

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



IPK – Sniffer paketů

IPK – Počítačové komunikace a sítě

Vypracoval/-a: Ladislav Dokupil

Čestně prohlašuji, že jsem tuto práci vypracoval/-a samostatně a pouze za využití pramenů, zmíněných v závěru práce.

Obsah

1 Úvod	3
2 Implementace	4
2.1 Hlavní funkce	4
2.2 Zpracování Argumentů	4
2.3 Výpis paketů	4
3 Testování	5
4 Zdroje	6

1 Úvod

Cílem projektu bylo vytvořit síťový analyzátor, který bude schopný na určitém síťovém rozhraní zachytávat a filtrovat pakety. Následně k němu bylo nutné vytvořit manuál a dokumentaci.

Analyzátor je schopen zachytávat všechny pakety na vybraném rozhraní. Typy paketů, které se budou zachytávat, lze nastavit pomocí předdefinovaných argumentů příkazové řádky. Výpis zachycených paketů probíhá na *stdout* ve podobě hexadecimální a ASCII znaků.

2 Implementace

K implementaci je využita knihovna *libpcap* jazyka C, která je využita k samotnému zachytávání a filtrování paketů. Dále jsou použity knihovny *netinet*, které definují hlavičky různých protokolů a pomocné funkce pro výpis IP adresy z paketů.

2.1 Hlavní funkce

Funkce `main()` zde slouží ke spojení různých částí programu. Nejprve zavolá funkci `parseArgs()`, ze které dostane počet paketů, rozhraní a filtr ke sledování. Následně pomocí knihovny *pcap* otevře rozhraní a aplikuje na něj daný filtr, nakonec zavolá funkci `pcap_loop()`, která nad každým paketem odpovídajícím filtru spustí funkci `packet_handler()`, dokud není zpracováno právě tolik paketů, kolik uživatel zadal.

2.2 Zpracování Argumentů

O vstupy programu se stará funkce `parseArgs()`. Ta z argumentů programu skládá textový řetězec, který je následně využit pro filtrování přijímaných paketů. Uživatel má pomocí argumentů možnost filtrovat TCP, UDP, ARP, ICMP komunikaci, nebo jakoukoliv jejich kombinaci, lze specifikovat i port, který je v případě filtrování pouze ARP a ICMP komunikace ignorován. Pokud je program spuštěn se samotným přepínačem `-i`, vypíše se seznam rozhraní, na kterých lze naslouchat a následně se program ukončí. Pokud je program spuštěn bez jakýchkoliv možností, nebo obsahuje nedefinovaný argument, ukončí se s chybovým kódem 1. Přepínač `-n` určuje počet paketů, které budou zachyceny, v případě záporného nebo nulového čísla se jedná o nekonečnou smyčku.

2.3 Výpis paketů

Zpracování a výpis paketů je prováděn ve funkci `packet_handler()`. Ta nejdříve vypíše datum zachycení paketu, poté se vypíše řádek tvaru: “IP_ZDROJ: PORT_ZDROJ > IP_CIL: PORT_CIL, length DELKA”, kde IP a PORT jsou vynechány, pokud je protokol nevyužívá. V případě IPv6 paketu je IP vypsána ve formátu RFC5952. Následně se nad paketem volá funkce `hexdump()`, která celý obsah paketu vypíše v hexadecimální a následně ASCII podobě, kde netisknutelné znaky jsou nahrazeny znakem tečka.

3 Testování

Testování funkčnosti analyzátoru bylo prováděno pomocí přímého porovnání výstupu programu s open source programem *wireshark*. Pro generování paketů byly použity nástroje jako *curl*, *ping*, nebo náhodné procházení webu pomocí prohlížeče. Následně byl otestován překlad programu a funkčnost na referenčním virtuálním stroji.

```
[root@L: ~]# ./ipk-sniffer -i wlp3s0 -n 2 --icmp
date: 2021-04-25T10:07:07.27+02:00
2002:c0a8:968:1::1000: -> 2002:c0a8:968::c0a8:968: -, length: 118
0x0000: 40 3f 8c cb b9 a2 d0 53 49 48 6f 3b 86 dd 60 0f @?.....S IHo;...`
0x0010: b1 5e 00 40 3a 40 20 02 c0 a8 09 68 00 01 00 00 .^.@:@ . ...h....
0x0020: 00 00 00 00 10 00 20 02 c0 a8 09 68 00 00 00 00 ..... .h....
0x0030: 00 00 c0 a8 09 68 80 00 4f 0e 00 06 00 01 2b 23 .....h.. 0.....+
0x0040: 85 60 00 00 00 00 08 e2 0a 00 00 00 00 00 10 11 .....
0x0050: 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 .....
0x0060: 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 "#$%&'()*+,-./01
0x0070: 32 33 34 35 36 37 234567
date: 2021-04-25T10:07:07.27+02:00
fe80::423f:8cff:febc:b9a2: -> ff02::1:ff00:1000: -, length: 86
0x0000: 33 33 ff 00 10 00 40 3f 8c cb b9 a2 86 dd 60 00 33....@? .....`
0x0010: 00 00 00 20 3a ff fe 80 00 00 00 00 00 00 42 3f ...:.... ....B?
0x0020: 8c ff fe cb b9 a2 ff 02 00 00 00 00 00 00 00 00 .....
0x0030: 00 01 ff 00 10 00 87 00 62 af 00 00 00 00 20 02 ..... b.....
0x0040: c0 a8 09 68 00 01 00 00 00 00 00 00 10 00 01 01 ...h.... ....
0x0050: 40 3f 8c cb b9 a2 @?.....
```

