VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Sieťové aplikácie a správa sietí
2017/2018

Programovanie sieťovej služby

Export DNS informácií pomocou protokolu Syslog

Vladimír Marcin (xmarci10)

19. novembra 2018

Obsah

1 Úvod **2**

2 Problematika **2**
2.1 DNS 2
2.2 Ako funguje DNS 2
2.3 Formát DNS správy 4
2.3.1 Header 4
2.3.2 Question 5
2.3.3 Formát zdrojového (RR) záznamu 5
2.4 Typy DNS záznamov 6
2.5 Syslog 8

3 Popis vlastného riešenia **8**
3.1 Základný popis aplikácie 8
3.2 Popis implementácie 8
3.2.1 Spracovanie argumentov 8
3.2.2 Odchytávanie packetov 9
3.2.3 Extrahovanie užitočných dát z packetu 9
3.2.4 Spracovanie užitočných dát z paketu 10
3.2.5 Sprostredkovanie štatistík 10
3.3 Návod na použitie 11

1 Úvod

Cieľom projektu, ktorý popisuje táto dokumentácia, bolo vytvoriť jednoduchý sniffer, ktorý spracuváva dáta protokolu DNS a vybrané štatistiky exportuje pomocou protokolu Syslog na centrálny logovací server.

V kapitole 2 je stručne zhrnutá problematika, ktorou sa zaoberá naša aplikácia a v kapitole 3 bude detailnejšie popísaná samotná aplikácia.

2 Problematika

Skôr ako začneme popisovať konkrétne riešenie, vysvetlíme si niekoľko termínov potrebných pre pochopenie problematiky projektu.

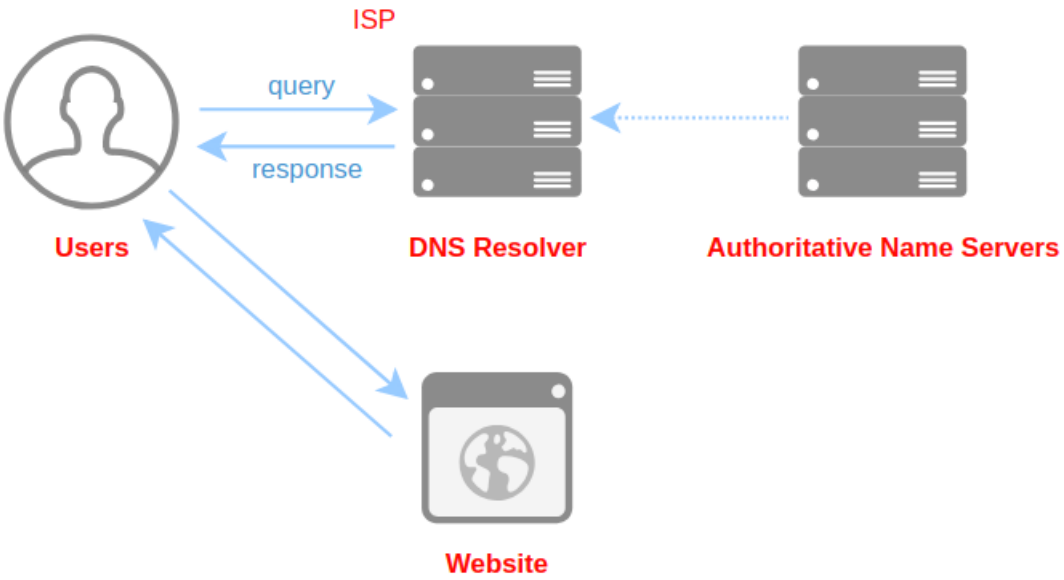
2.1 DNS

DNS (**D**omain **N**ame **S**ystem) je hierarchický systém doménových mien, ktorý je realizovaný DNS servermi a protokolom rovnakého mena, pomocou ktorého si vymieňajú informácie. Jeho hlavnou úlohou a príčinou vzniku sú vzájomné prevody doménových mien (ktoré sa ľuďom lepšie pamätajú) a IP adries (s ktorými pracujú stroje). Neskôr však pribral ďalšie funkcie a dnes slúži de facto ako distribuovaná databáza sieťových informácií [8].

Protokol používa porty TCP/53 i UDP/53 a je definovaný v RFC1035 [4].

2.2 Ako funguje DNS

DNS komunikácia prebieha v niekoľkých základných krokoch, ktoré sa vykonajú stále, keď sa objaví požiadavka pre DNS server o vyľadanie IP adresy webovej stránky.



