

Vysoké učení technické v Brně
Fakulta informačních technologií

IPK – Počítačové sítě a komunikácia

Varianta ZETA: Sniffer paketov

Dokumentácia k projektu



Brno, 25. apríla 2021

Slavomír Svorada (xsvora02)

Obsah

1.	Úvod	3
2.	Implementácia	3
1.	Stručné údaje	3
2.	Funkcia main()	3
3.	Funkcia pcap_open()	3
4.	Funkcia run_cup()	3
5.	Funkcia packet_check()	4
6.	Funkcia print_all()	4
3.	Príklady výstupu programu	4
4.	Testovanie programu	6
5.	Zdroje	7
6.	Záver	8

1. Úvod

Cieľom projektu bude návrh a implementácia sieťového analyzátoru. Analyzátor bude schopný na určitom sieťovom rozhraní zachytávať a filtrovať pakety. Projekt bude robený tak, že podporuje zachytávanie paketov TCP, UDP, ICMP a ARP. Pakety sa vypíšu na *stdout* s hlavičkou. Na začiatku je vytvorený riadok, ktorý obsahuje čas, IP adresy, porty a dĺžku v bytoch.

2. Implementácia

1. Stručné údaje

Projekt som robil v jazyku C takže som použil knižnicu *Libpcap*. Hlavný kód sa nachádza v súbore *ipk-sniffer.cpp*. Nachádza sa u 1 globálna premenná ktorá slúži pre knižnicu *pcap.h*. Program sa skladá zo 6 funkcií a to: *main()*, *pcap_open()*, *run_cup()*, *find_interfaces()*, *packet_check()* a *print_all()*.

2. Funkcia *main()*

Vo funkcií najskôr spracujem dlhé argumenty pomocou *longopts[]* [1]. Ďalej vo funkcií kontrolujem argumenty pomocou *getopt_long()* + *switch* [2]. V prípade, že máme zadaný 1 argument alebo argument -i bez hodnoty sa vypíše zoznam aktívnych rozhraní. Ináč si skontrolujem o aký typ sa jedná a vytvorenú premennú si prepíšem na formát vhodný pre filter.

3. Funkcia *pcap_open()*

Funkcia *pcap_open()* je volaná ako prvá z funkcie *main()*. Táto funkcia slúži pre získanie soket deskriptoru pre sniffer pakety. Vo funkcií používam ďalšie funkcie z knižnice *pcap.h*. Na úvod si zistím, či funkcia má rozhranie definované užívateľom. V prípade že nemá, volám funkciu *pcap_lookupdev()* vďaka ktorej získam prvé dostupné rozhranie. Pre získanie deskriptora z vybraného rozhrania volám funkciu *pcap_open_live()*. Pre získanie masky siete používam funkciu *pcap_lookupnet()*. Funkcia *pcap_compile()* prevedie reťazec filtrov na kód. A vďaka funkcií *pcap_setfilter()* môže byť aplikovaný [3].

4. Funkcia *run_cup()*

Funkcia *run_cup()* je volaná taktiež z funkcie *main()*. Z tejto funkcie volám funkciu *pcap_loop()* [4]. Pomocou tejto funkcie dôjde k „sniffovaniu“ paketov.

5. Funkcia *packet_check()*

Na začiatku funkcie vypisujem aktuálny čas, kde si zistím aj milisekundy a časové pásmo [5]. Následne kontrolujem, či sa jedná o IP alebo ARP [6]. V prípade, že sa jedná o IP skontrolujem či ide o IPv4 alebo o IPv6. Ak sa jedná o IPv4 tak si zistím IP adresu zdroja a IP adresu cieľa pomocou funkcie *inet_ntoa()* [7]. V prípade, že sa jedná o IPv6 si taktiež zistím obe IP adresy ale tentokrát z funkcie *inet_ntop()* [8]. Následne pomocou switch kontrolujem, či sa jedná o TCP alebo UDP a taktiež vypisujem dané IP adresy + porty za spomínaným časom. Na záver volám funkciu *print_all()* pre výpis dát z paketu.

6. Funkcia *print_all()*

Funkcia sa skladá z hlavného for cyklu, ktorý robí výpis bajtov v ASCII alebo v hexadecimálnej podobe. Aby som dostal výstup aký bol v zadaní, tak medzi jednotlivé bajty vypisujem medzery.

