
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**


# MONITOROVANIE DNS KOMUNIKÁCIE


**AUTOR PRÁCE** **ROMAN POLIAČIK**



**BRNO 2024**

# Obsah

**1 Úvod** **2**
1.1 Úvod do problematiky . . . 2
1.2 Teoretický základ . . . 3
1.2.1 Komunikácia DNS . . . 3
1.2.2 Štruktúra DNS správy . . . 4
1.2.3 Typy záznamov . . . 5
1.2.4 DNS Name Compression . . . 5

**2 Implementácia** **6**
2.1 Návrh . . . 6
2.2 Popis implementácie . . . 6
2.2.1 Všeobecný popis architektúry programu . . . 6
2.2.2 Všeobecný popis toku programu . . . 7
2.2.3 Podrobnejší popis zaujímavých častí a obmedzení . . . 7

**3 Návod na použitie** **9**

**4 Testovanie** **10**
4.1 skript test.sh . . . 10
4.1.1 Popis testovacieho skriptu . . . 10
4.1.2 Spustenie testu . . . 10
4.1.3 Obsah súboru `dns_queries.txt` . . . 10
4.1.4 Vyhodnotenie výsledkov . . . 11
4.2 Testovanie s PCAP súbormi . . . 11
4.2.1 PCAP súbor `alltypes.pcap` . . . 11
4.2.2 PCAP súbor `v6.pcap` . . . 11

**Literatúra** **12**

# Kapitola 1

# Úvod

## 1.1 Úvod do problematiky

Systém názvov domén(domain name system), skr. DNS, je kľúčovou súčasťou dnešného Internetu, ktorá prekladá názvy domén na adresy IP, ktoré sa používajú na vzájomnú komunikáciu medzi počítačmi. Monitorovanie DNS komunikácie je dôležitou súčasťou zachovania bezpečnosti a efektivity siete. Zabezpečuje správnu funkciu DNS serverov, čím predchádza výpadkom webových stránok alebo služieb, ku ktorým by mohlo dôjsť pri zlyhaní DNS dotazov. Monitorovaním DNS prevádzky môžu správcovia sietí rýchlo odhaliť a zmierniť bezpečnostné hrozby, ako sú DDoS útoky, DNS spoofing či cache poisoning. Zároveň poskytuje informácie o výkonnosti siete, pomáha identifikovať pomalé alebo neefektívne servery a umožňuje lepšie plánovanie budúcich kapacitných potrieb[11].

