

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMANÍCH TECHNOLOGIÍ

Počítačové komunikace a sítě – 2. projekt
Varianta ZETA: Sniffer paketů

24. dubna 2022

Kravchuk Marina (xkravc02)

1 Úvod

Cílem řešeného projektu byl návrh a implementace síťového analyzátoru. Tento analyzátor je schopný na určitém síťovém rozhraní zachytávat a filtrovat pakety.

Program podporuje zachytávání paketů na protokolu TPC, UDP, ICMP a ARP. Dále podporuje filtrování vypisovaných paketů podle protokolu, portu a rozhraní. Pakety se vypisují na stdout s oddělenou úvodní hlavičkou, která obsahuje přesný systémový čas zachycení paketu, MAC adresy, IP adresy a porty zdroje a cíle.

Program lze spustit s následujícími parametry:

```
./ipk-sniffer [-i rozhraní nebo -interface rozhraní] [-p port] [- -tcp nebo -t] [- -udp nebo -u] [- -arp] [- -icmp] [-n num]
```

kde:

- -i eth0 (právě jedno rozhraní, na kterém se bude poslouchat. Nebude-li tento parametr uveden, či bude-li uveden jen -i bez hodnoty, vypíše se seznam aktivních rozhraní)
- -p 23 (bude filtrování paketů na daném rozhraní podle portu; nebude-li tento parametr uveden, uvažují se všechny porty; pokud je parametr uveden, může se daný port vyskytnout jak v source, tak v destination části)
- -t nebo - -tcp (bude zobrazovat pouze TCP pakety)
- -u nebo - -udp (bude zobrazovat pouze UDP pakety)
- - -icmp (bude zobrazovat pouze ICMPv4 a ICMPv6 pakety)
- - -arp (bude zobrazovat pouze ARP rámce)
- -n 10 (určuje počet paketů, které se mají zobrazit, tj. i "dobu" běhu programu; pokud není uvedeno, uvažujte zobrazení pouze jednoho paketu, tedy jakoby -n 1)

2 Implementace

Pro tento projekt byl zvolen jazyk C. Pro sniffing je použita knihovna Pcap. Většina informací k tomuto projektu byla nalezena na stránkách tcpdump.org. Program se skládá ze tří částí: parsování argumentů, práce s funkcemi knihovny pcap a vypisování informací o paketech.

2.1 Parsování argumentů

Zpracování argumentů se nachází ve hlavní funkci main(). V cyklu projdeme přes všechny argumenty a naplníme příslušné proměnné hodnotami.

2.2 Práce s funkcemi knihovny pcap

Funkce main() volá funkci open_pcap_socket(). Tato funkce slouží zejména k získání a přípravě socket deskriptoru pro sniffer paketů. Tato funkce je posloupností čtyř dalších funkcí knihovny pcap.h: pcap_lookupnet – zjistí číslo sítě IPv4 a masku sítě pro zařízení. pcap_open_live(), která otevře vybrané rozhraní pro síťové přenosy a vrátí potřebný deskriptor. Dále je zapotřebí aplikovat filtr. K tomu slouží následující dvojice knihovnických funkcí. pcap_compile převede námi vytvořený řetězec filtrů na knihovnou interpretovatelný kód, který

může být aplikován pomocí funkce `pcap_setfilter()`. Teď deskriptor je připraven ke zachytání a filtrování paketů.

Následně je volána `pcap_loop()`. Ta slouží k samotnému „sniffování“ paketů. Jako jeden z argumentů si bere ukazatel na funkci (tzv. „callback“ funkci), pomocí které analyzujeme jednotlivé pakety.

2.3 Vypisování informací o paketech

Pro vypis paketů byli implementovány dvě funkce `print_packet_info` a `print_data`.