3. Príklady výstupu programu

Vypísanie zoznamu aktívnych rozhraní pomocou argumentu -i bez hodnoty:

```
svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$ sudo ./ipk-sniffer -i
[sudo] password for svormen:
You entered: i without argument
The interfaces present on the system are:
0 : eth0
1 : lo
2 : any
3 : bluetooth-monitor
4 : nflog
5 : nfqueue
6 : dummy0
7 : sit0
8 : bond0
svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$
```

Príklad spustenia s argumentum -l + s rozhraním eth0 :

```
svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$ sudo ./ipk-sniffer -i eth0
2021-04-25T16:16:53.850+02:00 172.18.80.1 : 54915 > 172.18.95.255 : 54915, length 305 bytes

0x0000: FF FF FF FF FF FF 00 15 5D 5C 40 11 08 00 45 00 ..... ]\@...E.
0x0010: 01 23 FD 8F 00 00 80 11 34 15 AC 12 50 01 AC 12 ..#..... 4...P...
0x0020: 5F FF D6 83 D6 83 01 0F 51 F5 ..... Q.

0x0030: 00 44 45 53 4B 54 4F 50 2D 43 43 35 35 38 31 48 .DESKTOP -CC5581H
0x0040: 00 00 00 00 00 00 00 00 B0 57 D1 65 79 02 00 00 ..... .W.ey...
0x0050: 80 BE 2F 34 B5 00 00 00 20 50 10 66 79 02 00 00 ../4.... P.fy...
0x0060: 33 27 00 00 00 00 00 00 A0 57 D1 65 79 02 00 00 3'..... .W.ey...
0x0070: 70 D5 B8 62 79 02 00 00 40 C2 2F 34 B5 00 00 00 p..by... @./4....
0x0080: 40 C2 2F 34 B5 00 00 00 C6 74 3A 53 FD 7F 00 00 @./4.... .t:S....
0x0090: 07 01 00 00 00 00 00 00 60 C0 2F 34 B5 00 00 00 ..... `./4....
0x00100: 70 A2 6F 67 79 02 00 00 20 EF A9 65 79 02 00 00 p.ogy... ..ey...
0x00110: 08 6F 51 7B 33 35 30 37 35 61 61 36 2D 62 63 62 .oQ{3507 5aa6-bcb
0x00120: 31 2D 34 35 34 38 2D 61 31 66 62 2D 64 35 62 38 1-4548-a 1fb-d5b8
0x00130: 37 39 31 39 32 37 36 33 7D 00 B8 62 79 02 00 00 79192763 }..by...
0x00140: 60 C0 2F 34 1C 00 00 00 00 00 00 00 00 00 00 00 `./4.... .....
0x00150: 01 00 00 00 00 00 00 00 00 BF 2F 34 B5 00 00 00 ..... ..../4....
0x00160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x00170: 00 00 00 00 00 00 00 00 00 00 00 00 79 02 00 00 ..... ..y...
0x00180: B8 EB D8 2F 50 A6 00 00 8A 51 F9 49 FD 7F 00 00 .../P... .Q.I....
0x00190: 07 01 00 BE 86 E0 C4 .....

svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$ |
```

