VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Analyzátor síťového provozu

Projekt do předmětu Síťové aplikace a správa sítí (ISA)

Listopad 2016 Klára Nečasová

Obsah

1 Úvod do problematiky

Cílem projektu je vytvořit konzolovou aplikaci pro analýzu síťového provozu, který je zachycen v souboru formátu libpcap. Úkolem aplikace je analyzovat tento soubor a zjišťovat počet přenesených bajtů na základě definovaných položek. Přípustné položky jsou: MAC adresa, IPv4 adresa, IPv6 adresa, TCP port, UDP port. Pro počítání bajtů dle zmíněných položek je nutné specifikovat směr komunikace, tedy: zdroj, cíl nebo kombinace obojího. Také je nutné, aby aplikace dokázala vygenerovat seznam deseti paketů s největším počtem bajtů (seřazeno od největšího po nejmenší).

Aplikace vypisuje na standardní výstup výsledky ve tvaru `hodnota1_hodnota2` (znak podtržítko označuje mezeru), kde `hodnota1` značí součet bajtů od druhé vrstvy (tedy hlavička L2 + hlavička L3 + hlavička L4 + samotná data) a `hodnota2` udává součet bajtů od konce hlavičky dané vrstvy (pokud bude zadán filtr `ipv4`, `ipv6`, data se budou počítat od konce L3 hlavičky, v případě filtru `tcp`, `udp` se budou data počítat od konce L4 hlavičky). Pokud byl zadán filtr `top10`, bude výstup ve formátu `adresa_hodnota1_hodnota2`.

Formát pcap (z anglického *packet capture*) se používá k zachycení síťového provozu ve formě celých paketů, soubory v tomto formátu mají příponu `.pcap`. S tímto formátem lze pracovat pomocí různých API (aplikačně programových rozhraní). Na unixových systémech toto rozhraní implementuje knihovna `libpcap`, na systémech Windows se můžeme setkat s knihovnou `WinPcap`. Programy Wireshark či tcpdump běžně pracují právě se soubory v pcap formátu a umožňují je analyzovat a filtrovat dle různých kritérií.

2 Návrh aplikace

Pro implementaci byl zvolen jazyk C++ a objektově orientovaný přístup. Nejdříve bylo nutné podrobně nastudovat formát pcap souboru [5].

Obrázek 1: Formát pcap souboru

Na začátku se nachází globální hlavička (*Global Header*), která má pevně danou velikost 24 bajtů a je vždy na začátku přeskočena. Poté již následují zachycené pakety, každý paket je rozdělen na hlavičku (*Packet Header*) a samotná data (*Packet Data*). Hlavička paketu má velikost 16 bajtů, mezi nejdůležitější položky patří položka `orig_len`, která udává délku dat v daném paketu.

Daný paket je analyzován od linkové vrstvy až po transportní vrstvu. Mezi podporované protokoly patří následující.

- Linková vrstva (*Link Layer*)
  - Ethernet (a podpora VLAN hlaviček – IEEE 802.1Q) [7]
- Síťová vrstva (*Network Layer*)
  - IPv4 [8, 4]
  - IPv6 [9, 1]
  - ARP [6]
- Transportní vrstva (*Transport Layer*)
  - UDP [11, 3]
  - TCP [10, 2]

Jádro aplikace bude tvořit funkce, která dokáže správně zpracovat soubor ve formátu pcap a parsovat pakety podporovaných protokolů.

3 Popis implementace

O zpracování argumentů příkazové řádky se stará třída `ProcessParams`, která mimo jiné zajišťuje i validaci a normalizaci MAC adresy, IPv4 adresy a IPv6 adresy. Třída `Converter` zajišťuje různé potřebné převody mezi datovými typy. Jádro aplikace tvoří třída `PcapReader`, která obsahuje funkci pro načtení souboru ve formátu pcap, funkce pro zpracování paketů na dané vrstvě (linková vrstva, síťová vrstva, transportní vrstva) a také funkce zastřešující výpočet a výpis požadovaných statistik o počtu přenesených bajtů.

Důležitou funkcí je funkce `readPcapFile()`, která načítá soubor ve formátu pcap, globální hlavička je přeskočena. Jednotlivé hlavičky protokolů se zpracovávají směrem od nejnižší vrstvy k vyšší. Funkce pro zpracovávání paketů jsou navrženy tak, že vždy vrací informaci o tom, který protokol následuje na vyšší vrstvě (kromě funkce `processTransportLayer()`, protože data nad transportní vrstvou již neanalyzujeme). Nejdříve je zpracována linková vrstva (funkce `processLinkLayer()`). V této funkci je mimo jiné potřeba ošetřit, zda ethernetový rámec neobsahuje single/double IEEE 802.1Q tag, který do hlavičky ethernetového rámce vkládá 32/64bitovou položku definující virtuální síť (VLAN). Díky tomu lze rozdělit fyzickou síť na více logických sítí.

O zpracování paketů na síťové vrstvě se stará funkce `processNetworkLayer()`, která se volá v případě, že se jedná o protokoly IPv4, IPv6 či ARP (Address Resolution Protocol). Na této vrstvě je také nutné řešit padding u protokolu IPv4 a ARP. V případě, že je délka paketu rovna 60 bajtům (minimální velikost rámce) a zároveň je součet délky ethernetové hlavičky (bez paddingu) a délky dat od konce L2 hlavičky menší než 60 bajtů, potom je hodnota paddingu vypočtena jako rozdíl minimální velikosti rámce a délky dat.