Obrázek 1 výstup analyzátoru

No.	Time	Source	Destination	Protocol	Length	Info
8	1.908100626	2002:c0a8:968:1::1000	2002:c0a8:968::c0a8:968	ICMPv6	118	Echo (ping) request id=0x0006, seq=1, ho...
9	1.944359430	fe80::423f:8cff:febc:b9a2	ff02::1:ff00:1000	ICMPv6	86	Neighbor Solicitation for 2002:c0a8:968:1...
10	1.944388334	2002:c0a8:968:1::1000	fe80::423f:8cff:febc:b9a2	ICMPv6	86	Neighbor Advertisement 2002:c0a8:968:1...
11	1.946824484	2002:c0a8:968::c0a8:968	2002:c0a8:968:1::1000	ICMPv6	118	Echo (ping) reply id=0x0006, seq=1, ho...
22	2.909937927	2002:c0a8:968:1::1000	2002:c0a8:968::c0a8:968	ICMPv6	118	Echo (ping) request id=0x0006, seq=2, ho...
23	2.915687775	2002:c0a8:968::c0a8:968	2002:c0a8:968:1::1000	ICMPv6	118	Echo (ping) reply id=0x0006, seq=2, ho...
90	21.197091992	192.168.0.107	192.168.0.1	ICMP	120	Destination unreachable (Port unreacha...
92	21.197112068	192.168.0.107	192.168.0.1	ICMP	120	Destination unreachable (Port unreacha...
1563	46.445050851	fe80::99c3:4a9:89f6:28a1	fe80::423f:8cff:febc:b9a2	ICMPv6	86	Neighbor Solicitation for fe80::423f:8...
1564	46.448261360	fe80::423f:8cff:febc:b9a2	fe80::99c3:4a9:89f6:28a1	ICMPv6	78	Neighbor Advertisement fe80::423f:8...

Frame 8: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface wlp3s0, id 0
Ethernet II, Src: LiteonTe_48:6f:3b (d0:53:49:48:6f:3b), Dst: Tp-LinkT_cb:b9:a2 (40:3f:8c:cb:b9:a2)
Internet Protocol Version 6, Src: 2002:c0a8:968:1::1000, Dst: 2002:c0a8:968::c0a8:968
Internet Control Message Protocol v6

```
0000 40 3f 8c cb b9 a2 d0 53 49 48 6f 3b 86 dd 60 0f @?.....S
0010 b1 5e 00 40 3a 40 20 02 c0 a8 09 68 00 01 00 00 .^.@:@ . ...h....
0020 00 00 00 00 10 00 20 02 c0 a8 09 68 00 00 00 00 .....
0030 00 00 c0 a8 09 68 80 00 4f 0e 00 06 00 01 2b 23 .....h..
0040 85 60 00 00 00 00 08 e2 0a 00 00 00 00 00 10 11 .....
0050 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 .....
0060 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 "#$%&'()*+,-./
0070 32 33 34 35 36 37 234567
```

Obrázek 2 vzorový výstup wiresharku (horní 2 pakety)

4 Zdroje

Tcpdump&libpcap. *Tcpdump* [online]. 2010 [cit. 2021-4-24]. Dostupné z: <https://www.tcpdump.org/pcap.html>

RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL. *Tools.ietf.org* [online]. 1981 [cit. 2021-4-24]. Dostupné z: <https://tools.ietf.org/html/rfc792>

RFC 4443 - Internet Control Message Protocol (ICMPv6). *Tools.ietf.org* [online]. 2006 [cit. 2021-4-24]. Dostupné z: <https://tools.ietf.org/html/rfc4443>

RFC 826 - An Ethernet Address Resolution Protocol. *Tools.ietf.org* [online]. 1982 [cit. 2021-4-24]. Dostupné z: <https://tools.ietf.org/html/rfc826>