Príklad spustenia s argumentom --udp :

```
svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$ sudo ./ipk-sniffer -i eth0 --udp
2021-04-25T16:20:17.610+02:00 172.18.80.1 : 56956 > 239.255.255.250 : 1900, length 215 bytes

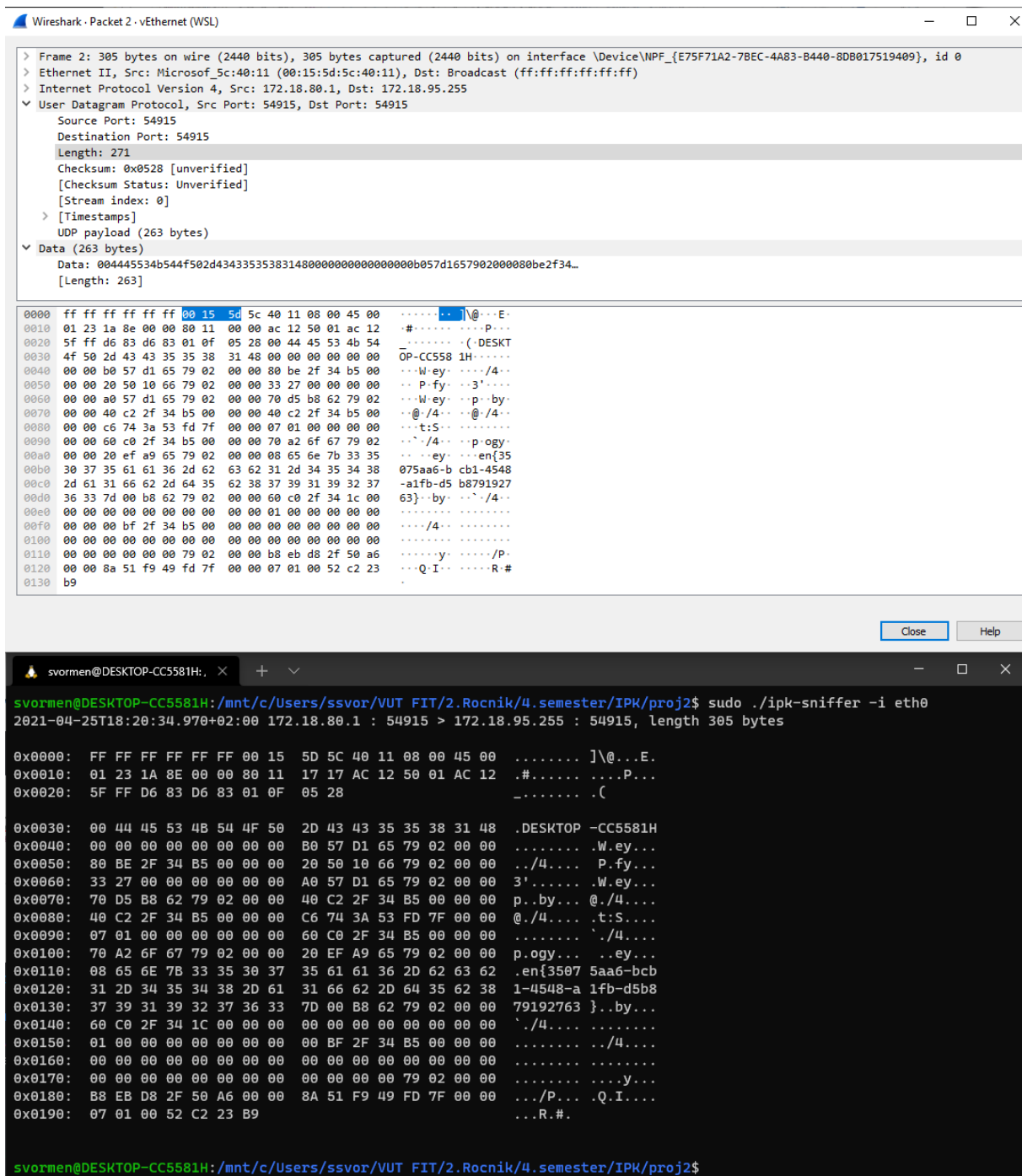
0x0000: 01 00 5E 7F FF FA 00 15 5D 5C 40 11 08 00 45 00 ..^..... ]\@...E.
0x0010: 00 C9 16 E0 00 00 01 11 B6 36 AC 12 50 01 EF FF ..... .6..P...
0x0020: FF FA DE 7C 07 6C 00 B5 F2 46 ...|.l.. .F

0x0030: 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F M-SEARCH * HTTP/
0x0040: 31 2E 31 0D 0A 48 4F 53 54 3A 20 32 33 39 2E 32 1.1..HOS T: 239.2
0x0050: 35 35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 55.255.2 50:1900.
0x0060: 0A 4D 41 4E 3A 20 22 73 73 64 70 3A 64 69 73 63 .MAN: "s sdp:disc
0x0070: 6F 76 65 72 22 0D 0A 4D 58 3A 20 31 0D 0A 53 54 over"..M X: 1..ST
0x0080: 3A 20 75 72 6E 3A 64 69 61 6C 2D 6D 75 6C 74 69 : urn:di al-multi
0x0090: 73 63 72 65 65 6E 2D 6F 72 67 3A 73 65 72 76 69 screen-o rg:servi
0x00100: 63 65 3A 64 69 61 6C 3A 31 0D 0A 55 53 45 52 2D ce:dial: 1..USER-
0x00110: 41 47 45 4E 54 3A 20 47 6F 6F 67 6C 65 20 43 68 AGENT: G oogle Ch
0x00120: 72 6F 6D 65 2F 39 30 2E 30 2E 34 34 33 30 2E 38 rome/90. 0.4430.8
0x00130: 35 20 57 69 6E 64 6F 77 73 0D 0A 0D 0A 5 Window s....

svormen@DESKTOP-CC5581H:/mnt/c/Users/ssvor/VUT FIT/2.Rocnik/4.semester/IPK/proj2$ sudo ./ipk-sniffer -i eth0 --udp
```

4. Testovanie programu

Výstup môjho programu som porovnal s výstupom s Wireshark.



