Vysoké učení technické v Brně
Fakulta informačních technologií

Počítačové komunikace a sítě – 2. projekt

Packet sniffer

22. dubna 2020

Jakub Sekula

1 Packet sniffer

Packet sniffer je buďto program nebo hardwarové zařízení sloužící k zachycení, logování a analýze síťového provozu a dat. Tyto zařízení pomáhají k identifikaci, vyhodnocení a hledání závady na síťovém provozu podle typu aplikace, zdroje a cíle [1] . Můj packet sniffer umí zachytávat síťový provoz na konkrétním zařízení.

2 Zpracování argumentů

Program načítá argumenty pomocí funkce `getopt_long()`. Funkce `getopt` nešla použít, protože argumenty -t a -u mohou být i ve svých delších podobách, tedy –tcp a –udp.

3 Zachycení paketu

3.1 Příprava otevření sítě

Z argumentu programu -i si pokusím zjistit IPV4 číslo sítě a masku sítě. Toto je realizováno pomocí funkce `pcap_lookupnet`. V případě neúspěchu program vypíše chybu na stderr a nastaví masku a číslo sítě na 0.

3.2 Otevření rozhraní pro záchyt paketů

Otevření rozhraní zadaného parametrem -i, je realizované pomocí funkce `pcap_open_live`. Jestliže funkce proběhne správně, vrací prostředek na manipulaci s packetem, takzvaný handle. V případě, že funkce nevrátí handler, vypíši chybu na stderr a program končí s návratovou hodnotou 2.

3.3 Filtrování paketů

Program umožnuje filtraci jak rozhraní, tak sítových protokolů **tcp** a **udp**. Aby bylo možné uplatnit filtry, existuje dvojice funkcí `pcap_compile a pcap_setfilter` první funkce zkompiluje filtrový výraz. Druhá funkce nastaví filtr na handler paketu.

3.4 Samotné zachycení paketu

Zachytit paket lze dvěmi způsoby. První z nich je použít funkci `pcap_next`. Druhá možnost je použít funkci `pcap_loop`. Rozdíl mezi těmito funkcemi je, že funkce `pcap_loop` dokáže zavolat zpracovávací funkci vícekrát. Kolikrát se zavolá je stanoveno v jejím volání.

[1] Zdroj: https://www.dnsstuff.com/packet-sniffers

4 Parsování paketu

4.1 Získání protokolu

Díky funkci `pcap_loop` zpracováváme paket v takzvané callback funkci, v případě mého programu se tato funkce jmenuje `packet_handle`. Tato funkce získá z hlavičky paketu jeho velikost a následně, i pomocí přetypování, druh protokolu. Jak již bylo zmíněno, program umí pracovat s protokoly **tcp** a **udp**. Následně se dostáváme do switche, kde se podle protokolu volá funkce `printFirtLine`.

4.2 Výpis prvního řádku

Na prvím řádku se má vypsat čas, doménové jméno nebo ip adresa, port a to samé krom času i pro koncovou stranu. Fukce si zjistí z ip adres, získaných z paketu doménové jméno zdroje a koncového bodu. Doménové jméno získávám pomocí funkce `getnameinfo`, pokud funkce uspěje vrací v parametru název domény a 0. Za předpokladu, že se vrátí 0, tak na první řádek vypisuji doménové jméno, jinak vypíši ip adresu. Z paketu dále získávám časový záznam, který zkonvertuju na strukturu tm a pak z ní vypíši hodnoty. A z paketu ještě získávám porty.

4.3 Výpis data z paketu

Funkce PrintData vypíše veškerý obsah packetu. Funkci předám paket a jeho velikost. Ve funkci pak vypisuji pomocí while cyklu celý obsah paketu a inkrementuju počítadlo po 1, až dokud nenarazím na velikost paketu.

Na každý řádek se vleze 16 znaků z paketu. Na začátku řádku je ještě přídán hexadecimální počítadlo řádků. Zajímavostí je řešení výpisu znaků, které by se zobrazovaly jako netisknutelné znaky, při jejich výpisu. Řešením je kontrolovat ascii hodnoty znaků a pokud nejsou v rozmezí 32 až 126, tak se místo znaku tiskne znak tečka a nevytiskne se na výstup žádný netisknutelný znak. Tento proces se následně opakuje pro zvolený počet paketů.

5 Testování

Svůj program jsem testoval porovnáváním výstupů s programem Wireshark. V programu jsem si zvolil rozhraní, na kterém chci zkoumat provoz sítě, to samé rozhraní jsem dal jako vstupní argument svému programu, otevřel jsem prohlížeč, zastavil jsem program Wireshark a i svůj program a porovnával jsem jejich výstupy.

6 Literatura

Při práci na projektu jsem velké množství informací a i kódu získával z článku "Programming with pcap"[2]. Velkou oporou mi také byla kniha "Computer networking : a top-down approach featuring the internet"[1].

Reference

[1] James F Kurose. *Computer networking : a top-down approach featuring the internet*. computer networking. Pearson/Addison-Wesley, Boston, 3rd ed. edition, 2005.

[2] Guy Harris Tim Carstens. Programming with pcap. [online], 2002.