VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentácia k projektu do predmetu ISA

Export DNS informácií pomocou protokolu Syslog

19. novembra 2018

Matúš Liščinský (xlisci02)

Obsah

1 Úvod **2**
1.1 DNS v skratke 2
1.2 Syslog protokol 4

2 Základné informácie o programe **5**

3 Popis implementácie **5**
3.1 Odchytávanie sieťovej premávky a prvotné spracovanie paketu 5
3.2 Práca s DNS záznamom 5
3.3 Vytváranie štatistík a ich zasielanie na logovací server 6
3.4 Zaujímavé časti implementácie 7

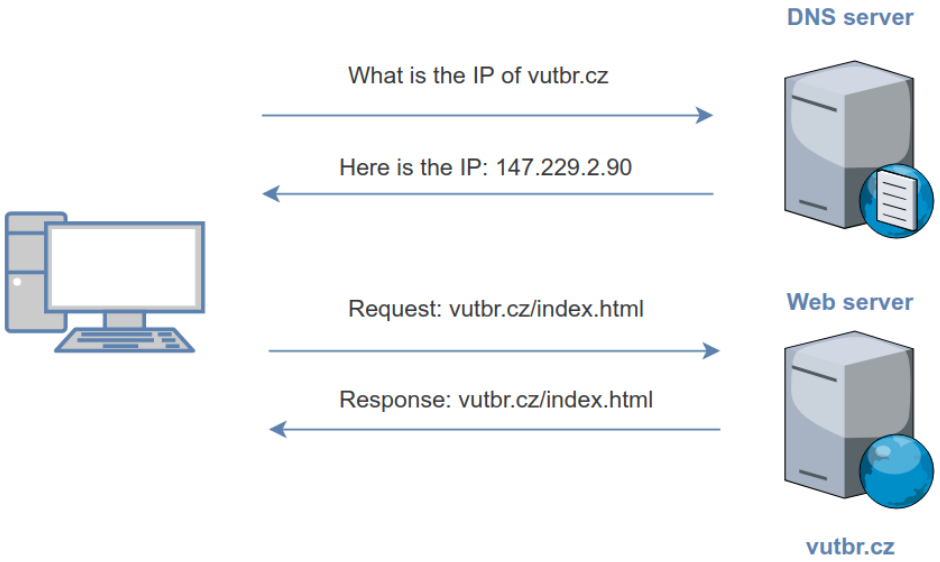
4 Návod na použitie **8**
4.1 Spracovanie štatistík zo súboru 8
4.2 Spracovanie online sieťovej prevádzky 8
4.3 Príklady štatistickej interpretácie dát DNS záznamov 9

1 Úvod

Dokumentácia popisuje implementáciu aplikácie, ktorá spracúva dáta protokolu DNS a vybrané štatistiky exportuje pomocou protokolu Syslog na logovací server.

1.1 DNS v skratke

DNS *(Domain Name System)* je systém, ktorého úlohou je preklad doménových mien na ich IP adresy a naopak. Ukladá totiž prístup k informácii o názve stroja (hostname) a názve domény v istej distribuovanej databáze v počítačových sieťach. Doménové meno si užívateľ ľahko zapamätá a dokáže ho intuitívne napísať napríklad do webového prehliadača.



Obr. 1: Zjednodušená ilustrácia účelu DNS

Túto funkciu zabezpečujú DNS servery, ktoré obsahujú databázu domén a IP adries s nimi spojených. Internetový prehliadač pri zadaní adresy stránky zistí, na ktorom DNS serveri sa nachádzajú záznamy pre konkrétnu doménu a na DNS serveri zisťuje IP adresy, ktoré sú pre doménu nastavené. DNS záznam domény obsahuje viacero typov záznamov, z ktorých každý má inú funkciu. Najpoužívanejšie sú nasledovné typy záznamov [1]:

- **A** slúži pre nastavanie konkrétnej IP adresy, z ktorej sa má načítavať obsah pre danú adresu (doménu alebo subdoménu). [9]
- **AAAA** záznam je alternatívou k záznamu typu A. Používa sa v prípade, že je IP adresa cieľového serveru vo formáte IPv6. [11]
- **CNAME** je skratka pre Canonical Name record a spôsobuje, že jeden názov domény je aliasom pre iný. Takáto doména má platné všetky subdomény a DNS záznamy originálu. V DNS zázname je nadradený nad A záznamami a MX záznamami, teda prehliadač najskôr prečíta CNAME záznam, a keď žiadny CNAME záznam neexistuje, dopytuje sa na konkrétnu IP adresu v A záznamoch. [9]
- **MX** záznam slúži k nastaveniu servera, kam bude smerovať elektronická pošta zasielaná na e-maily na danej doméne. [9]
- **TXT** záznamy slúžia na pridanie textových informácií do DNS záznamov domény. To sa často využíva pre overenie domény alebo doplnkové informácie potrebné k prepojeniu s domény s externými službami. [9]