The image shows two windows side-by-side. The top window is Wireshark, displaying a packet capture on interface \Device\NPF_{E75F71A2-7BEC-4A83-B440-80B017519409}. The selected packet is an Ethernet II frame from 172.18.80.1 to 172.18.95.255, containing a User Datagram Protocol (UDP) packet from port 54915 to port 54915. The data field shows a 263-byte payload. The bottom window is a terminal running a command: `sudo ./ipk-sniffer -i eth0`. The terminal output shows a timestamped log entry: `2021-04-25T18:20:34.970+02:00 172.18.80.1 : 54915 > 172.18.95.255 : 54915, length 305 bytes`. Below this, the terminal displays a hex dump of the captured data, which matches the hex dump shown in the Wireshark packet details pane. The hex dump starts with `0x0000: FF FF FF FF FF FF 00 15 5D 5C 40 11 08 00 45 00` and ends with `0x0190: 07 01 00 52 C2 23 B9`.

```
> Frame 2: 305 bytes on wire (2440 bits), 305 bytes captured (2440 bits) on interface \Device\NPF_{E75F71A2-7BEC-4A83-B440-80B017519409}, id 0
> Ethernet II, Src: Microsof_5c:40:11 (00:15:5d:5c:40:11), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 172.18.80.1, Dst: 172.18.95.255
  User Datagram Protocol, Src Port: 54915, Dst Port: 54915
    Source Port: 54915
    Destination Port: 54915
    Length: 271
    Checksum: 0x0528 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Timestamps]
    UDP payload (263 bytes)
  Data (263 bytes)
    Data: 004445534b544f502d4343353538314800000000000000b057d1657902000080be2f34...
    [Length: 263]

0000  ff ff ff ff ff ff 00 15 5d 5c 40 11 08 00 45 00  ..... ]\@...E.
0010  01 23 1a 8e 00 00 00 11 00 00 ac 12 50 01 ac 12  ..#.....P...
0020  5f ff d6 83 d6 83 01 0f 05 28 00 44 45 53 4b 54  _.....(..DESKT
0030  4f 50 2d 43 43 35 35 38 31 48 00 00 00 00 00 00  OP-CC558 1H.....
0040  00 00 b0 57 d1 65 79 02 00 00 80 be 2f 34 b5 00  ..W.ey.../4...
0050  00 00 20 50 10 66 79 02 00 00 33 27 00 00 00 00  ..P.fy...3'....
0060  00 00 a0 57 d1 65 79 02 00 00 70 d5 b8 62 79 02  ..W.ey...p.by...
0070  00 00 40 c2 2f 34 b5 00 00 00 40 c2 2f 34 b5 00  _@/4...@/4...
0080  00 00 c6 74 3a 53 fd 7f 00 00 07 01 00 00 00 00  ..t:S.....
0090  00 00 60 c0 2f 34 b5 00 00 00 70 a2 6f 67 79 02  ..../4...p.ogy...
00a0  00 00 20 ef a9 65 79 02 00 00 08 65 6e 7b 33 35  ..ey...en{35
00b0  30 37 35 61 61 36 2d 62 63 62 31 2d 34 35 34 38  075aa6-b cb1-4548
00c0  2d 61 31 66 62 2d 64 35 62 38 37 39 31 39 32 37  -a1fb-d5 b8791927
00d0  36 33 7d 00 b8 62 79 02 00 00 60 c0 2f 34 1c 00  63}..by.../4...
00e0  00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00  .....
00f0  00 00 00 bf 2f 34 b5 00 00 00 00 00 00 00 00 00  ..../4...
0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0110  00 00 00 00 00 00 79 02 00 00 b8 eb d8 2f 50 a6  ....y...../P...
0120  00 00 8a 51 f9 49 fd 7f 00 00 07 01 00 52 c2 23  ...Q.I.....R.#
0130  b9
```

```
svormen@DESKTOP-CC5581H: ~$ sudo ./ipk-sniffer -i eth0
2021-04-25T18:20:34.970+02:00 172.18.80.1 : 54915 > 172.18.95.255 : 54915, length 305 bytes

0x0000:  FF FF FF FF FF FF 00 15 5D 5C 40 11 08 00 45 00  ..... ]\@...E.
0x0010:  01 23 1A 8E 00 00 00 11 17 17 AC 12 50 01 AC 12  ..#.....P...
0x0020:  5F FF D6 83 D6 83 01 0F 05 28 00 44 45 53 4B 54  _.....(..DESKT
0x0030:  00 44 45 53 4B 54 4F 50 2D 43 43 35 35 38 31 48  .DESKTOP -CC5581H
0x0040:  00 00 00 00 00 00 00 00 B0 57 D1 65 79 02 00 00  ..... W.ey...
0x0050:  80 BE 2F 34 B5 00 00 00 20 50 10 66 79 02 00 00  ../4.... P.fy...
0x0060:  33 27 00 00 00 00 00 00 A0 57 D1 65 79 02 00 00  3'..... W.ey...
0x0070:  70 D5 B8 62 79 02 00 00 40 C2 2F 34 B5 00 00 00  p..by... @/4....
0x0080:  40 C2 2F 34 B5 00 00 00 C6 74 3A 53 FD 7F 00 00  @/4.... t:S....
0x0090:  07 01 00 00 00 00 00 00 60 C0 2F 34 B5 00 00 00  ..... /4....
0x00a0:  70 A2 6F 67 79 02 00 00 20 EF A9 65 79 02 00 00  p.ogy... ey...
0x00b0:  08 65 6E 7B 33 35 30 37 35 61 61 36 2D 62 63 62  .en{3507 5aa6-bcb
0x00c0:  31 2D 34 35 34 38 2D 61 31 66 62 2D 64 35 62 38  1-4548-a 1fb-d5b8
0x00d0:  37 39 31 39 32 37 36 33 7D 00 B8 62 79 02 00 00  79192763 }..by...
0x00e0:  60 C0 2F 34 1C 00 00 00 00 00 00 00 00 00 00 00  ..../4...
0x00f0:  01 00 00 00 00 00 00 00 00 BF 2F 34 B5 00 00 00  ..... /4....
0x0100:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0x0110:  00 00 00 00 00 00 00 00 00 00 00 00 00 79 02 00  ..... y....
0x0120:  B8 EB D8 2F 50 A6 00 00 8A 51 F9 49 FD 7F 00 00  .../P... Q.I....
0x0130:  07 01 00 52 C2 23 B9  ..... R.#.
```

```
svormen@DESKTOP-CC5581H: ~$
```

