



IPK - 2. projekt

Packet Sniffer

Obsah

Obsah	2
Zadání	3
Implementace	3
Jazyk	3
Vstupní argumenty	3
Rozhraní	3
Tvorba filtru	3
Filtrování paketů	3
Zpracování paketu	4
Ukončení programu	4
Testování	4
Ukázky manuálních testů	4
Použité zdroje	6

Zadání

Cílem projektu bylo navrhnout a implementovat síťový analyzátor v C/C++/C#, který bude schopný na určitém síťovém rozhraní 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 vypsaný všechny aktivní rozhraní pomocí funkce `print_devices()` a program se ukončí.

```
root@student-vm:/home/student/Desktop# ./proj2 -i

The interfaces present on the system are:
0 : enp0s3
1 : lo
2 : any
3 : bluetooth-monitor
4 : nflog
5 : nfqueue
root@student-vm:/home/student/Desktop#
```

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 *......stac
0x002c: 63 6B 6F 76 65 72 66 6C 6F 77 03 63 6F 6D 00 00 koverflow.com..
0x003c: 1C 00 01 00 00 29 02 00 00 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 *..♦....stac
0x002c: 63 6B 6F 76 65 72 66 6C 6F 77 03 63 6F 6D 00 00 kcoverflow.com..
0x003c: 1C 00 01 00 00 29 10 00 00 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 .static.net....
0x003c: 01 00 00 29 02 00 00 00 00 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

```

root@student-vm:/home/student/Desktop# ./proj2 -i enp0s3 -n 3 --tcp
2021-05-10T05:06:07.862+0200 10.0.2.15 : 50667 > 192.168.0.1 : 53, length 94 bytes

0x0000: 52 54 00 12 35 02 08 00 27 CA E4 D4 08 00 45 00 RT..5... '.....E.
0x0010: 00 50 61 84 .Pa.
0x0014: 40 00 40 11 0C 61 0A 00 @.@..a..

0x001c: 0C 90 01 00 00 01 00 00 00 00 00 01 0A 79 6F 75 .....you
0x002c: 74 75 62 65 2D 75 69 01 6C 06 67 6F 6F 67 6C 65 tube-ui. l.google
0x003c: 03 63 6F 6D 00 00 01 00 01 00 00 29 02 00 00 00 .com.... )....
0x004c: 00 00 00 00 ....

2021-05-10T05:06:07.862+0200 10.0.2.15 : 43706 > 216.58.201.78 : 443, length 762 bytes

0x0000: 52 54 00 12 35 02 08 00 27 CA E4 D4 08 00 45 00 RT..5... '.....E.
0x0010: 02 EC E4 4B ...K
0x0014: 40 00 40 06 A6 28 0A 00 02 0F D8 3A C9 4E AA BA @.@..(.. ....N..
0x0024: 01 BB C9 06 ....

0x0028: 17 03 03 02 BF A7 07 10 1A 76 3F 2E 8B C0 66 FB .....v?...f.
0x0038: B2 8C 59 06 AB 37 65 9B AF 3E AE C5 71 6D 2C 98 ..Y..7e. .>..qm,.
0x0048: 06 77 26 DB F4 FE 3F AC 48 89 53 99 4B 93 C7 91 .w&...?. H.S.K...
0x0058: 67 3A 8D 94 0D 13 94 E3 DF 2A 7B E5 82 E9 EB AF g:.....*{.....
0x0068: A3 D8 59 76 5C 96 79 BF 45 5C C4 4F 4B 31 5B 1F ..Yv\..y. E\..OK1[.
0x0078: 42 4A C1 6B 0A 2A BF 92 D3 5D 23 17 65 51 B6 BE BJ.k.*.. ]#.eQ..
0x0088: 9C 25 F7 DF C9 6D 51 BF 5D 48 74 10 E8 75 0D DE .%...mQ. ]Ht..u..
0x0098: C8 17 6D 00 15 B1 ED B4 17 B2 BF 6C 49 B3 3D 67 ..`..... ..lI.=g
0x00a8: D0 11 05 6D 6D 5A 81 F2 15 DC 5E A5 91 35 EC D1 ...mmZ.. ..^..S..
0x00b8: FD 09 C9 3D B3 2A C2 8F 57 1C D2 95 68 64 FA 00 ...=,*.. W...hd..
0x00c8: A5 6C 4E 54 2D 92 88 70 86 06 34 2F E7 3B AB 18 .lNT-..p ..4/..;..
0x00d8: 51 6D 68 F7 65 74 89 B5 C3 4C 1E 2D EA B7 D6 CF Qmh.et.. .L.-....
0x00e8: 61 87 75 85 9E FC 15 9D F4 C6 F1 BB 2C 8F 75 2A a.u..... ,.u*
0x00f8: F6 CC A6 D4 0E 37 DE F8 29 56 75 44 3D 8F 70 CC .....7.. )VuD=.p.
0x0108: 39 B1 64 E9 DD 4B 42 D7 E5 69 F5 61 54 D3 6B 3C 9.d...KB. .i.aT.k<
0x0118: D4 06 29 ED 03 EB 2F 6F 0D 70 51 0B 61 5D 21 26 ..).../o .pQ.a]!&
0x0128: 3B E9 6E B7 24 A0 D6 F6 B9 59 61 7B E7 54 0F 47 ;.n.$... .Ya{.T.G
0x0138: 00 B9 5B 27 50 95 8C 83 66 28 F8 07 50 41 BC FD ..['P... f(..PA..
0x0148: 9A 1C 52 40 3D 9D 38 FB 95 85 18 C4 DF C8 A3 2B ..R@=.8. ....+
0x0158: 81 5F 91 89 A4 02 36 8F 85 CB CE 0B A5 1A 43 40 ._. ....6. ....C@
0x0168: 16 BC 7C 1A D7 76 6A FB 77 A0 55 E9 3C 99 03 50 ..|...vj. w.U.<..P
0x0178: 76 47 0C 82 7F 64 CF 87 B8 16 15 2D 0B 03 37 67 vG..d.. ...7g
0x0188: 09 63 11 C7 FC 9C F5 39 79 83 33 5E 3E 47 81 E2 .c.....9 y.3^>G..
0x0198: 94 E4 AC 0B 99 0C 1F CF 0A EA 5F 8B B3 39 4C 2A ..... ..9L*
0x01a8: CA F1 34 27 42 E4 7C 49 68 4D A1 09 55 3A 65 55 ..4'B. |I hM..U:eU
0x01b8: BE F9 84 B9 4F 62 E2 4A D4 CE EF 57 F2 EA 78 F6 ....Ob.J ...W..x.
0x01c8: 0F 29 F1 47 39 C5 0E B3 A5 1D 03 B5 8F AA 01 4C .).G9... .....L

```

TCP pakety ze sledování YouTube

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>