## 1.2 Teoretický základ

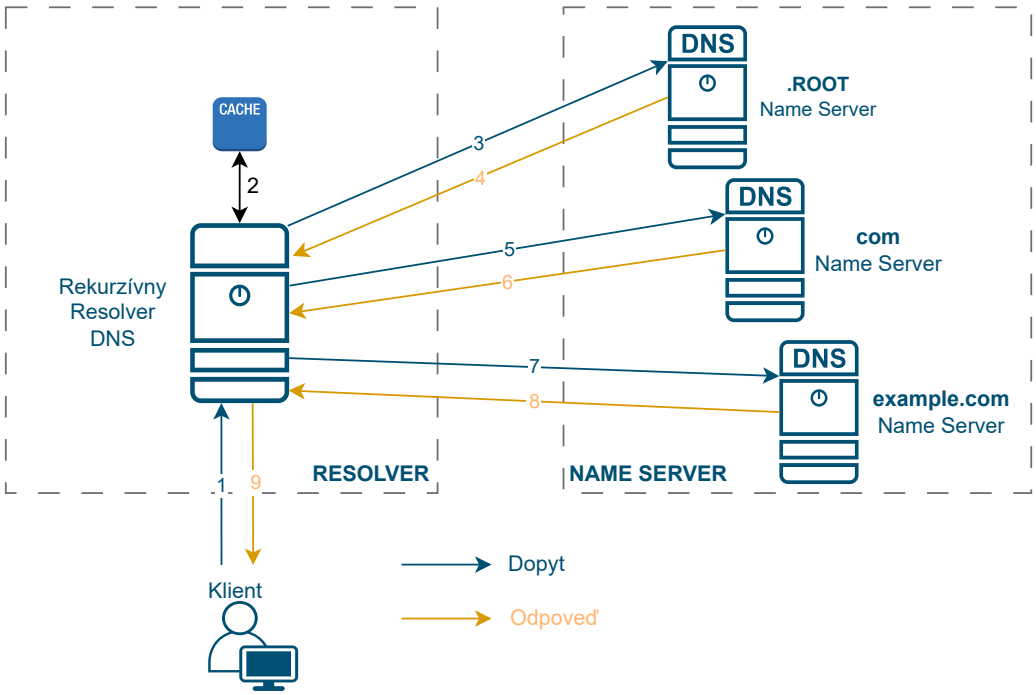
### 1.2.1 Komunikácia DNS

Pri DNS komunikácii sa zvyčajne stretávame s dvoma hlavnými úlohami: resolver a name server.

- **Resolver**: Posiela DNS dopyty, s cieľom preložiť názvy domén na IP adresy. Resolver je systémová rutina operačného systému, ktorá spracováva dotazy klienta (aplikácie) [3] [7]
- **Name Server**: Servery, ktoré obsahujú DNS záznamy a odpovedajú na dopyty od resolverov.

Zovšeobecnený proces rezolúcie DNS:

1. Klientská aplikácia požiada o IP adresu pre konkrétny doménový názov.
2. Resolver odošle DNS dopyt lokálnemu alebo rekurzívnemu DNS serveru.
3. Ak server nemá odpoveď, požiada o ňu vyššie úrovne DNS serverov, ktoré využívajú stromovú štruktúru názvov, začínajúc koreňovými DNS servermi.
4. Server získa IP adresu pre daný názov domény a odošle ju resolveru.
5. Resolver poskytne IP adresu klientskej aplikácii.



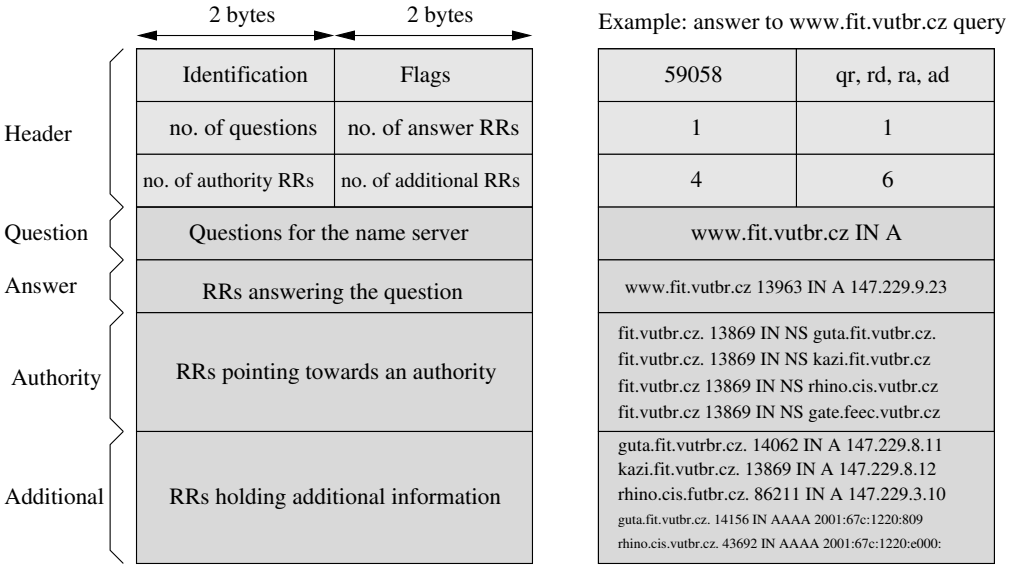

Obr. 1.1: Zjednodušený diagram DNS komunikácie[1]

[1]Preložený obrázok prevzatý z: https://www.researchgate.net/figure/DNS-architecture-DNS-Domain-Name-System-gTLD-general-Top-Level-Domain_fig2_345017736[21.10.2024].