Obr. 1: DNS komunikácia

1. krok: Žiadosť o webovú stránku

Proces začína v momente, keď užívateľ požiada počítač o preklad doménového mena na IP adresu (napr.: chce navštíviť stránku `https://www.fit.vutbr.cz/`). Následne sa hľadá IP adresa odpovedajúca doménovému menu v lokálnej DNS vyrovnávacej pamäti, ktorá uchováva nedávno používané informácie.

2. krok: Query na DNS Resolver

Ak požadované informácie niesú k dispozícii lokálne, počítač posiela DNS *query*[1] na DNS server (resolver) pridelený poskytovateľom internetovej služby. Ten vyrieši DNS query za nás.

Resolver má svojú vlastnú vyrovnávaciu pamäť a vzhľadom na to, že viacero klientov jedného ISP používajú rovnaký resolver je vysoká šanca, že populárne domény sa budú nachádzať vo vyrovnávacej pamäti. V takom prípade je vrátená DNS *response*[2] s požadovanými informáciami.

Pozn.: nie vždy musí byť používaný DNS resolver poskytovaný internetovým poskytovateľom služieb.

3. krok: Query na root servery

Ak sa odpoveď nenachádza ani v lokálnej pamäti DNS resolveru, ten posiela query na root servery. Tieto servery nepoznajú odpoveď na prijaté queries, ale vedia tieto queries nasmerovať na servery, ktoré to vedia.

4. krok: Query na TLD server

Root servery skúmajú prvú časť požiadavku, čítajúc sprava doľava (`https://www.fit.vutbr.cz/`) a v našom prípade by bola query smerovaná na top-level domain (TLD) name server **.cz**.

5. krok: Query na authoritative name servers

TLD server sa pozrie na ďalšiu časť nášho požiadavku (`https://www.fit.vutbr.cz/`) a pošle našu query na server zodpovedný za túto doménu.

Tieto server sa nazývajú authoritative name servers a sú zodpovedné za držanie všetkých informácií o špecifikcej doméne. Tieto informácie sú uložené v podobe DNS záznamov, ktorých existuje viacero druhov a každý obsahuje iné informácie (Resource Records (RR)[3] záznamy).

V našom príklade, chceme vedieť IP adresu pre `https://www.fit.vutbr.cz/`, takže požiadame authoritative name server o záznam obsahujúci adresu pre toto doménové meno (záznam typu A).

6. krok: Prijatie záznamu resolverom

Resolver potom príjme A záznam pre `fit.vutbr.cz/` od authoritative name servera a uloží si ho do lokálnej vyrovnávacej pamäte. Ak hocikto iný bude tento záznam potrebovať, resolver mu ho poskytne bez nutonosti znovu vykonávať celý proces zisťovania odpovede.

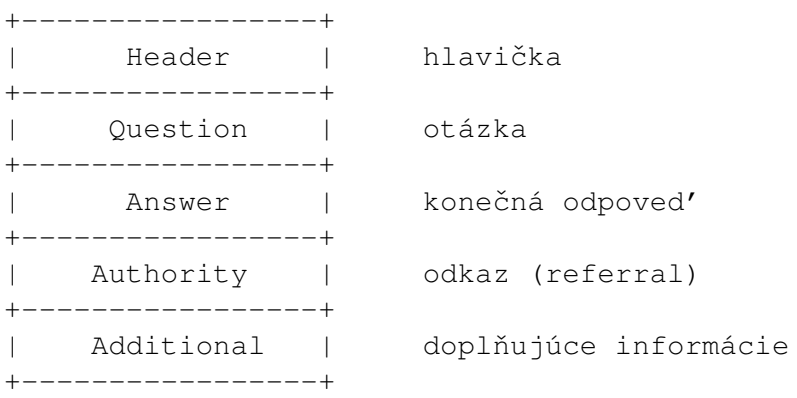
7. krok: Prijatie odpovede (response)

Resolver pošle response obsahujúcu A záznam nášmu počítaču. Ten si ju uloží v svojej DNS vyrovnávacej pamäti, prečíta hľadanú IP adresu a predá ju prehliadaču. Prehliadač následne nadviaže spojenie s webserverom a zobrazí stránku [3].