- **SPF** sú DNS TXT záznamy, ktoré zahŕňajú informácie o tom, ktoré servery majú oprávnenie odosielať maily z domény, na ktorej sú nastavené. Slúžia ako ochrana proti posielaniu spamu, keď server, ktorý prijíma poštu pri zapnutom overovaní SPF kontroluje, či je pošta pre konkrétnu doménu odoslaná zo servera, z ktorého má odosielanie povolené v SPF zázname. [7]
- **SRV** záznam slúži na určenie adresy serverov pre špecifické služby. Špecialitou SRV záznamov je, že umožňujú rozloženie záťaže na viacero adries na základe váhy a priority. [6]
- **NS** - záznam umožňuje určiť adresu, kde sú umiestnené DNS záznamy danej domény. Zmena sa vykonáva u národného registrátora danej domény, ktorý vďaka tomu vie, kam nasmerovať všetky prichádzajúce dopyty na danú adresu. [9]
- **SOA** - (start of authority record) alebo začiatok záznamu autority špecifikuje DNS server poskytujúci autoritatívnu informáciu o internetovej doméne. [9]
- **PTR** - mapuje hostname na kanonické meno stroja. Ak je PTR záznam nastavený na doménu in-addr.arpa, znamená to, že IP adresa implementuje *tzv. reverse DNS lookup* pre danú adresu. [9]
- **DNSKEY** - záznam DNSKEY určuje záznam kľúča DNSSEC. Zóna podpisuje svoje autoritatívne sady záznamov prostriedkov súkromným kľúčom a príslušný verejný kľúč ukladá do záznamu prostriedku DNSKEY. [2]
- **DS** - (Delegation Signer) - je umiestnený v nadradenej DNS doméne a obsahuje otlačok verejného kľúča uloženého v DNSKEY zázname podpísanej domény. [2]
- **NSEC** - rieši problém vieryhodného odmietnutia/popretia existencie záznamu. Pri dotaze na neexistujúci záznam je vrátený podpísaný záznam NSEC pokrývajúci lexikálny priestor medzi predchádzajúcim a nasledujúcim existujícim záznamom. NSEC záznamy tvoria spojový zoznam, pomocou ktorého je možné prejsť všetny záznamy v zóne. [2]
- **NSEC3** - slúži k vyriešeniu problému autorizovanej odpovedi o neexistencii domény. Vrámci podpísania zónového súboru obsahujúceho všetky záznamy pre danú doménu dôjde k tomu, že medzi každú subdoménu v evidencii je vložený NSEC záznam, ktorý okdazuje na ďalšiu subdoménu v abecednom poradí. Tento NSEC záznam je potom štandardným spôsobom podpísaný. [3] [8]
- **NSEC3PARAM** - tento záznam obsahuje parametre pre záznam NSEC3. Tieto parametre sú potrebné pre autoritatívny server domény na vyrátanie hashu domén použitých pri odpovedi o neexistencii doménového mena. Prítomnosť záznamu NSECPARAM v zóne určuje, že autoritatívny server pri negatívnej odpovedi ma použiť záznam NSEC3 s predpísanými parametrami v NSECPARAM. [8]
- **RRSIG** - záznam určuje digitálne podpisy, ktoré sa používajú v procese overovánia DNSSEC. Obsahuje používaný kryptografický algoritmus, meno, kým bol podpísaný, čas uvedenia a expirácie. [2]

1.2 Syslog protokol

Syslog protokol je sieťový protokol, ktorý poskytuje komplexný mechanizmus hlásenia správ. Popisuje spôsoby a formát pre záznam systémových udalostí. Je využíváný najmä v unixových systémoch.

Medzi informácie poskytované zdrojom syslog správy patrí kód zariadenia *(facility code)*, úroveň závažnosti *(severity level)* a samotný záznam. Záznam môže byť uložený lokálne (napr. súbor `/var/log/messages`) alebo možno záznamy preposielať na vzdialený server. Súčasťou záznamu môže byť aj doménové meno resp. IP adresa zdroja záznamu. [10]

Formát syslog správy je pomocou ABNF *(Augmented Backus-Naur form)* popísaný v RFC 5424. Každá syslog správa musí na začiatku obsahovať hodnotu priority vo formáte <PRI>. Tá sa počíta podľa vzťahu:

`PRI = FACILITY_CODE * 8 + SEVERITY_LEVEL`.

