nastaví proměnná typu bool na true. Při argumentech -i INTERFACE, -p PORT a -n NUMBER je následující argument uložen a v případě -n NUMBER jsou navíc provedeny kontroly, zda je NUMBER kladné číslo.

Rozhraní

Následně se pomocí funkce `pcap_open_live()` otevře rozhraní a nebo v případě neuvedení žádného rozhraní uživatelem jsou vypsány všechny aktivní rozhraní pomocí funkce `print_devices()` a program se ukončí.



Výpis aktivních rozhraní

Tvorba filtru

Program pokračuje tvorbou filtračního řetězce, který bude později předán funkci `pcap_compile()`. Řetězec je postupně tvořen pomocí podmínek s proměnnými typu bool, které byly nastaveny při kontrole argumentů. V případě, že byl zadán i argument -p PORT a alespoň jeden z argumentů -t/--tcp nebo -u/--udp, tak se na konec řetězce přidá i port. Výsledný řetězec může vypadat například takto : `icmp or arp or (tcp and port 30)`. Potom se pomocí funkcí `pcap_lookupnet()`, `pcap_compile()` a `pcap_setfilter()` vytvoří samotný filtr.

Filtrování paketů

Po vytvoření filtru se program zacyklí ve funkci `pcap_loop()`, která filtruje příchozí pakety a ty, které projdou filtrem předává funkci `process_packet()`, co podle hodnoty v hlavičce paketu rozhodne, o jaký protokol se jedná a podle protokolu zavolá nad paketem některou z funkcí `print_tcp_packet()`, `print_udp_packet()`, `print_icmp_packet()`, nebo `print_arp_packet()`.

Zpracování paketu

Jednotlivé funkce na tisknutí paketů jsou si dost podobné, zejména `print_tcp_packet()` a `print_udp_packet()`, které sice pracují pouze s jinou hlavičkou, ale tisknou stejné informace, kterými jsou čas, IP a port obou stran a data, kde IP se získává funkcí `inet_ntoa()` a port zase funkcí `ntohs()`.
V případě zbylých dvou funkcí `print_icmp_packet() a print_arp_packet()` se netiskne port a v případě `print_arp_packet()` se tiskne MAC adresa místo IP.
Všechny funkce využívají funkce `print_time()`, která vypíše datum a čas příchodu nebo odchodu paketu a funkce `PrintData()`, která vypisuje obsah IP hlavičky, pak obsah hlavičky protokolu a následně samotná data paketu a to ve formátu:
“offset vypsaných bajtů:” “výpis bajtů hexa” “výpis bajtů ASCII”

Ukončení programu

Kdy má program skončit je určeno uživatelem zadaným argument -n NUMBER, který značí počet paketů, co se mají analyzovat. Při zadání hodnoty 0 se pakety zpracovávají dokud uživatel projekt ručně nezastaví.
V případě, že argument není zadán se zpracuje pouze jeden paket.

Testování

Projekt byl testován manuálně a zejména formou kontroly splnění požadavků zadání. Také jsem trochu testoval porovnáváním paketů s Wiresharkem. Všechno testování probíhalo na referenčním virtuálním OS.