[1] http://social.dnsmadeeasy.com/blog/the-mechanics-behind-the-internet-what-is-a-dns-query/
[2] https://blog.dnsimple.com/2015/03/whats-in-a-dns-response/
[3] http://www.zytrax.com/books/dns/ch8/

2.3 Formát DNS správy

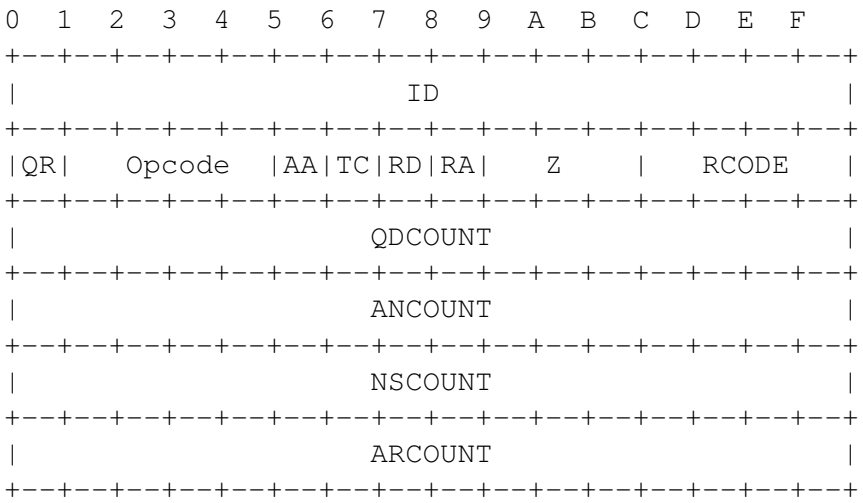
Všetky DNS správy majú nasledovný formát:



Podľa toho či sa jedná o query alebo response, sú vyplnené rozdielne časti správy.

2.3.1 Header

Hlavička má nasledujúci formát:



V tomto diagrame, každá bunka reprezentuje jeden bit. V každom riadku je 16 stĺpcov, reprezentujúcich dva bajty dát. Diagram je rozdelený do riadkov pre lepšiu čitateľnosť, no v skutočnosti sa jedná o sériu bajtov nasledujúcich bezprostredne za sebou.

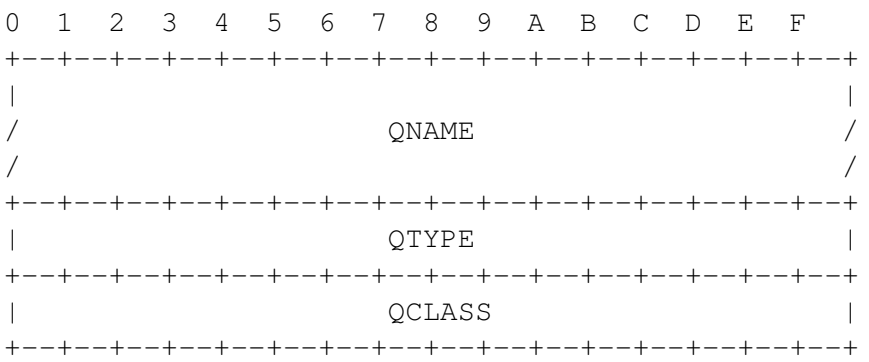
Keďže queries aj responses zdieľajú spoločný formát hlavičky, nie všetky položky sú relevantné pre obe druhy správ (nepoužité bunky sú naplnené nulami). Podrobný popis všetkých položiek je možné nájsť v RFC1035 [4].

Položky pre nás zaujimáve sú:

- `QR`: flag, ktorý hovorí o tom, či sa jedná o query (0) alebo response (1)
- `RCODE`: 4 bitová položka, ktorá je nastavená ako súčasť DNS response správy. Hodnota 0 v tomto poli znamená, že nedošlo k žiadnej chybe pri spracovávaní query. Ostatné možné hodnoty viz RFC1035 [4].
- `QDCOUNT`: 16 bitový unsigned integer udávajúci počet položiek v sekcii otázok
- `ANCOUNT`: 16 bitový unsigned integer udávajúci počet RR záznamov v sekcii odpovedí