V našom prípade je Facility nastavené na *local0* a Severity na *Informational*, tj. hodnota priority bude:

`16 * 8 + 6 = 134`, v správe zapísaná ako `<134>`.

Facility code (kód zariadenia) sa používa na špecifikáciu typu programu, ktorý správu hlási. Správy od rôznych zariadení môžu byť spracované inak. Zoznam dostupných zariadení je možno nájsť opäť v dokumente popisujúcom syslog protokol RFC 5424. [4].

Severity level (závažnosť záznamu) je rozdelená na 8 úrovní od najzávažnejšej(0) až po ladiace informácie(7). Obvykle sú v syslogu zaznamenávané udalosti od úrovne 4. [5]

Hodnota	Závažnosť	Popis
0	Emergency	Najvyššia závažnosť, systém je nestabilný
1	Alert	Udalosť vyžadujúca okamžitú akciu
2	Critical	Kritická chyba systému
3	Error	Chyba programu
4	Warning	Varovanie programu
5	Notice	Dôležité informačné záznamy
6	Informational	Informačné správy nevyžadujúce špeciálnu pozornosť
7	Debug	Ladiace informácie

Tabuľka 1: Úplný zoznam úrovní závažností záznamov

2 Základné informácie o programe

Zdrojový kód programu je napísaný v jazyku C/C++ a logicky umiestnený do niekoľkých modulov. Zadanie špecifikuje vlastný referenčný systém (CentOS7), na ktorom je program preložiteľný a spustiteľný.

Program spracúvava DNS sieťovú prevádzku na určitom rozhraní *(online mode)*, prípadne na všetkých rozhraniach, vytvára štatistiky a interpretuje ich. Rovnako je dostupná aj možnosť spracovať štatistiky DNS prevádzky z offline zo súboru `*.pcap` *(offline mode)*. Aplikácia poskytuje niekoľko spôsobov reprezentácie zozbieraných štatistík. Jednou z nich je synchrónne zasielanie na užívateľom definovaný syslog server, ďalšou je klasický výpis na štandardný výstup.

Pri online monitorovaní sieťovej prevádzky si používateľ môže nechať pravidelne posielať zozbierané štatistiky na definovaný logovací server. Informácie sa objavia na štandardnom výstype po zaslaní signálu `SIGUSR1` programu, respektíve v prípade offline režimu po spracovaní súboru.

3 Popis implementácie

3.1 Odchytávanie sieťovej premávky a prvotné spracovanie paketu

Základnym pilierom aplikácie je C/C++ knižnica pre zachytávanie sieťovej prevádzky `libpcap`. Ponúka dokonca možnosť elegantne filtrovať pakety použitím BPF *(Berkeley Packet Filter)*. Konkrétne táto aplikácia chce príjímať iba pakety so zdrojovým portom 53 a filter tak vyzerá jednoducho: *"src port 53"*. Dokumentáciu k tejto knižnici je možné si prečítať na http://www.tcpdump.org.

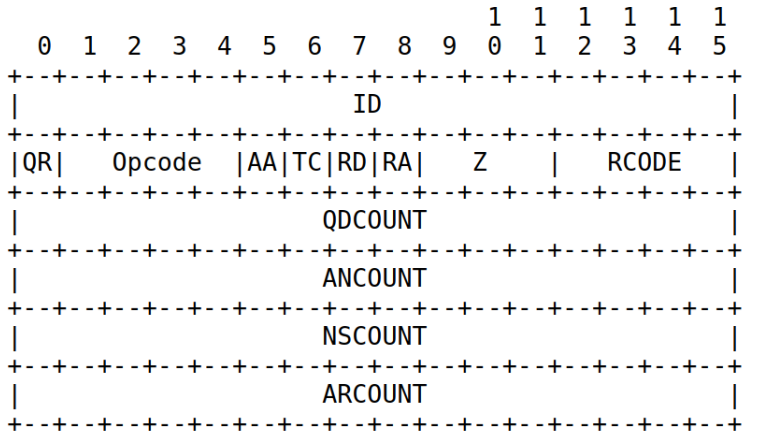
Zachytený paket je ďalej predávaný fukniám na jeho spracovanie a prvotne s ním pracujú funkcie, ktorých deklarácie sa náchádzajú v hlavičkovom súbore `pkt-headers.h`. Tieto funkcie kontrolujú hlavičky vrstiev L2, L3 a L4 OSI modelu, konkrétne ide o :

- **Layer 2 :** *Ethernet, Linux "cooked"capture (SLL)*
- **Layer 3 :** *IPv4, IPv6, IPv6 extension headers*
- **Layer 4 :** *TCP, UDP*

Pri kažom pokuse o čítanie z pamäte kde je uložený paket, sa najprv otestuje, či ukazateľ ukazuje na adresu v pamäti, z ktorej možno bezpečne čítať. Používa sa pri tom ukazateľová aritmetika zahrnutá v inline funckii `check_pointer()`. Dá sa tak odhaliť poškodený či chybný paket, ktorý bude zahodený.