### 1.2.2 Štruktúra DNS správy

DNS správa sa skladá z nasledujúcich častí:

- **Hlavička (Header)**: Obsahuje polia, ktoré určujú typ správy (QR), operačný kód (OPCODE), príznaky (Flags), návratový kód (RCODE), a počty rôznych sekcií záznamov (QDCount, ANCount, NSCount, ARCount).
  - **Flags** – príznaky o povahe DNS odpovede alebo požiadavky:
    - **AA** – odpoveď pochádza z autoritatívneho servera, a nie z cache.
    - **TC** – Správa bola skrátená kvôli limitu veľkosti.
    - **RD** – Označuje požiadavku na rekurzívne vyhľadávanie.
    - **RA** – Server podporuje rekurzívne vyhľadávanie.
    - **Z** – Rezervované pole (3 bity), v súčasnosti nemá žiadnu funkciu.
- **Sekcia Otázky (Question)**: Obsahuje detaily o tom, čo sa hľadá. Má tri časti:
  - **QNAME**: Doménové meno, o ktorom klient požaduje informácie.
  - **QTYPE**: Typ záznamu, ktorý sa vyžaduje (napr. A, AAAA, MX...).
  - **QCLASS**: Trieda záznamu, dnes zvyčajne `IN` pre internet.
- **Sekcia Odpoveď (Answer)**: Odpovede na otázku, obsahujú požadované záznamy.
- **Autorita (Authority)**: Informácie o autoritatívnych DNS serveroch pre danú doménu.
- **Doplňujúce informácie (Additional)**: Dodatočné informácie, ktoré nepatria do predošlých častí (napr. IP adresy autoritatívnych serverov).




Obr. 1.2: Štruktúra DNS správy[2]

[2]Obrázok prevzatý z prezentácií prednášok predmetu ISA.

### 1.2.3 Typy záznamov

- **A (Address Record)** – Mapuje doménové meno na IPv4 adresu.
- **AAAA (IPv6 Address Record)** – Mapuje doménové meno na IPv6 adresu.
- **NS (Name Server)** – Určuje autoritatívne menné servery pre doménu.
- **MX (Mail Exchange)** – Informuje o poštovom serveri, ktorý prijíma poštu pre danú doménu.
- **SOA (Start of Authority)** – Obsahuje informácie o zóne DNS, ako je názov primárneho servera DNS pre doménu, sériové číslo, kontakt na správcu a časové intervaly pre aktualizácie(Refresh, Retry, Expire, Minimum).
- **CNAME (Canonical Name)** – Alias doménového mena; priradzuje kanonické (oficiálne) meno počítača k aliasu.
- **SRV (Service Record)** – Informácie o dostupných službách, napríklad názov hostiteľa alebo číslo portu.
- **...a ďalšie** – V tomto projekte sa však budeme zaoberať iba týmito typmi záznamov.

### 1.2.4 DNS Name Compression

Pri DNS komunikácii sa často využíva technika kompresie názvov domén (*name compression*), ktorá je definovaná v RFC 1035. Táto technika znižuje veľkosť DNS správ tým, že namiesto opakovaného uvádzania celých doménových mien sa v správe používajú odkazy na predchádzajúce výskyty týchto názvov. Využívajú na to ukazatele, ktoré odkazujú na pozíciu predchádzajúceho výskytu doménového mena v správe.

Ukazovateľ má špecifický formát: Dvojbajtová hodnota, kde prvé dva bity sú nastavené na "11"(čo signalizuje, že ide o ukazovateľ), a zvyšných 14 bitov obsahuje offset na miesto v DNS správe, kde sa dané doménové meno nachádza.

# Kapitola 2

# Implementácia

## 2.1 Návrh

Tento projekt sa venuje implementácii obdobného monitoru DNS komunikácie, ktorý dokáže zachytávať a analyzovať DNS pakety v reálnom čase počúvaním na rozhraní alebo zo súboru PCAP. Program `dns-monitor` je implementovaný v jazyku C a využíva knižnice `libpcap`[6] na zachytávanie paketov a následne ich parsuje pomocou štandardných knižníc a knižnicou `libresolv`[4].