2.3.2 Question

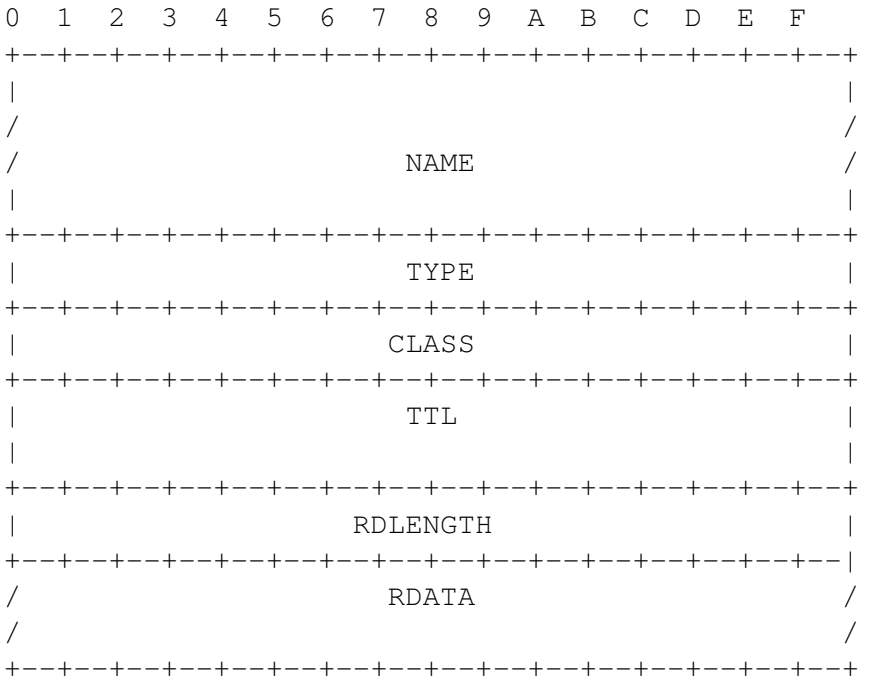
Qestion sekcia v DNS správe ma nasledujúci formát:



- `QNAME`: Doménové meno reprezentované ako sekvencia labelov, kde každý z nich je reprezentovaný jeho dĺžkov 1 oktet a za ním nasleduje počet oktetov, ktorý udala táto dĺžka. Koniec doménového mena je reprezentovaný nulovým oktetom.
- `QTYPE`: dva oktety reprezentujúce typ query správy.
- `QDCOUNT`: dva oktety reprezentujúce triedu query správy.

2.3.3 Formát zdrojového (RR) záznamu

Všetky RR záznamy zdieľajú formát, ktorý je zobrazený na obrázku nižšie. Answer, authority a additional sekcie DNS správy potom obsahujú **n** záznamov tohto formátu, pričom **n** je hodnota, ktorá je uvedená v DNS hlavičke.



- `NAME`: meno domény, ktorej sa týka RR záznam
- `TYPE`: dva oktety určujúce RR type, čo znamená, že určuje význam dát v RDATA
- `CLASS`: dva oktety určujúce triedu dát v RDATA
- `TTL`: 32 bitový unsigned integer, určujúci časový interval, po ktorý môže byť resource record uložený vo vyrovnávacej pamäti
- `RDLENGTH`: 16 bitový unsigned integer, ktorý určuje dĺžku RDATA udávanú v oktetoch
- `RDATA`: položka premennej dĺžky. Formát závisí od položiek TYPE a CLASS. Napríklad ak TYPE=A a CLASS=IN, potom RDATA pozostáva zo 4 oktetov ARPA Internetovej adresy.

2.4 Typy DNS záznamov

DNS záznam domény obsahuje viacero typov záznamov, z ktorých každý má inú funkciu. Najčastejšie používané typy, ktoré podporuje aj naša aplikácia budú stručne popísané v tejto podkapitole.

- **A**[RFC1035]: (**a**dress record) obsahuje IPv4 adresu priradenú danému menu, napríklad keď menu foo.bar.cz prislúcha IP adresa 1.2.3.4, bude zdrojový záznam vyzerať následovne:

  ```
  foo IN A 1.2.3.4
  ```

- **AAAA**[RFC1886]: (IPv6 address record) obsahuje IPv6 adresu. Stroju z prvého príkladu by sme IPv6 adresu 2001:718:1c01:1:02e0:7dff:fe96:daa8 priradili záznamom:

  ```
  foo IN AAAA 2001:718:1c01:1:02e0:7dff:fe96:daa8
  ```

- **CNAME**[RFC1035]: (**c**anonical **name** record) je alias (iné meno pre už zavedené). Typicky sa používa pre servery známych služieb, ako napríklad WWW. Jeho definícia pomocou prezývky ho umožňuje neskôr ľahko presťahovať na iný počítač. Ak náš foo.bar.cz má slúžiť zároveň ako www.bar.cz, zdrojový záznam bude vyzerať:

  ```
  www IN CNAME foo
  ```

