VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Počítačové komunikace a sítě – 2. projekt

Síťový analyzátor

3. května 2020

Michal Sova

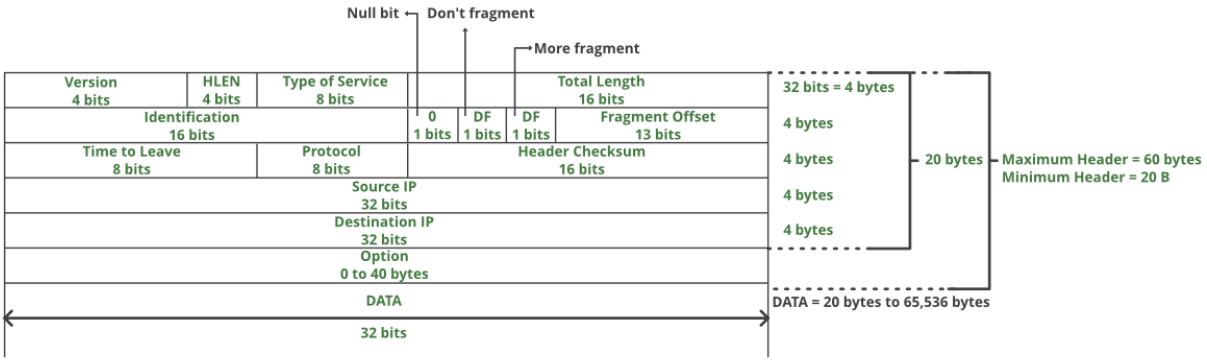
Obsah

1 Implementace

Analyzátor nejdříve zpracuje argumenty programu[1]. Otevře rozhraní pro zachytávání paketů, nastaví filtr a pomocí funkce `pcap_loop` a `callback` funkce[2]. V `callback` funkci nejdříve zkontroluje verzi IP, poté podle příslušné verze zavolá funkci `getAddress`. Tato funkce zjistí z IP nebo IPv6 hlavičky typ protokolu zdrojovou a cílovou adresu a velikost IP hlavičky. Podle typu protokolu se získá z TCP nebo UDP zdrojový a cílový port. Následně vše vypíše na výstup. Výpis paketu je mezerou rozdělen na hlavičku a zbytek paketu.

1.1 IP

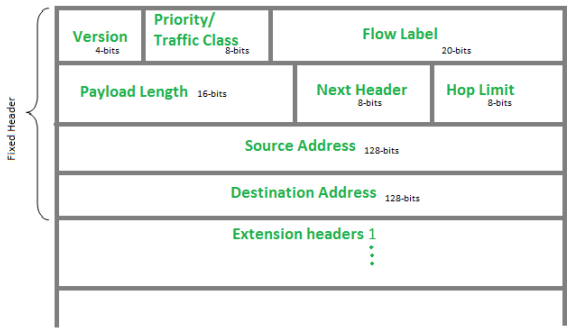
V IP hlavičce se analyzátor zaměřuje především na verzi(4 bity). Dále se zaměřuje na velikost hlavičky(4 bity), která po vynásobení 4 udává velikost IP hlavičky v bajtech. Analyzátor také vyčte z hlavičky zdrojovou IP adresu, cílovou IP adresu a typ další hlavičky (TCP/UDP).



Obrázek 1: Hlavička IPv4[1]

1.2 IPv6

IPv6 má pevnou velikost hlavičky, která je 40 bajtů. Analyzátor pro určení další hlavičky se dívá na Next Header, kde by tato informace měla být obsažená.



Obrázek 2: Hlavička IPv6[2]

1.3 Transmission Control Protocol a User Datagram Protocol

V hlavičkách TCP a UDP protokolů lze nalézt zdrojové a cílové porty. TCP a UDP protokoly by měly mít konstantní, avšak velikostně různé, hlavičky.