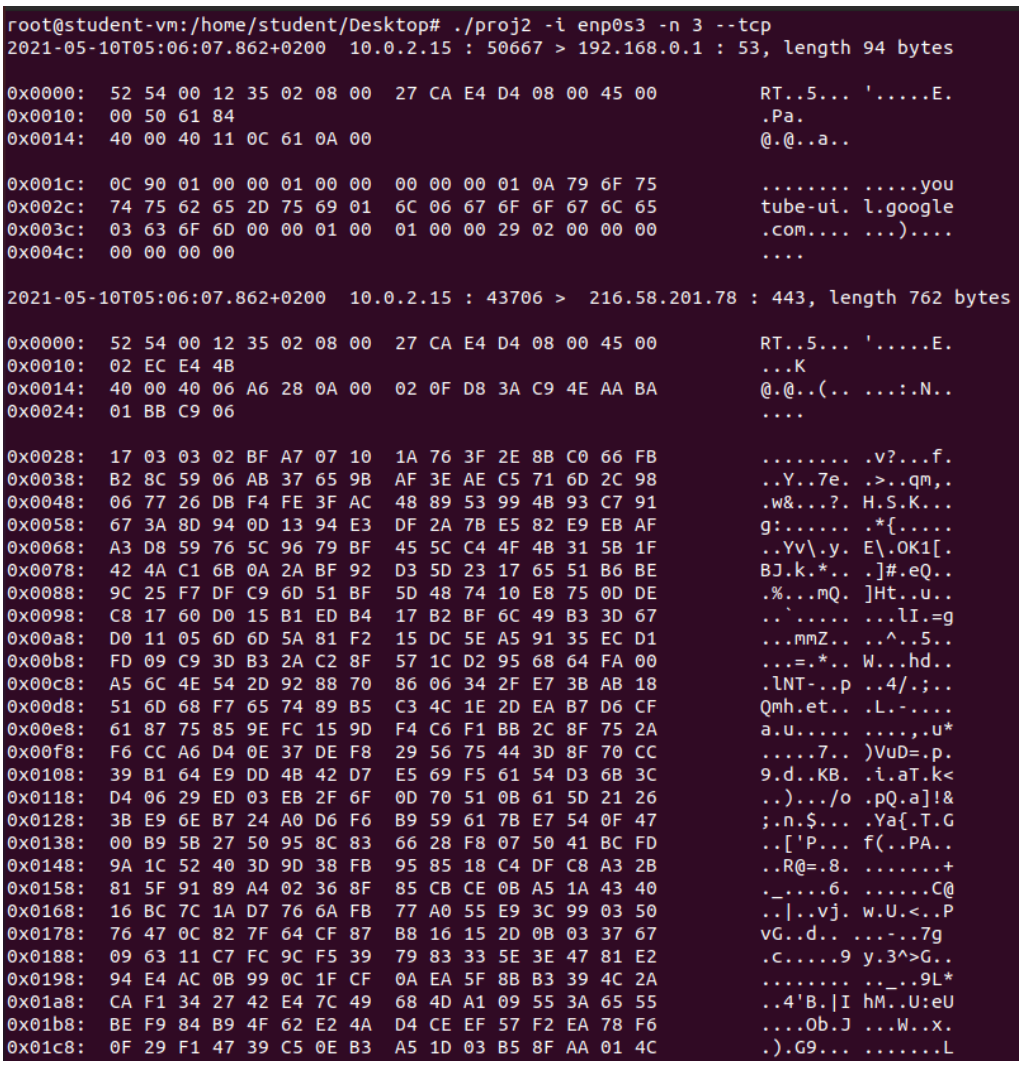
Ukázky manuálních testů

```
root@student-vm:/home/student/Desktop# ./proj2 -i enp0s3 -n 3 --udp
2021-05-10T05:04:35.093+0200  10.0.2.15 : 35035 > 192.168.0.1 : 53, length 88 bytes

0x0000:  52 54 00 12 35 02 08 00  27 CA E4 D4 08 00 45 00        RT..5... '.....E.
0x0010:  00 4A 2A 21                                            .J*!
0x0014:  40 00 40 11 43 CA 0A 00                                @.@.C...

0x001c:  2A FC 01 00 00 01 00 00  00 00 00 01 0D 73 74 61        *....... .....sta
0x002c:  63 6B 6F 76 65 72 66 6C  6F 77 03 63 6F 6D 00 00        ckoverfl ow.com..
0x003c:  1C 00 01 00 00 29 02 00  00 00 00 00 00 00              .....)..... ..

2021-05-10T05:04:35.094+0200  192.168.0.1 : 53 > 10.0.2.15 : 35035, length 88 bytes

0x0000:  08 00 27 CA E4 D4 52 54  00 12 35 02 08 00 45 00        ..'...RT ..5...E.
0x0010:  00 4A 1E C6                                            .J..
0x0014:  00 00 40 11 8F 25 C0 A8                                ..@..%..

0x001c:  2A FC 81 80 00 01 00 00  00 00 00 01 0D 73 74 61        *..•.... .....sta
0x002c:  63 6B 6F 76 65 72 66 6C  6F 77 03 63 6F 6D 00 00        ckoverfl ow.com..
0x003c:  1C 00 01 00 00 29 10 00  00 00 00 00 00 00              .....)..... ..

2021-05-10T05:04:36.121+0200  10.0.2.15 : 54220 > 192.168.0.1 : 53, length 86 bytes

0x0000:  52 54 00 12 35 02 08 00  27 CA E4 D4 08 00 45 00        RT..5... '.....E.
0x0010:  00 48 2A 3C                                            .H*<
0x0014:  40 00 40 11 43 B1 0A 00                                @.@.C...

0x001c:  1B 42 01 00 00 01 00 00  00 00 00 01 03 63 64 6E        .B...... .....cdn
0x002c:  07 73 73 74 61 74 69 63  03 6E 65 74 00 00 1C 00        .sstatic .net....
0x003c:  01 00 00 29 02 00 00 00  00 00 00 00                    ...)..... ....
```