Na základe požiadaviek zadania a štandardov RFC 1034[2] a RFC 1035 bol nástroj na spracovanie DNS správ `dns-monitor` implementovaný výlučne cez protokol UDP, keďže podpora pre TCP nebola v zadaní požadovaná. Rovnako pri nastavovaní BPF[10] (Berkeley Packet Filter) bol v programe zadefinovaný filter (okrem protokolu UDP) na štandardný port 53, ktorý sa bežne používa pre DNS komunikáciu[12].

## 2.2 Popis implementácie

### 2.2.1 Všeobecný popis architektúry programu

Program `dns-monitor` bol implementovaný v jazyku C s využitím jeho modularity. Je rozdelený do nasledujúcich častí:

- **dns-monitor.c**: Hlavný súbor programu; inicializuje aplikáciu, spracováva signály a spúšťa hlavný cyklus zachytávania paketov.
- **argparse.c/h**: Parsuje argumenty príkazového riadku a následne ich validuje.
- **pcap_handler.c/h**: Jadro logiky programu. Zodpovedný za zachytávanie a spracovanie DNS paketov v reálnom čase alebo z PCAP súboru pomocou knižnice `libpcap`. Podporuje IPv4 aj IPv6 komunikáciu z rôznych sieťových prostredí a extrahuje, vypisuje a ukladá informácie zo zachytených DNS správ.
- **dns_utils.c/h**: Modul, ktorý implementuje ukladanie prekladov a unikátnych doménových mien do súborov pomocou jednosmerných spojených zoznamov (single linked lists). Taktiež zabezpečí správne uvoľnenie týchto zoznamov a následné uzatvorenie súborov, do ktorých sa ukladal výstup, pri ukončení programu alebo pri zachytení signálu ako je napr. `SIGINT`.

### 2.2.2 Všeobecný popis toku programu

Program začína vo funkcii `main`, kde sa najprv spracujú a validujú argumenty príkazového riadku pomocou funkcií `parse_arguments`. Po úspešnom spracovaní argumentov sa inicializuje zachytávanie paketov alebo čítanie z PCAP súboru prostredníctvom funkcie `initialize_pcap` z modulu `pcap_handler`. Pre signály `SIGINT`, `SIGTERM` a `SIGQUIT` sa nastavia signal handlers pre korektné(graceful) ukončenie programu. Potom sa nastaví BPF filter na zachytávanie DNS prenosu, ako bolo spomenuté vyššie.

Hlavný cyklus programu sa spustí volaním funkcie `pcap_loop`, ktorá pre každý zachytený paket volá tzv. *callback funkciu* `packet_handler`. Vo funkcii `packet_handler` sa identifikuje typ linkovej vrstvy a extrahuje sa IP paket. Na základe verzie IP protokolu sa spracuje IPv4 alebo IPv6 hlavička a overí sa(aj napriek BPF filtru), či ide o UDP paket. Následne sa extrahuje UDP hlavička a DNS payload. DNS správa sa spracuje vo funkcii `parse_dns_packet`, kde sa pomocou knižnice `libresolv` a funkcie `ns_initparse` inicializuje parsovanie správy. z hlavičky DNS správy sa vytiahne identifikátor a príznaky, a zistia sa počty záznamov v jednotlivých sekciách. Potom sa spracujú sekcie Question, Answer, Authority a Additional pomocou funkcie `parse_resource_records`, kde sa prechádza cez jednotlivé záznamy a extrahujú sa potrebné informácie. Po spracovaní sa údaje vypíšu na štandardný výstup a prípadne sa uložia doménové mená a preklady IP adries pomocou funkcií z modulu `dns_utils`.

### 2.2.3 Podrobnejší popis zaujímavých častí a obmedzení

#### Dôsledok výberu protokolu UDP a portu 53 pre spracovanie DNS správ

Následkom tohto výberu bude to, že nástroj nebude až tak(teda vôbec) efektívny pri monitorovaní DNS dotazov prenášaných cez protokoly DNS over HTTPS (DoH) alebo DNS over TLS (DoT). Nakoľko tieto protokoly šifrujú celú DNS komunikáciu vrátane hlavičky, otázok a odpovedí pomocou TLS, bez dešifrovania týchto správ by nebolo možné čítať a ani analyzovať obsah DNS správ. Nakoniec bolo zvážené, že potrebné dešifrovanie týchto protokolov za účelom ich analýzy nie je cieľom tohto nástroja. [8] [9]