2 využití knihoven

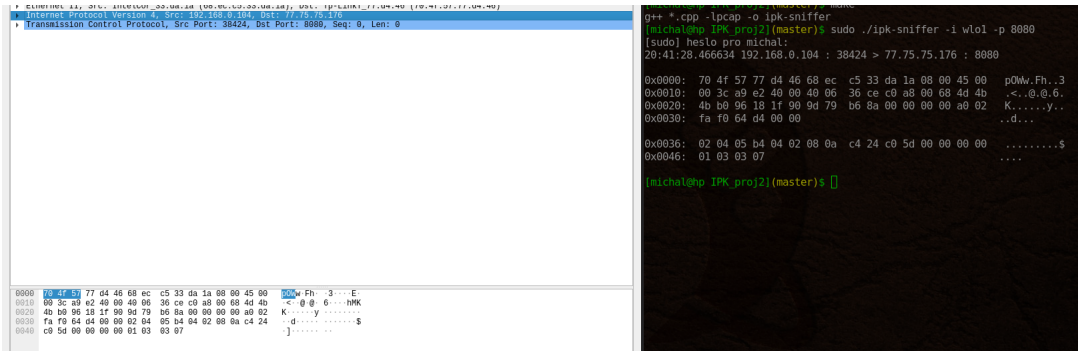
Bylo využito knihovny pcap.h a jejich funkcí pro zachytávaní paketů. Knihovny netinet/ip.h[3] pro použití struktury `struct ip`, ze které analyzátor načítá adresy a kontroluje další hlavičku. Podobně díky knihovně netinet/ip6.h[4] a struktuře `struct ip6_hdr`. Z knihoven netinet/tcp.h[5] a netinet/udp.h[6] byly využity struktury `struct tcphdr` a `struct udphdr`.

3 Testování

K testování bylo využito nástroje Wireshark[1].

3.1 Příklad testu

Na Wiresharku byl nastaven filtr: `(tcp port 8080) or (udp port 8080)` na rozhraní `wlo1` a program byl spuštěn s následujícími parametry: `./ipk-sniffer -i wlo1 -p 8080` pro vyfiltrování stejných paketů. Poslání paketu na tomto portu bylo vyvoláno příkazem curl, konkrétně: `curl "http://seznam.cz:8080"`.



Obrázek 3: Příklad testu

Jak lze vidět na obrázku 3 pakety jsou stejné, pomineme-li odsazení hlavičky. Rovněž adresy a porty jsou stejné.

[1] https://www.wireshark.org

4 Použité zdroje

Reference

[1] Ashwin, V., 2015. How To Parse Program Options In C++ Using Getopt_Long. [online] Code Yarns. Dostupné z:https://codeyarns.com/2015/01/30/how-to-parse-program-options-in-c-using-getopt_long/

[2] Carstens, T., 2020. Programming With Pcaptcpdump/LIBPCAP Public Repository. [online] Tcpdump.org. Dostupné z: https://www.tcpdump.org/pcap.html

[3] Unix.superglobalmegacorp.com. n.d. Netinet/Ip.H Source. [online] Dostupné z: https://unix.superglobalmegacorp.com/Net2/newsrc/netinet/ip.h.html

[4] Code.woboq.org. n.d. Ip6.H Source Code [Glibc/Inet/Netinet/Ip6.H] - Woboq Code Browser. [online] Dostupné z: https://code.woboq.org/userspace/glibc/inet/netinet/ip6.h.html

[5] Unix.superglobalmegacorp.com. n.d. Netinet/Tcp.H Source. [online] Dostupné z: https://unix.superglobalmegacorp.com/BSD4.4/newsrc/netinet/tcp.h.html

[6] Unix.superglobalmegacorp.com. n.d. Netinet/Udp.H Source. [online] Available at: https://unix.superglobalmegacorp.com/Net2/newsrc/netinet/udp.h.html

4.1 Obrázky

Reference

[1] GeeksforGeeks. n.d. Introduction And Ipv4 Datagram Header - Geeksforgeeks. [online] Dostupné z: https://www.geeksforgeeks.org/introduction-and-ipv4-datagram-header/

[2] GeeksforGeeks. n.d. Internet Protocol Version 6 (Ipv6) Header - Geeksforgeeks. [online] Dostupné z: https://www.geeksforgeeks.org/internet-protocol-version-6-ipv6-header/

5 Upřesnění chování analyzátoru

- Analyzátor ignoruje neznámé argumenty.
- Analyzátor podporuje je protokoly tcp a udp.

5.1 Omezení

- Analyzátor nepřekládá IP adresy na doménové jména.