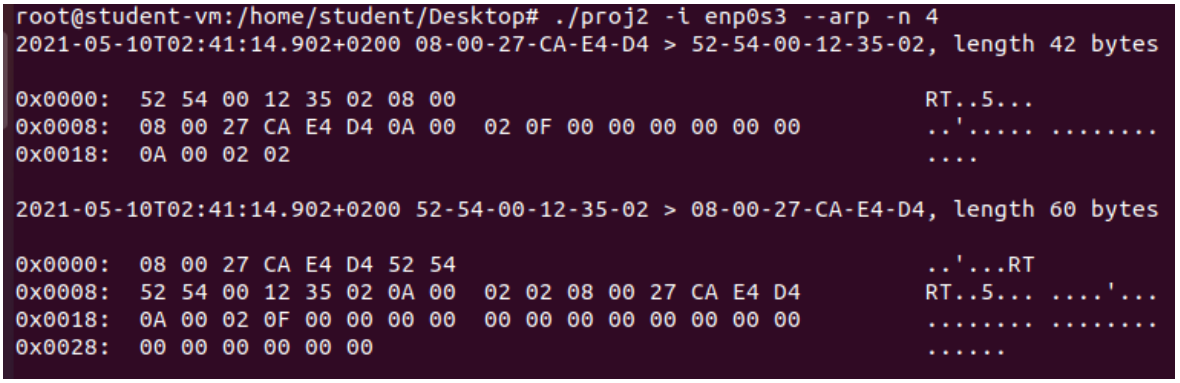
UDP pakety z obnovením webové stránky

```
root@student-vm:/home/student/Desktop# ./proj2 -i enp0s3 --icmp -n 4
2021-05-10T03:43:21.686+0200  10.0.2.15 > 216.58.201.78, length 98 bytes

0x0000:  52 54 00 12 35 02 08 00  27 CA E4 D4 08 00 45 00        RT..5... '.....E.
0x0010:  00 54 39 3E                                            .T9>
0x0014:  40 00 40 01 53 D3 0A 00                                @.@.S...
0x001c:  B9 8F 98 60 00 00 00 00  42 FB 04 00 00 00 00 00        ...`.... B.......
0x002c:  10 11 12 13 14 15 16 17  18 19 1A 1B 1C 1D 1E 1F        ........ ........
0x003c:  20 21 22 23 24 25 26 27  28 29 2A 2B 2C 2D 2E 2F         !"#$%&' ()*+,-./
0x004c:  30 31 32 33 34 35 36 37                                01234567

2021-05-10T03:43:21.686+0200  216.58.201.78 > 10.0.2.15, length 98 bytes

0x0000:  08 00 27 CA E4 D4 52 54  00 12 35 02 08 00 45 00        ..'...RT ..5...E.
0x0010:  00 54 19 84                                            .T..
0x0014:  00 00 73 01 80 8D D8 3A                                ..s.•..:
0x001c:  B9 8F 98 60 00 00 00 00  42 FB 04 00 00 00 00 00        ...`.... B.......
0x002c:  10 11 12 13 14 15 16 17  18 19 1A 1B 1C 1D 1E 1F        ........ ........
0x003c:  20 21 22 23 24 25 26 27  28 29 2A 2B 2C 2D 2E 2F         !"#$%&' ()*+,-./
0x004c:  30 31 32 33 34 35 36 37                                01234567
```

ICMP pakety z pingování google.com



TCP pakety ze sledování YouTube



ARP pakety

Použité zdroje

Zpracování argumentů:

- https://linux.die.net/man/3/getopt_long
- https://stackoverflow.com/questions/19604413/getopt-optional-arguments

Čas:

- https://stackoverflow.com/questions/13804095/get-the-time-zone-gmt-offset-in-c
- https://stackoverflow.com/questions/3673226/how-to-print-time-in-format-2009-08-10-181754-811

Packet sniffer:

- https://www.binarytides.com/packet-sniffer-code-c-libpcap-linux-sockets/
- https://www.tcpdump.org/pcap.html