#### Spracovanie rôznych typov linkových vrstiev

Pri implementácii funkcie `packet_handler` sa rozhodlo podporovať spracovanie paketov zachytených na rôznych typoch linkových vrstiev[1], ktoré knižnica *libpcap* podporuje. Keďže tieto typy pokrývajú bežné scenáre, bola pre ne pridaná podpora:

- **DLT_EN10MB**: Ethernet (14 bajtov hlavičky), najrozšírenejší formát v bežných sieťach.
- **DLT_LINUX_SLL** a **SLL2**: Linux Cooked Capture (16 bajtov hlavičky) a LCCv2(20 bajtov), vhodný pre sledovanie na Linuxových systémoch.
- **DLT_NULL** a **DLT_LOOP**: Loopback rozhrania (4 bajty hlavičky), ktoré pokrývajú komunikáciu na loopback zariadeniach.
- **DLT_RAW, DLT_IPV4** a **DLT_IPV6**: Raw(surové) IP pakety bez linkovej hlavičky.

Na detekciu typu linkovej vrstvy je použitá funkcia `pcap_datalink`, a na základe detekovaného typu sa nastaví posun `packet` o veľkosť `link_header_length`(určuje veľkosť

hlavičky danej linkovej vrstvy). Týmto sa v rámci zachyteného paketu správne dostane na začiatok IP paketu a môže sa pokračovať v ďalšiom spracovaní sieťovej komunikácie.

### Parsovanie DNS správ pomocou `libresolv`

Na parsovanie DNS správ sa využila knižnica `libresolv`, konkrétne funkcie `ns_initparse`, `ns_parserr`, ktoré sú definované v hlavičkovom súbore `arpa/nameser.h`.

Najprv sa správa inicializuje pomocou funkcie `ns_initparse`, ktorá vytvorí potrebnú štruktúru `handle` na parsovanie DNS správy. Potom sa z DNS hlavičky jednoducho extrahuje identifikátor správy , príznaky a počty pomocou funkcií `ns_msg_id`, `ns_msg_getflag` a `ns_msg_count`.

Použitie `arpa/nameser.h`[5] výrazne zjednodušilo implementáciu spracovania správ. Avšak, narazilo sa na problém s multiplatformovou kompatibilitou, konkrétne na FreeBSD, kde knižnica `libresolv` spôsobovala problémy počas kompilácie. Kedže v týchto systémoch je táto knižnica zabudovaná, pri kompilácii flag `-libresolv` todo/dopíš.

Tento problém sa vyriešil úpravou Makefile, kde sa najprv skúša skompilovať program s knižnicou `libresolv`, a ak toto zlyhá, kompilácia sa zopakuje bez nej.

### Riešenie kompresie doménových mien

Doménové mená v DNS správach môžu byť komprimované podľa štandardu RFC 1035, čo znamená, že obsahujú odkazy na predchádzajúce časti správy. Na dekompresiu týchto mien, riešená vo funkcii `dns_extract_name`, sa využila funkcia `dn_expand` z knižnice `libresolv`. Táto funkcia prijíma ukazovatele na začiatok správy, koniec správy a komprimované meno, a vracia počet bajtov prečítaných z komprimovanej časti. Dekomprimované doménové meno sa uloží do poskytnutého výstupného bufferu a dĺžka tohto bufferu je určená posledným argumentom funkcie.

### Parsovanie DNS Resource Recordov

Na spracovanie DNS Resource Recordov (RR) sa znovu využila funkcia `ns_parserr` z knižnice `libresolv`, ktorá umožňuje extrahovať jednotlivé záznamy zo sekcií `Answer`, `Authority` a `Additional`.

Pri parsovaní jednotlivých záznamov sa pomocou funkcie `dn_expand` dekomprimuje názov domény, ak je komprimovaný podľa štandardu RFC 1035. Okrem toho sa dynamicky alokovuje pamäť pre RDATA (Resource Data) pomocou `malloc`. Veľkosť bufferu pre `rdata` je vypočítaná pre každý typ záznamu osobitne.

