VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Počítačové komunikace a sítě ZETA - Sniffer paketů

Obsah

1	Úvod
2	Naštudovaná literatúra
	Naštudovaná literatúra 2.1 IPv4
	2.2 UDP
	2.3 TCP
3	Implementácia
	Implementácia 3.1 Spracovanie argumentov
	3.2 Odchytávanie paketov
1	Testovanie

1 Úvod

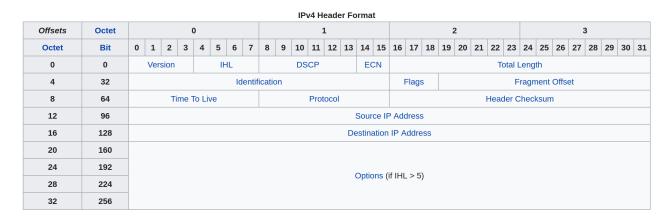
Cieľom projektu bolo navrhnúť a implementovať sieťový analyzátor, ktorý dokáže na danom sieťovom rozhraní zachytávať a filtrovať pakety. Informácie o packetoch (čas odchytenia, zdroj a ciel) a telo packetu sú vypísané na štandardný výstup. Ak sa podarí preložť IP adresu na hostname tak je vypísaný hostname, v ostatných prípadoch IP adresa.

2 Naštudovaná literatúra

Pre účely vypracovania projektu bolo potrebné si naštudovať protokoly transportnej a sieťovej vrstvy OSI modelu sieťovej komunikácie.

2.1 IPv4

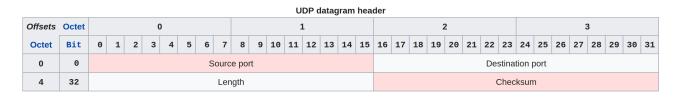
IPv4 je verzia 4 Internet Protocolu. Je opísaný v IETF RFC 791. IPv4 používa 32-bitové adresy. Datagram je uložený vo veľkom endiane. [1]



Obrázek 1: IPv4 formát hlavičky [1]

2.2 UDP

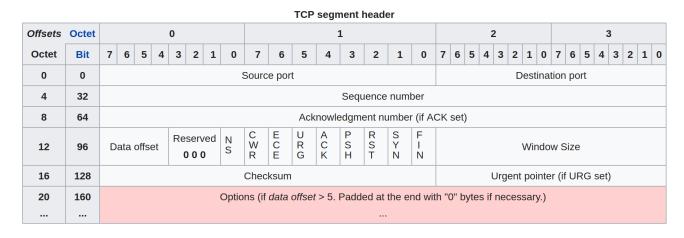
UDP hlavička sa skladá zo 4 polí. Polia zdrojového a cieľového portu sú 16bitové a identifikují odosielajúci a prijímajúci proces. Po číslach portov nasleduje povinná dĺžka UDP paketu vrátane dát, v bajtoch. Minimálna hodnota je 8 bajtov. Zvyšné pole hlavičky je 16bitový kontrolný súčet pokrývajúci hlavičku aj dáta. [2][3]



Obrázek 2: UDP formát hlavičky [2]

2.3 TCP

Aplikácie posielajú siefou pomocou TCP toky dát s 8-bitovými slabikami a TCP rozdeľuje tok bajtov do segmentov s vhodne zvolenou veľkosťou. TCP potom podáva výsledné pakety Internet protokolu na doručenie internetom TCP modulu na opačnom konci spojenia. TCP vykonáva kontrolu, aby sa uistil, že sa žiaden paket nestratí tak, že dá každému paketu poradové číslo, ktoré na druhom konci opäť TCP modul kontroluje a zabezpečuje tiež, že dáta sú doručené v správnom poradí. TCP spojenie má tri fázy: nadviazanie spojenia, prenos dát a ukončenie spojenia. Na nadviazanie spojenia sa používa tzv. 3-way handshake. 4-way handshake sa používa na ukončenie spojenia. [3][4]



Obrázek 3: TCP formát hlavičky [4]

3 Implementácia

Implementácia bola rozdelená do dvoch celkov, spracovanie argumentov a odhytávanie paketov. Hlavný súbor programu je main.c, z ktorého sa volajú nosné funkcie modulov. V súbore error.h sú zadefinované chybové kódy, ktoré program vracia ak nastane jedna z chýb. Program sa prekladá za pomoci nástroja **Make**. Príkazy na preloženie sú definované v súbore Makefile. Ak nie je programu odovzdaný parameter *interface*, program vypíše všetky rozhrania, ktoré nájde funkcia pcap_findalldevs(). Celková štruktúra programu na odchytávanie packetov bola inšpirovaná zdrojmi [5],[6].