První funkce zajišťuje analýzu příchozího paketu. Má za úkol vypsat úvodní hlavičku dle zadání projektu a následně spustit funkci pro výpis dat. Druhá funkce vypíše jednotlivé bity paketu v řádcích po 16 znaků a jejich ASCII reprezentaci.

```
timestamp: 2022-04-16T16:40:05.023+02:00
dst MAC:   8C:85:90:69:C1:0F
src MAC:   C0:C9:E3:32:D2:AA
frame length: 74 bytes
src ip: 172.224.51.9
dst ip: 192.168.0.102
src port: 17828
dst port: 60

0x0000:  8C 85 90 69 C1 0F C0 C9  E3 32 D2 AA 08 00 45 A4  ...i.... .2....E.
0x0010:  00 3C 53 EF 40 00 36 11  4F 26 AC E0 33 09 C0 A8  .<S.@.6. 0&..3...
0x0020:  00 66 01 BB FE CE 00 28  42 E7 4A A9 42 FC 5D 76  .f.....( B.J.B.]v
0x0030:  ED 8D 55 31 21 E0 98 59  31 E8 0F 0E AF 3A 9F 93  ..U1!..Y 1.....:..
0x0040:  C7 B6 31 45 E0 E8 0D 59  BC 1D                ..1E...Y ..
```

Obrázek 1: Příklad vypsání paketu

3 Testování programu

Výstup programu byl testován porovnáním s výstupem programu Wireshark. Wireshark je široce používaný analyzátor síťových protokolů.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|----------------|-------------------|----------|--------|--------------|
| 8 | 0.077012 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 9 | 0.077030 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 10 | 0.077034 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 706 | 25.219974 | 34.120.186.93 | 192.168.0.102 | TLSv1... | 93 | Application |
| 711 | 25.236138 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 712 | 25.237483 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 713 | 25.300560 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63830 |
| 756 | 36.236189 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63798 |
| 757 | 36.242413 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63798 |
| 758 | 36.242433 | 34.120.186.93 | 192.168.0.102 | TCP | 66 | 443 → 63798 |
| 567 | 14.572106 | 95.161.76.100 | 192.168.0.102 | TCP | 66 | 443 → 51806 |
| 564 | 13.333029 | Apple_69:c1:0f | Tp-LinkT_32:d2:aa | ARP | 42 | 192.168.0.10 |
| 794 | 41.306686 | Apple_69:c1:0f | Tp-LinkT_32:d2:aa | ARP | 42 | 192.168.0.10 |

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, ID 0

Ethernet II, Src: Tp-LinkT_32:d2:aa (c0:c9:e3:32:d2:aa), Dst: Apple_69:c1:0f (8c:85:90:69:c1:0f)

Destination: Apple_69:c1:0f (8c:85:90:69:c1:0f)

Address: Apple_69:c1:0f (8c:85:90:69:c1:0f)

....0. = LG bit: Globally unique address (factory default)

....0. = IG bit: Individual address (unicast)

| | | | |
|------|-------------------------|-------------------------|-------------------|
| 0000 | 8c 85 90 69 c1 0f c0 c9 | e3 32 d2 aa 08 00 45 00 | ...i.... .2....E. |
| 0010 | 00 34 96 88 00 00 7a 06 | 0c 58 22 78 ba 5d c0 a8 | .4....z. .X"x.].. |
| 0020 | 00 66 01 bb f9 56 11 00 | b5 60 c0 20 e9 bb 80 10 | .f...V.. .'. |
| 0030 | 04 0c 51 16 00 00 01 01 | 08 0a fd 24 d6 7d 16 de | ..Q..... ..\$.}.. |
| 0040 | 2d e7 | | .. |

Specifies if this is an individual (unicast) or group (broadcast/multicast) address (eth.dst.ig), 3 bytes

Packets: 476

timestamp: 2022-04-16T19:54:10.143+02:00

dst MAC: 8C:85:90:69:C1:0F

src MAC: C0:C9:E3:32:D2:AA

frame length: 66 bytes

src ip: 34.120.186.93

dst ip: 192.168.0.102

src port: 17664

dst port: 52

| | | | |
|---------|-------------------------|-------------------------|-------------------|
| 0x0000: | 8C 85 90 69 C1 0F C0 C9 | E3 32 D2 AA 08 00 45 00 | ...i.... .2....E. |
| 0x0010: | 00 34 96 88 00 00 7A 06 | 0C 58 22 78 BA 5D C0 A8 | .4....z. .X"x.].. |
| 0x0020: | 00 66 01 BB F9 56 11 00 | B5 60 C0 20 E9 BB 80 10 | .f...V.. .'. |
| 0x0030: | 04 0C 51 16 00 00 01 01 | 08 0A FD 24 D6 7D 16 DE | ..Q..... ..\$.}.. |
| 0x0040: | 2D E7 | | .. |