### Ukladanie doménových mien a prekladov

Modul *dns-utils* zahŕňa implementáciu ukladania doménových mien a prekladov IP adries do príslušných súborov. Používa spojené zoznamy (Linked lists), pri každom novom doménovom mene alebo preklade sa najprv prehľadá príslušný zoznam, či už daný záznam v ňom neexistuje. Ak nie, pridá sa na začiatok zoznamu a zapíše sa do súboru. Pri spracovaní záznamov typu A a AAAA sa ukladajú preklady doménových mien na IP adresy. Pri záznamoch typu NS, MX, SOA, CNAME a SRV a v sekcii Question sa ukladajú doménové mená nachádzajúce sa v poliach *Name* a *RDATA*.

# Kapitola 3

# Návod na použitie

Projekt je možné skompilovať pomocou príkazu `make`.

Na spustenie programu sa používa nasledujúca syntax:

```
./dns-monitor (-i <interface> | -p <pcapfile>) [-v] [-d <domainsfile>] [-t
                        <translationsfile>] [-h]
```

Popis parametrov:

- `-i <interface>`: Názov sieťového rozhrania, na ktorom bude program naslúchať DNS komunikácii.
- `-p <pcapfile>`: Názov PCAP súboru, ktorý program spracuje.
- `-v`: Režim "verbose", v ktorom program vypisuje kompletné detaily o DNS správach, vrátane hlavičiek a obsahu jednotlivých sekcií.
- `-d <domainsfile>`: Názov súboru, do ktorého sa uložia všetky UNIKÁTNE zachytené a spracované doménové mená.
- `-t <translationsfile>`: Názov súboru, do ktorého sa uložia preklady doménových mien na IPv4/IPv6 adresy.
- `-h`: Vypíše nápovedu k programu (help).

# Kapitola 4

# Testovanie

## 4.1 skript test.sh

Za účelom testovania programu `dns-monitor` bol vytvorený jednoduchý skript `test.sh`. Tento skript sa nachádza v adresári `tests/dig` a je možné ho spustiť príkazom `make test` z koreňového adresára projektu.
*!pozn.: skript je potrebné spúšťať s oprávneniami `sudo` a vyžaduje existenciu nástroja `dig`!*

### 4.1.1 Popis testovacieho skriptu

Po spustení skriptu je potrebné zadať názov sieťového rozhrania, na ktorom bude `dns-monitor` načúvať. Program sa spustí vo verbose režime s parametrami `-d domains.txt` a `-t translations.txt`, pričom výstup sa presmeruje do súboru `output.txt` a prípadné chybové hlášky do súboru `errors.txt`.

Následne skript číta zo súboru `dns_queries.txt` dotazy, ktoré sa majú pomocou nástroja `dig` vykonať (pr.: `dig seznam.cz AAAA`). Výstup z nástroja `dig` sa ukladá do súboru `dig_output.txt`.

### 4.1.2 Spustenie testu

Test je možné spustiť príkazom:

```
make test
```

Tento príkaz skompiluje program `dns-monitor` a spustí testovací skript.

### 4.1.3 Obsah súboru `dns_queries.txt`

Súbor `dns_queries.txt` obsahuje zoznam DNS dotazov vo formáte:

```
<doména> <typ_dotazu>
```

Príklad obsahu:

```
fit.vutbr.cz A
www.seznam.cz CNAME
_sip._tcp.github.com SRV
```

### 4.1.4 Vyhodnotenie výsledkov

Výsledky testu boli manuálne porovnané medzi výstupom programu `dns-monitor` a výstupom príkazu `dig`:

- Overilo sa, že `dns-monitor` správne zachytil a vypísal všetky DNS správy, vrátane správnej interpretácie sekcií Question, Answer, Authority a Additional.
- Súbor `errors.txt` slúži na kontrolu výstupu prípadných chýb. Nezaznamenali sa žiadne chyby a jediný výstup v tomto súbore je po zachytení signálu programom, ktorý informuje o ukončení programu.
- Obsah súborov `domains.txt` a `translations.txt` overil správnosť zapisovania doménových mien a aj ich preklady na IP adresy.