5. Zdroje

Pri implementácii projektu som využíval nasledujúce zdroje:

1. getopt example, *Example of Parsing Arguments with getopt* [online]. Dostupné z: https://www.gnu.org/software/libc/manual/html_node/Example-of-Getopt.html
2. Using getopt_long (C++) how do I code up a long & short option to both require arguments? *Stack Overflow - Where Developers Learn, Share, & Build Careers* [online]. Dostupné z: <https://stackoverflow.com/questions/8793020/using-getopt-long-c-how-do-i-code-up-a-long-short-option-to-both-require-a>
3. Programming with pcapTCPDUMP/LIBPCAP public repository, *Programming with pcapTCPDUMP/LIBPCAP public repository* [online]. Dostupné z: <https://www.tcpdump.org/pcap.html>
4. M. Casado, *dissect2.c* [online]. Dostupné z: <http://yuba.stanford.edu/~casado/pcap/dissect2.c>
5. How to use gettimeofday function in C language?, *Linux Hint* [online]. Dostupné z: https://linuxhint.com/gettimeofday_c_language/
6. ntohs(), *Help – Eclipse SDK* [online]. Dostupné z: http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fn%2Fntohs.html
7. inet_ntoa(3) – Linux man page, *inet_ntoa(3): Internet address change routines – Linux man page* [online]. Dostupné z: https://linux.die.net/man/3/inet_ntoa
8. inet_ntop(3) – Linux manual page, *inet_ntop(3) – Linux manual page* [online]. Dostupné z: https://man7.org/linux/man-pages/man3/inet_ntop.3.html

6. Záver

Projekt bol pre mňa celkom zaujímavý ale časovo náročný. Naučil som sa hromadu nových vecí. Napríklad ako pracovať s paketmi, ako ich odchytať. Taktiež som sa naučil viac používať program wireshark. Práca s TCP a UDP by mala pracovať v poriadku. Zachytávanie ARP rámcov a spracovavanie icmp paketov sa mi implementovať z časového hladiska bohužiaľ nepodarilo. (poučenie do budúcnosti).