Funkce `processTransportLayer()` zaštiťuje zpracování paketů na transportní vrstvě a volá se v případě, že se jedná o protokoly TCP nebo UDP.

Objem přenesených dat počítají funkce `editStatistics()`, `editStatisticsValue()` a `editStatisticsTop10()`. Ve funkci `editStatistics()` se zajistí, aby se filtr aplikoval pouze na adresy dané hodnotou filtru (např. pouze na MAC adresu 00:00:00:00:00:05). V případě, že byla zadána hodnota filtru `top10`, zisk příslušných statistik zajišťuje funkce `editStatisticsTop10()`, v opačném případě funkce `editStatisticsValue()`. O výpis výsledků ve správném formátu se stará funkce `printStatistics()`.

4 Návod na použití programu

Aplikace se spouští z příkazového řádku.
`./analyzer [-i soubor] [ -f typFiltru ] [ -v hodnotaFiltru ] [ -s ] [ -d ]`
Význam jednotlivých přepínačů je uveden níže.

- `-i soubor` (povinný parametr) vstupní soubor ve formátu libpcap
- `-f typFiltru` (povinný parametr) určuje, dle které položky se počítá objem dat. Přípustné hodnoty jsou: `mac`, `ipv4`, `ipv6`, `tcp`, `udp`
- `-v hodnotaFiltru` (povinný parametr) udává hodnotu filtru. Hodnotou filtru může být i `top10`, popis viz sekce 1.
- `-s` filtr se aplikuje na zdrojové adresy
- `-d` filtr se aplikuje na cílové adresy
- Poznámka: minimálně jeden z parametrů `-s` či `-d` musí být zadán.

Pro překlad aplikace lze využít přiložený soubor `Makefile`. Překlad se provede pomocí příkazu `make`. Odstranění vygenerovaných souborů zajistí `make clean`, potřebné soubory je možné komprimovat pomocí `make tar`.

4.1 Příklady spuštění programu

`./analyzer -i file0.pcap -f tcp -v 105 -s`
Program vyfiltruje veškerou komunikaci kromě TCP s hodnotou portu 105, který je zdrojovým portem. Poté je vypsána statistika daného portu.

`./analyzer -i file1.pcap -f udp -v 100 -d`
Program vyfiltruje veškerou komunikaci kromě UDP s hodnotou portu 100, který je cílovým portem. Poté je vypsána statistika zadaného portu.

`./analyzer -i file2.pcap -f ipv6 -v fd00::6 -s -d`
Program vyfiltruje veškerou komunikaci probíhající na IPv6 vrstvě. Poté je vypsána statistika pro zadanou IPv6 adresu, přičemž filtr je aplikován jak na zdrojové adresy, tak na cílové.

`./analyzer -i file3.pcap -f ipv4 -v top10 -s -d`
Program analyzuje veškerou komunikaci probíhající na IPv4 vrstvě. Poté je vypsána statistika o deseti záznamech s nejvyšším počtem přenesených bajtů, přičemž filtr je aplikován jak na cílové adresy, tak na zdrojové.

`./analyzer -i file4.pcap -f mac -v 00:00:00:00:00:04 -s`
Program analyzuje veškerou komunikaci probíhající na MAC vrstvě, zdrojová adresa musí odpovídat zadané hodnotě 00:00:00:00:00:04. Poté je vypsána statistika pro zadanou hodnotu filtru.

Literatura

[1] Deering, S.; Hiden, R.: Internet Protocol, Version 6 (IPv6) Specification. [online, cit. 7.11.2016].
URL <https://tools.ietf.org/html/rfc2460>

[2] Postel, J.: Transmision Control Protocol. [online, cit. 8.11.2016].
URL <https://www.ietf.org/rfc/rfc793.txt>

[3] Postel, J.: User Datagram Protocol. [online, cit. 6.11.2016].
URL <https://tools.ietf.org/html/rfc768>

[4] Postel, J.; aj.: RFC 791: Internet protocol. [online, cit. 5.11.2016].
URL <https://tools.ietf.org/html/rfc791>

[5] Wiki, W.: Libpcap File Format. 2015, [online, cit. 20.10.2016].
URL <https://wiki.wireshark.org/Development/LibpcapFileFormat>

[6] Wikipedia: Address Resolution Protocol. [online, cit. 1.11.2016].
URL <https://cs.wikipedia.org/wiki/Address_Resolution_Protocol>

[7] Wikipedia: Ethernet Frame. [online, cit. 23.10.2016].
URL <https://en.wikipedia.org/wiki/Ethernet_frame>

[8] Wikipedia: IPv4. [online, cit. 4.11.2016].
URL <https://en.wikipedia.org/wiki/IPv4>

[9] Wikipedia: IPv6. [online, cit. 5.11.2016].
URL <https://en.wikipedia.org/wiki/IPv6_packet>

[10] Wikipedia: Transmision Control Protocol. [online, cit. 8.11.2016].
URL <https://en.wikipedia.org/wiki/IPv6_packet>

[11] Wikipedia: User Datagram Protocol. [online, cit. 6.11.2016].
URL <https://en.wikipedia.org/wiki/User_Datagram_Protocol>