- **MX**[RFC1035]: (**mail** e**x**change record) oznamuje adresu a prioritu serveru pre príjem elektronickej pošty pre danú doménu. Tentokrát sú parametre dva - priorita (prirodzené číslo, menšie znamená vyššiu prioritu) a doménové meno serveru. Ak poštu pre počítač foo.bar.cz príjma najlepšie počítač mail.bar.cz a prípadne ako záložný aj mail.baz.cz, bude zónový súbor[4] obsahovať záznamy:

  ```
  foo IN MX 10 mail
  IN MX 20 mail.baz.cz.
  ```

- **NS**[RFC1035]: (**n**ame **s**erver record) ohlasuje meno autoritatívneho DNS serveru pre danú doménu. Ak bude mať doména bar.cz poddoménu sth.bar.cz, ktorej servery budú ns.bar.cz (primárny) a ns.baz.cz (sekundárny), bude zónový súbor pre bar.cz obsahovať:

  ```
  sth IN NS ns
  IN NS ns.baz.cz.
  ```

[4] http://www.zytrax.com/books/dns/ch6/mydomain.html

- **PTR**[RFC1035]**:** (**p**oin**t**e**r**) je špeciálny typ záznamu pre reverzné zóny. Obsahuje na pravej strane meno počítača pridelené adrese na strane ľavej.

```
4 IN PTR foo.bar.cz.
```

- **SOA**[RFC1035]**:** (**s**tart **o**f **a**uthority record) je zahajujúci záznam zónového súboru. Obsahuje meno primárného serveru, adresu elektronickej pošty jeho správcu a nasledujúce údaje:
  - *Serial* – sériové číslo, ktoré je potrebné zvýšiť s každou zmenou v zázname. Podľa neho sekundárný server pozná, že v doméne došlo k zmene. Pokiaľ ho zabudnete zväčšiť, rozíde sa obsah sekundárnych serverov s primárnym. Pre prehľadnosť často vo formáte YYYYMMDDHH.
  - *Refresh* – ako často sa má sekundárny server pýtať na novú verziu zóny (v sekundách)
  - *Retry* – v akých intervaloch má sekundárny server opakovať svoje pokusy, pokiaľ sa mu nepodarí spojiť s primárnym
  - *Expire* – čas po ktorom sekundárny server označí svoje záznamy ako neaktuálne, ak sa mu nepodarí kontaktovať primárny server
  - *TTL* – implicitná doba platnosti záznamu

  Časové údaje su v sekundách. Novšie implementácie umožňujú pre väčšie pohodli používať prípony 'm', 'h', 'd', 'w' (minute, hodina, deň, týždeň) [8].

```
@ IN SOA ns.bar.cz. spravca.bar.cz. (
200605140
1h
5m
1w
1d
)
```

- **TXT**[RFC1035]**:** tento záznam je možné využiť pre zapísanie ľubovoľného textového reťazca do DNS záznamu domény. Možné použitie je napríklad overenie vlastníka domény, kedy Vás poskytovateľ hostingových či iných služieb požiada o vloženie TXT záznamu s určitým textom do DNS [6].

- **SPF**[RFC7208]**:** (**S**ender **P**olicy **F**ramework) je špeciálny záznam, s ktorého pomocou môžme nadefinovať, ktoré SMTP servery sú pre danú doménu autorizované pri odosielaní e-mailov [7].

Ďalšou skupinou záznamov, ktoré podporuje naša aplikácie sú záznamy DNSSEC[5]. V skratke je DNSSEC techonológia, ktorá chráni domény proti presmerovaniu a zaručuje, že obsah je autentický. Každá doména je totiž podpísaná, a pokiaľ by sa niekto chcel pripojiť k DNS serveru bez znalosti kľúča, DNS server to zamietne [1].

- **DNSKEY**[RFC4034]**:** obsahuje verejný kľúč, ktorého odpovedajúcim prívátnym kľúčom sú podpísané DNS záznamy tejto domény

- **RRSIG**[RFC4034]**:** (**R**esource **R**ecord **Sig**nature) obsahuje digitálny podpis príslušnej množiny DNS záznamov

- **DS**[RFC4034]**:** (**D**elegation **S**igner) je umiestnený v nadradenej DNS doméne a obsahuje otlačok verejného kľúča uloženého v DNSKEY zázname podpísanej domény. Pomocou DS záznamov sa vytvára reťazec dôvery do nadradených domén.