## 4.2 Testovanie s PCAP súbormi

Okrem živého zachytávania DNS komunikácie na vybranom rozhraní sa vykonalo aj testovanie čítaním PCAP súborov.

### 4.2.1 PCAP súbor `alltypes.pcap`

Súbor `alltypes.pcap` obsahuje DNS komunikáciu so všetkými podporovanými typmi záznamov (A, AAAA, NS, MX, SOA, CNAME, SRV). Súbor bol vytvorený pomocou použitia príkazu `dig` pre jednotlivé typy dotazov na doménu `example.com` a bol zachytený a exportovaný pomocou nástroju `Wireshark`[1].

Program `dns-monitor` bol spustený s parametrom `-p` na načítanie PCAP súboru a jeho výstupy boli uložené podobne ako pri použití skriptu `test`.

Výstup sa porovnával manuálne s výstupom súboru `alltypes.pcap` otvoreného v nástroji Wireshark. V závere sa overila správnosť spracovania, pričom sa taktiež sledovalo správne ukladanie domén a prekladov, vrátane výpisu všetkých sekcií a záznamov.

### 4.2.2 PCAP súbor `v6.pcap`

Súbor `v6.pcap` obsahuje DNS komunikáciu s IPv6 adresami. Overenie prebiehalo totožne ako pri súbore `alltypes.pcap`.

[1] https://www.wireshark.org/

# Literatúra


[1] *LINK-LAYER HEADER TYPES* [Tcpdump.org]. Unknown date. Dostupné z: https://www.tcpdump.org/linktypes.html.

[2] *Domain names - concepts and facilities* [RFC 1034]. RFC Editor, november 1987. DOI: 10.17487/RFC1034. Dostupné z: https://www.rfc-editor.org/info/rfc1034.

[3] *Domain names - implementation and specification* [RFC 1035]. RFC Editor, november 1987. DOI: 10.17487/RFC1035. Dostupné z: https://www.rfc-editor.org/info/rfc1035.

[4] *libresolv - Resolver Library* [FreeBSD Man Pages]. August 1998. 4 August 1998. Dostupné z: https://man.freebsd.org/cgi/man.cgi?query=libresolv&manpath=SunOS+5.7.

[5] *Nameser.h File Reference* [Generated for RTL-lwIP-0.4]. 2004. Generated on Wed Jan 14 12:59:07 2004. Dostupné z: https://rtl-lwip.sourceforge.net/sources/nameser_8h.html.

[6] *pcap(3PCAP) Man Page* [Tcpdump.org Documentation]. September 2024. Updated: 18 September 2024. Dostupné z: https://www.tcpdump.org/manpages/pcap.3pcap.html.

[7] Doc. Ing. Petr Matoušek, M. *Systém DNS* [Presentation, Lecture 5]. Unknown date. Dostupné z: https://moodle.vut.cz/pluginfile.php/898253/mod_resource/content/5/isa-dns.pdf.

[8] Hoffman, P. E. a McManus, P. *DNS Queries over HTTPS (DoH)* [RFC 8484]. RFC Editor, október 2018. DOI: 10.17487/RFC8484. Dostupné z: https://www.rfc-editor.org/info/rfc8484.

[9] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D. et al. *Specification for DNS over Transport Layer Security (TLS)* [RFC 7858]. RFC Editor, máj 2016. DOI: 10.17487/RFC7858. Dostupné z: https://www.rfc-editor.org/info/rfc7858.

[10] IBM. *Berkeley Packet Filters* [IBM Documentation]. Jún 2023. Last Updated: 2023-06-25. Dostupné z: https://www.ibm.com/docs/en/qsip/7.4?topic=queries-berkeley-packet-filters.

[11] Kentik. *DNS Monitoring: An Essential Aspect of Network Health* [Kentipedia]. Feb 2024. Updated: February 22, 2024. Dostupné z: https://www.kentik.com/kentipedia/dns-monitoring/.

[12] Matoušek, P. *Síťové služby a jejich architektura.* Publishing house of Brno University of Technology VUTIUM, 2014. 396 s. ISBN 978-80-214-3766-1. Na strane 124, sekcia 3.4. Dostupné z: https://www.fit.vut.cz/research/publication/10567.