3.1 Spracovanie argumentov

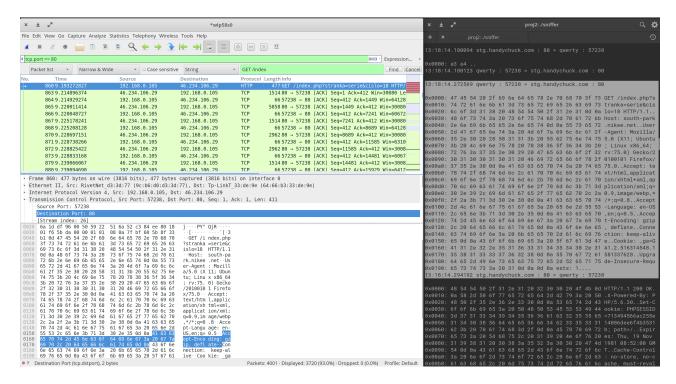
Implementované v súboroch args.c a args.h. Spracovanie argumentov prebieha za pomoci využitia knižnice getopt.h. Používa sa funkcia getopt_long() z dôvodu podpory dlhých (--opt) ako aj krátkych (-o) argumentov. V prípade nevalidných vstupných argumentov sa ukončí program s chybovým kód **ERR_ARGS**, ktorý je zadefinovaný ako 1.

3.2 Odchytávanie paketov

Implementované v súboroch sniffer.c a sniffer.h. K odchytávaniu paketov sa využíva knižnica pcap.h. Z hlavnej funkcie start_sniffing() sa volajú podfunkcie na nastavenie filtra, otvorenie rozhrania na odchytávanie packetov a odchytávací cyklus pcap_loop(). Funkciu loop_callback() volá cyklus vždy keď zachytí packet. Táto funkcia packet spracuje a vypíše jeho údaje na štandardný výstup. Preklad IP adries na hostname je uskutočnení pomocou funkcíi getaddrinfo() a getnameinfo(). V prípade, že sa nepodarí preložiť, je vypísaná IP adresa. Ak je nastavený filter na UDP packety, tak sa preklad IP adries zacyklý, pretože DNS využíva UDP. Pri každom zachytenom packete sa vytvorí ďalší na preloženie adresy. Odchytávanie IPv6 packetov nie je podporované, keďže sme to nemali ako testovať. Nepodarilo sa nám na našej domácej sieti na žiadnom počítači (testované boli 3) prijmať či odosielať IPv6 packety. Či už s využitím tcpdump, Wireshark alebo online testerov.

4 Testovanie

Program bol testovaný za pomoci nástroja **Wireshark**. Obom boli zadané rovnaké filtre a boli spustené. Zachytené packety v oboch programoch boli navzájom porovnané, či sa zhodujú.



Reference

- [1] "IPv4," April 30, 2020. https://en.wikipedia.org/wiki/IPv4.
- [2] "User Datagram Protocol." Wikipedia. Wikimedia Foundation, April 23, 2020. https://en.wikipedia.org/wiki/User_Datagram_Protocol.
- [3] Mitchell, Bradley. "An Inside Look at TCP Headers and UDP Headers." Lifewire. Lifewire, April 1, 2020. https://www.lifewire.com/tcp-headers-and-udp-headers-explained-817970.
- [4] "Transmission Control Protocol." Wikipedia. Wikimedia Foundation, April 29, 2020. https://en.wikipedia.org/wiki/Transmission_Control_Protocol.
- [5] Tcpdump Group. "Tcpdump/Libpcap Public Repository." TCPDUMP/LIBPCAP public repository. The Tcpdump Group. Accessed May 2, 2020. https://ww
- [6] NanoDano. "Using Libpcap in C." DevDungeon, October 28, 2018. https://www.devdungeon.com/content/using-libpcap-c?fbclid=IwAR2NP-LDruBjX-Hwa1To67uX6_4KmmxAxWzbT_6TIpOkG93p2bR3o4A9-XU.