- **NSEC**[RFC4034]**:** (**N**ext **Sec**ure) využíva sa pre informáciu o nexxistencii žiadaného záznamu [5]

[5] https://www.nic.cz/page/513/about-dnssec/

2.5 Syslog

Syslog je štandard pre záznam programových správ. Umožňuje oddeliť:

- software generujúci správy
- od systému, ktorý ich ukladá
- a softwaru, ktorý poskytuje reporty a analýzy

Syslog môže slúžiť systémovému manažmentu a bezpečnostnému auditu ako zdroj infromácií pre analýzu alebo ladenie systému. Logovacie správy môžu byť uložené buď lokálne v logovacích súboroch alebo môžu byť posielané na nejaký vzdialený cieľ.

Syslog je protokol typu klient/server: logovacia aplikácia pošle textovú správu na syslog príjimač. Syslog správy môžu byť posielané cez UDP alebo TCP protokol. Syslog používa číslo portu 514.

Formát syslog správy je definovaný v RFC5424 [2].

3 Popis vlastného riešenia

Táto kapitola obsahuje popis môjho riešenia pozostávajúceho so stručného popisu aplikácie, následne popisu samotnej implementácie a v závere sa nachádza stručný návod na použitie.

3.1 Základný popis aplikácie

Aplikácia je napísaná v jazyku C/C++ a je rozdelená do niekoľkých modulov:

- `dns-sniffer.cc`: modul obsahujúci main, ošetrenie argumentov a odchytávanie komunikácie
- `base64.{cc,h}`: kódovanie a dekodóvanie BASE64 kódu.
- `dissect.{cc,h}`: extrahovanie užitočných dát z DNS packetu
- `dns-stats.{cc,h}`: spracovanie DNS dát a aktualizovanie štatistík
- `syslog.{cc,h}`: posielanie štatistík na syslog server

Aplikácia pracuje v dvoch režimoch:

- *online*: v tomto režime aplikácia zachytáva DNS komunikácia na niektorom z aktívnych sieťových rozhraní stroja na ktorom beží
- *offline*: aplikácia spracováva DNS komunikáciu, ktorá sa nachádza v predanom pcap súbore

Výstupom aplikácie sú spracované DNS štatistiky, ktoré sú buď vypísané na štandardný výstup, alebo odoslané na centrálny logovací server ak je zadaný.

Pre korektné ukončenie aplikácie, bežiacej v online móde je potrebné ju ukončiť sekvenciou `CONTROL-C`. Obsluha signálu `SIGINT` uvoľní aplikáciou alokovanú pamäť a získané zdroje.

3.2 Popis implementácie

V tejto podkapitole si popíšeme konkrétne jednotlivé časti implementácie s dôrazom na zaujímavejšie pasáže.

3.2.1 Spracovanie argumentov

Argumenty sú spracovávané vo funkcii `main`. Na spracovanie som použil `getopt()`, kvôli jednoduchému používaniu. Aby som ošetril prípustné kombinácie argumentov, vytvoril som jendoduchú bitovú mapu, ktorá to zabezpečuje.

3.2.2 Odchytávanie packetov

Na odchytávanie packetov používam knižnicu `libpcap`, ktorá sa stará o obe režimi (online i offline) a posiela aplikácii packety buď zachytené na sieťovom rozhraní, alebo prečítané zo súboru. Ak aplikácii nezadáme názov rozhrania, na ktorom ma počúvať alebo súbor, ktorý ma spracovávať, tak zachytáva komunikáciu na všetkých aktívnych rozhraniach. Rozšírenie aplikácie o túto funkcionalitu bolo jednoduché, keďže to knižnica `libpcap` defaultne podporuje (do funkcie pcap_open_live() zadáme parameter 'any').

Aplikácia zachytáva iba DNS komunikáciu čo je zabezpečené použitím filtru (načúvanie na porte 53). Ako optimalizáciu je tento filter nastavený tak, že do aplikácie sú na spracovanie posielané iba pakety, ktoré prišli z DNS serveru (`src port 53`), to znamená, DNS odpovede (responses).

Poškodené alebo nekompletné pakety sú zahodené bez výpisu chybového hlásenia.

