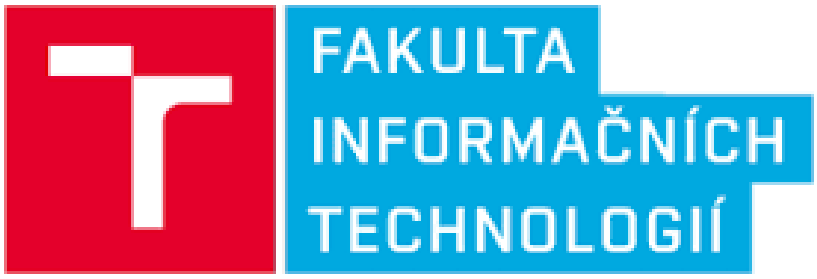
Vysoké učení technické v Brně

Fakulta informačních technologií



Počítačové komunikace a sítě

Dokumentace projektu č.2

Network Sniffer

Daniel Sehnoutek (xsehno02)

Brno, 13.4.2024

Obsah

Úvod

Vitajte v dokumentácii k projektu **Network Sniffer**, odvážny čitatelia. Tento projekt je zameraný na zachytávanie a filtrovanie paketov na určitom rozhraní.

Krátky súhrn teórie

Ethernet hlavička – poskytuje zdrojovú a cieľovú MAC adresu na prvých 12 bytoch, od 14 bytu sa môžu vyskytovať dáta alebo hlavička protokolu vyššej vrstvy.

IP hlavička – či už IPv4 alebo IPv6 poskytuje informácie o zdrojovej alebo cieľovej IP adrese. Ale taktiež určujú špecifickejšie protokoly. V tejto implementácií sa jednalo o ICMP, IGMP, NDP a MLD

Typy správ ARP – protokol ARP získava informácie o tom aká MAC adresa sa vzťahuje na akú IP adresu a funguje na druhé vrstve

TCP a UDP hlavička – obsahuje viacero informácií, môj sniffer sa predovšetkých sústredí na získanie čísla zdrojového a cieľového portu

Implementácia

Projekt je implementovaný v jazyku C.

Spracovanie argumentov

Keďže je veľa typov argumentov a rôzne argumenty môžu aj nemusia mať hodnotu rozhodol som sa spracovávať argumenty pomocou selekcie. Správne zadaný argument sa uloží do predpripravenej štruktúry `Setup`, v prípade že sa jedná o hodnotu uloží sa tam daná hodnota a zmení sa boolean hodnota príslušného argumentu v `bool FLAGS[ENUM_LEN],` ak sa jedná iba o zistenie či daný argument bol alebo nebol zadaný zmení sa iba boolean hodnota v poli. V prípade nesprávneho zadanie argumentov sa vypíše chybová hláška podľa situácie a korektne sa ukončí program.

Vytvorenie stringu na filtrovanie

Knižnice na konfiguráciu sieťového analyzátora očakávajú reťazec s informáciou o filtrovaní protokolov v určitom tvare, tvorbu tohto reťazca zabezpečuje funkcia `FilterStringCreating` pomocou už spomínaného poľa booleanov `setup.FLAGS[]`

Konfigurácia sieťového analyzátora

Prebieha pomocou knižnice `pcap` a jej funkcií. Za zmienku stojí hlavne funkcia `pcap_open_live` na ustanovenie spojenia, `pcap_compile` a `pcap_setfilter` na správne spracovanie a nastavenie filtru so zadanými protokolmi

Zachytávanie paketov a ich výpis

Zachytávanie paketov prebieha pomocou `pcap_loop` funkcie, ktorá požadovala implementáciu `packet_handler` funkcie. V tejto funkcii som definoval potrebné premenné a štruktúry na získanie a následné rozdelenie informácií z hlavičiek paketov.

Tieto informácie som vypísal v požadovanom formáte na štandardný výstup.

Použitie

Na preklad programu slúži Makefile, jeho spustenie je možné prevedením príkazu `make` cez terminal