Obrázek 2: Porovnání vypisu paketu

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------------|-------------------|----------|--------|----------------------|
| 563 | 13.332961 | Tp-LinkT_32:d2:aa | Apple_69:c1:0f | ARP | 42 | Who has 192.168.0.10 |
| 564 | 13.333029 | Apple_69:c1:0f | Tp-LinkT_32:d2:aa | ARP | 42 | 192.168.0.102 is at |
| 793 | 41.306497 | Tp-LinkT_32:d2:aa | Apple_69:c1:0f | ARP | 42 | Who has 192.168.0.10 |
| 794 | 41.306686 | Apple_69:c1:0f | Tp-LinkT_32:d2:aa | ARP | 42 | 192.168.0.102 is at |
| 17 | 1.808366 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 22 | 1.830269 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 177 | 4.014550 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 179 | 4.630469 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 187 | 5.656034 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 562 | 12.544935 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 566 | 14.360407 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 605 | 16.047993 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |
| 608 | 16.518497 | 192.168.0.102 | 162.159.130.234 | TCP | 54 | 51814 → 443 [ACK] Se |

| |
|--|
| ▼ Ethernet II, Src: Apple_69:c1:0f (8c:85:90:69:c1:0f), Dst: Tp-LinkT_32:d2:aa (c0:c9:e3:32:d2:aa) |
| ▼ Destination: Tp-LinkT_32:d2:aa (c0:c9:e3:32:d2:aa) |
| Address: Tp-LinkT_32:d2:aa (c0:c9:e3:32:d2:aa) |
|0. = LG bit: Globally unique address (factory default) |
|0 = IG bit: Individual address (unicast) |

| | | |
|------|---|------------------|
| 0000 | c0 c9 e3 32 d2 aa 8c 85 90 69 c1 0f 08 06 00 01 | ...2.... .i..... |
| 0010 | 08 00 06 04 00 02 8c 85 90 69 c1 0f c0 a8 00 66 |i.....f |
| 0020 | c0 c9 e3 32 d2 aa c0 a8 00 01 | ...2.... .. |

Specifies if this is an individual (unicast) or group (broadcast/multicast) address (eth.dst.ig), 3 bytes
Packets: 4761 · Display

```

timestamp: 2022-04-16T19:53:27.786+02:00
dst MAC: C0:C9:E3:32:D2:AA
src MAC: 8C:85:90:69:C1:0F
frame length: 42 bytes
src ip: 193.15.192.168
dst ip: 0.102.192.201

ARP PACKET

0x0000: C0 C9 E3 32 D2 AA 8C 85 90 69 C1 0F 08 06 00 01 ...2.... .i.....
0x0010: 08 00 06 04 00 02 8C 85 90 69 C1 0F C0 A8 00 66 ..... .i.....f
0x0020: C0 C9 E3 32 D2 AA C0 A8 00 01 ...2.... ..

```

Obrázek 3: Porovnání vypisu paketu

4 Zdroje

Při zpracovávání projektu jsem využila následující internetové zdroje.

1. <https://www.tcpdump.org/index.html>
2. <https://ru.wikipedia.org/wiki/ARP>
3. <https://docs.huihoo.com/doxygen/linux/kernel/3.7/index.html>
4. https://www.winpcap.org/docs/docs_412/html/structpcap__pkthdr.html
5. <http://rus-linux.net/MyLDP/algol/libpcap.html>
6. <http://www.netcode.ru/cpp/?artID=5143>
7. https://www.winpcap.org/docs/docs_412/html/group__wpcap__tut5.html