3.2.3 Extrahovanie užitočných dát z packetu

Extrahovanie DNS dát z paketu prebieha tak, že sú postupne odstraňované hlavičky vyšších vrstiev:

- **Linková vrstva:** pomocou funkcie `pcap_datalink()` zistíme identifikátor protokolu na linkovej vrstve a podľa toho preskočíme príslušnú hlavičku. Aplikácia podporuje Ethernetové packety a taktiež Linux "cooked"pakety, ktoré sú vrátené v prípade načúvania na všetkých aktívnych rozhraniach.
- **Sieťová vrstva:** podľa hodnoty z linkovej hlavičky zistím či sa jedná o protokol IPv4 alebo IPv6 a preskočím hlavičku sieťovej vrstvy. Ak sa jedná o IPv6 preskočím aj prípadné rozširujúce hlavičky.
- **Transportná vrstva:** na základe hodnoty v IP hlavičke rozoznáme transportný protokol:
  - UDP – v tomto prípade preskočím transportnú hlavičku a v tomto bode mám k dispozícii ukazateľ na dáta aplikačnej vrstvy (DNS dáta) a ich veľkosť. Tieto údaje sú posielané modulu ktorý tieto dáta spracováva.
  - TCP – ak sa jedná o TCP protokol, môže na transportnej vrstve dôjsť k segmentácii a z toho nestačí iba preskočiť transportnú hlavičku. Pre tento účel som si vytvoril dátovú štruktúru, kde v prípade, že došlo k segemntácii ukladám všetky časti jednej TCP správy a táto správa je posunutá na spracovanie až v prípade, že dorazila celá.

```
struct tcp_dns_message{
    uint16_t read_dns_data;
    uint16_t dns_data_length;
    uint32_t next_seq_number;
    std::vector<std::pair <unsigned long, unsigned char *>>ptr_to_tcp;
};
```

Listing 1: Štruktúra TCP správy

V tejto štruktúre sa nachádza vektor `ptr_to_tcp` ukazateľov na jednotlivé časti jednej TCP správy, veľkosť danej správy `dns_data_length` (zistená z DNS hlavičky), veľkosť dát, ktoré sme doposiaľ dostali `read_dns_data` a číslo `next_seq_number`, ktoré udáva sekvenčné číslo ďalšej časti danej DNS správy v prípade, že ešte na nejakú čakáme.

V programe potom máme vektor týchto štruktúr. Po príchode TCP packetu zistíme jeho sekvenčné číslo z TCP hlavičky a prehľadávame vektor TCP správ, či náhodou tento segment nepatrí do niektorej už došlej sekvencie (porovnávame s `next_seq_number`). V prípade, že sa sekvenčné čislo zhoduje s `next_seq_number` daný segment sa vloží do sekvencie a vypočíta sa nové `next_seq_number`. Ak nie vytvorí sa nová sekvencia, inicializuje sa `dns_data_length` a `next_seq_number`.

V prípade, že `read_dns_data == dns_data_length` správa dorazila celá a môže byť spracovaná.

3.2.4 Spracovanie užitočných dát z paketu

Užitočné dáta paketu predstavuje DNS správa, ktorej štruktúra je popísaná v kapitole 2.3. Z týchto dát sa vytvárajú štatistiky vo formáte:

```
domain-name rr-type "rr-answer" count
/*Priklad*/
google.com A "172.217.23.238" 68
```

Tieto štatistiky sú uložené v mape, kde kľúčom je reťazec "`domain-name rr-type rr-answer`" a hodnotou je číslo `count`, ktoré reprezentuje počet koľko krát sa daný záznam objavil v skúmanej DNS komunikácii.

Aplikácia podporuje typy záznamov popísané v kapitole 2.4. Význam položiek v `rr-answer` pre jednotlivé typy záznamov si môžete dohľadať v príslušnom RFC (viz kapitola 2.4).

Formát `rr-answer` v štatistikách sa zhoduje s presentation formátom v jednotlivých RFC. Naša aplikácia má taktiež formát `rr-answer` zhodný s formátom utility `dig` (čo som využíval pri testovaní).

V prípade, že DNS odpoveď obsahuje záznam, ktorý nieje podporovaný našou aplikáciou formát štatistiky má nasledujúci formát:

```
domain-name TYPE{cislo typu} "{hexa_rdata}" count
/*Priklad*/
_sip._tcp.cesnet.cz TYPE33 "0064000A13C4056379727573066365736E657402637A00" 1
```

Používateľ si tak môže dohľadať o aký typ záznamu sa jedná a v prípade záujmu vo vlastnej réžii dekódovať dáta záznamu.

3.2.5 Sprostredkovanie štatistík

Ak je zadaný syslog server, štatistiky sa vždy odošlú na syslog server a to:

- v online režime periodicky po ubehnutí predom stanoveného času,
- alebo v offline režime po spracovaní celého pcap súboru.