Spustenie program

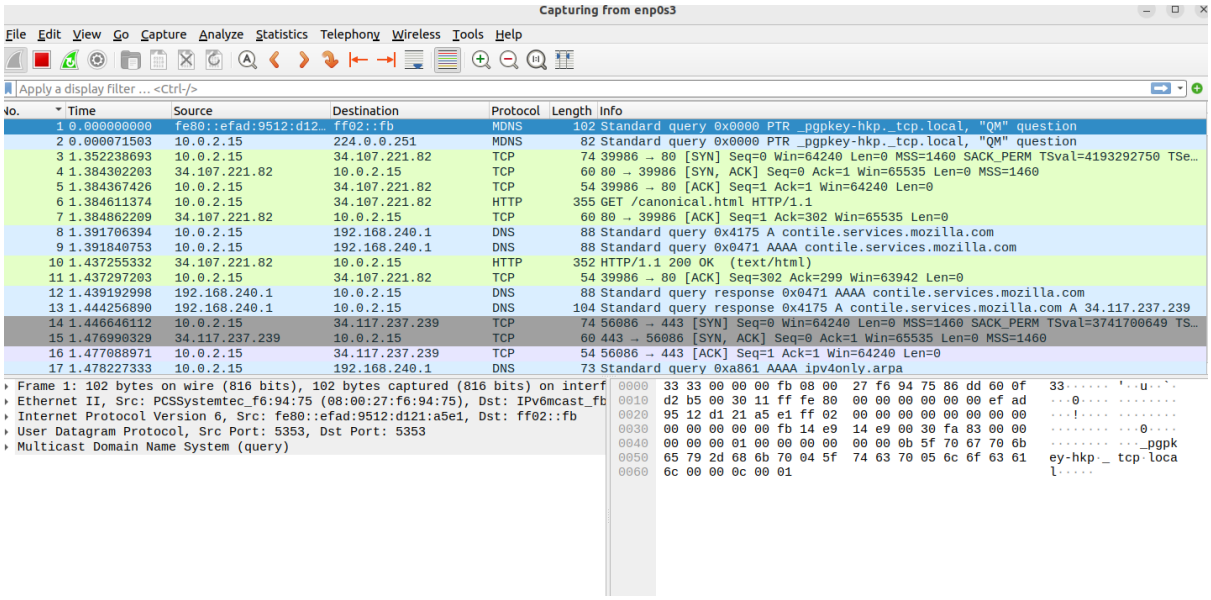
Execution

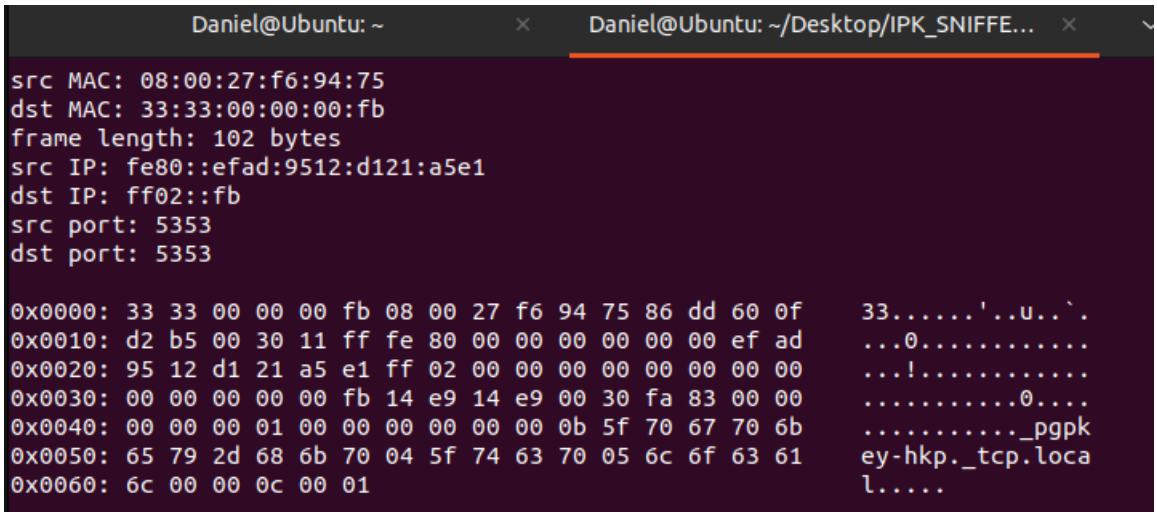
```
./ipk-sniffer [-i interface | --interface interface] {-p|--port-source|--port-destination port [--tcp|-t] [--udp|-u]} [--arp] [--icmp4] [--icmp6] [--igmp] [--mld] {-n num}
```

Program sa dá spustiť s veľa rôznymi kombináciami argumentov. Asi najdôležitejšie je si uvedomiť, že všetky protokoly sme schopný púšťať so všetkými a tým modifikovať filter vyhľadávania paketov. Základná hodnota počtu paketov na zobrazenie je jeden, toto číslo môžeme upraviť pomocou argumentu `-n`. Spustením ipk-snifferu bez parametrov alebo iba s parametrom `-i` alebo `-interface` bez názvu sieťového rozhrania nám vypíše dostupné sieťové rozhrania a kombinácia argumentu `-i` alebo `-interface` s inými argumentami je možná iba zadaním hodnoty sieťového rozhrania. Navyše argumenty je možné zadávať v ľubovoľnom poradí. Nesprávne zadanie argumentov je ošetrené chybovými hláškami a korektnou termináciou podľa danej situácie.

Testovanie

Správna funkčnosť programu bola testovaná pomocou software Wireshark a porovnávaniu výpisov z Wiresharku s mojimi výpismi. Ďalej všetky typy protokolov boli tiež otestované rôznymi spôsobmi ako sú napríklad pingy na nejaký vzdialený server a podobne.





Zdroje

IPK prezentácie

IPK prednášky

Článok o pcap knižnici

https://vichargrave.github.io/programming/develop-a-packet-sniffer-with-libpcap/

Skrátená dokumentácia pcap knižnice

https://www.tcpdump.org/pcap.html

EtherType wikipédia

https://en.wikipedia.org/wiki/EtherType