Na výstup sa štatistiky vypíšu v prípade:

- že ide o offline režim a nebol zadaný syslog server
- že ide o online režim a aplikácii je doručený signál `SIGUSR1`

Sprostredkovanie štatistík či už jedným alebo druhým spôsobom má vždy na starosti detský proces. Pričom rodičovský proces pokračuje v spracovávaní štatistík. Sprostredkovanie potom vyzerá následovne:

```
    pid_t pid;

    /* ak fork zlyha (podmienka nebude platit) rodicovsky proces
     * sprostredkuje statistiky a az potom pokracuje v spracovavani
     * dalsich (statistiky sa poslu aj za cenu ze nezachytime
     * aktualne prebiehajucu komunikaciu)
     */
    if((pid = fork()) > 0) {
        /* iba pre online mod */
        start_new_period(seconds);
        return;
    }
    send_or_print_stats();

    if(pid == 0)
        exit(0);
```

Tento kód je súčasťou obsluhy signálu, ktorý sa posiela periodicky po uplinutí času počas ktorého sa majú spracovávať štatistiky (vtedy sa budú posielať), alebo signálu `SIGUSR1` (vtedy sa budú vypisovať).

3.3 Návod na použitie

Program má 4 nepovinné argumenty. Podporovaná je iba krátka forma argumentov. Ak chceme aby program načúval na sieťovom rozhraní musíme ho spustiť ako **root**.

Spustenie aplikácie:

```
./dns-export [-r file.pcap] [-i interface] [-s syslog-server] [-t seconds]

    -r : spracuje dany pcap subor
    -i : nacuva na danom sietovom rozhrani a spracovava DNS provoz,
         defaultna hodnota any
    -s : hostname/ipv4/ipv6 adresa syslog serveru
    -t : doba vypoctu statistik, defaultna hodnota 60s
```

Prípustné kombinácie argumentov:

```
./dns-export
```

Aplikácia načúva na všetkých aktívnych sieťových rozhraniach a v prípade, že obdrží signál `SIGUSR1` vypíše doposiaľ spracované štatistiky.

```
./dns-export -r file.pcap
```

Aplikácia po spracovaní súboru `file.pcap` vypíše štatistiky na `STDOUT`.

```
./dns-export -r file.pcap -s syslog-server
```

Aplikácia po spracovaní súboru `file.pcap` odošle štatistiky na `syslog-server`.

```
./dns-export -s syslog-server
```

Aplikácia načúva na všetkých aktívnych sieťových rozhraniach a každých 60s pošle štatistiky na `syslog-server`.

```
./dns-export -s syslog-server -t seconds
```

Aplikácia načúva na všetkých aktívnych sieťových rozhraniach a každých `seconds` sekúnd pošle štatistiky na `syslog-server`.

```
./dns-export -i interface
```

Aplikácia načúva na rozhraní `interface` a v prípade, že obdrží signál `SIGUSR1` vypíše doposiaľ spracované štatistiky.

```
./dns-export -i interface -s syslog-server
```

Aplikácia načúva na rozhraní `interface` a každých 60s pošle štatistiky na `syslog-server`.

```
./dns-export -i interface -s syslog-server -t seconds
```

Aplikácia načúva na rozhraní `interface` a každých `seconds` sekúnd pošle štatistiky na `syslog-server`.

Literatúra

[1] Arends, R.; Austein, R.; Larson, M.; aj.: Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005, updated by RFC 4470.
URL http://www.ietf.org/rfc/rfc4034.txt

[2] Gerhards, R.: The Syslog Protocol. RFC 5424 (Proposed Standard), March 2009.
URL http://www.ietf.org/rfc/rfc5424.txt

[3] Gonyea, C.: DNS: Why It's Important and How It Works. online.
URL https://dyn.com/blog/dns-why-its-important-how-it-works/

[4] Mockapetris, P.: *RFC 1035 Domain Names - Implementation and Specification*. Internet Engineering Task Force, November 1987.
URL http://tools.ietf.org/html/rfc1035

[5] Moučka, B.: DNSSEC. online.
URL http://webserver.ics.muni.cz/bulletin/articles/659.html

[6] Web4you: DNS - TXT záznam. online.
URL https://helpdesk.web4u.cz/index.php?/Knowledgebase/Article/View/276/9/dns---txt-zaznam

[7] Web4you: SPF záznam. online.
URL https://helpdesk.web4u.cz/index.php?/Knowledgebase/Article/View/157/9/spf-zaznam

[8] Wikipedia: Domain Name System. online.
URL https://cs.wikipedia.org/wiki/Domain_Name_System#Typy_z%C3%A1znam%C5%AF