3.2 Práca s DNS záznamom

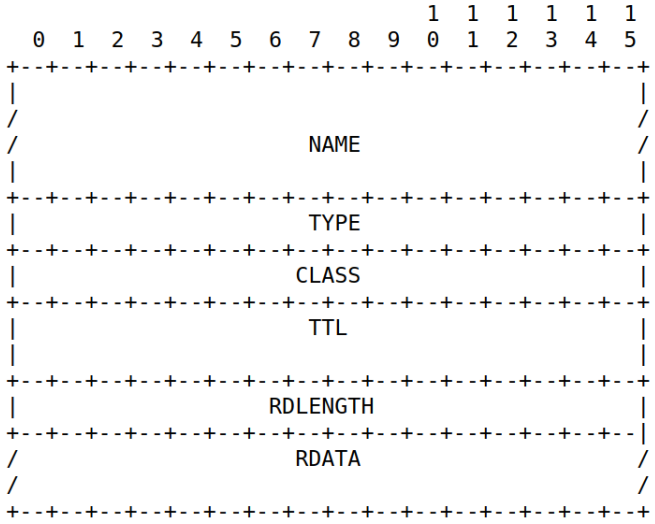
DNS komunikácia beží na TCP/UDP porte 53. V prípade, že ide o korektný paket s DNS záznamom, dostáva sa na rad spracovanie DNS hlavičky. Túto časť už rieši samostatný modul `dns-msg`. Hlavičku DNS majú spoločnú všetky typy DNS záznamov a jej štruktúra vyzerá nasledovne: [9]



Obr. 2: Hlavička DNS správy podľa RFC 1035

Aplikácia vytvára štatistiky z **DNS response** správ a tie majú nastevený bit `QR` na hodnotu 1. Použiteľná správa musí mať okrem toho nulovú hodnotu poľa `RCODE` (nedošlo k chybe/zlyhaniu) a kladnú hodnotu poľa `ANCOUNT` definujúcu počet záznamov v sekcii odpovedí *(answer section)*.

Najzaujímavejšia časť je práve sekcia odpovedí odkiaľ sa dolujú informácie pre štatistiku. Jednotlivé typy záznamov maju totožný formát prvých 5 polí záznamu: `NAME, TYPE, CLASS, TTL, RDLENGTH`, líšia sa od seba štruktúrou a informáciami, ktoré nesú v poli `RDATA`.



Obr. 3: Telo DNS odpovede podľa RFC 1035

Aplikácia je schopná spracovať a zaznamenať do štatistík 16 typov záznamov, menovite:

- **A** - RFC 1035
- **AAAA** - RFC 3596
- **CNAME** - RFC 1035
- **MX** - RFC 1035
- **NS** - RFC 1035
- **SOA** - RFC 1035
- **TXT** - RFC 1035
- **SPF** - RFC 7208
- **PTR** - RFC 1035
- **SRV** - RFC 2782
- **DNSKEY** - RFC 4034
- **DS** - RFC 4034
- **NSEC** - RFC 4034
- **NSEC3** - RFC 5155
- **NSEC3PARAM** - RFC 5155
- **RRSIG** - RFC 4034

Ide teda o najčastejšie používané DNS a DNSSEC záznamy a ich význam a účel je opísaný v úvode, v kapitole 1.1. V prípade, že aplikácia nepozná daný typ záznamu zapíše ho do štatistík ako *"Unsupported DNS record type"* a dáta záznamu ignoruje.

3.3 Vytváranie štatistík a ich zasielanie na logovací server

Pre ukladanie zaznamenaných dát je použitá štandardná C++ mapa `std::map` a túto časť aplikácie zastrešuje modul `stats`. O ich prípadné posielanie na server sa starajú funkcie z modulu `syslog`. Jednotlivé štatistické záznamy sa posielajú na logovací server v čo najväčších skupenstvách s dĺžkovým limitom 1024B. Používa sa pri tom transportný protokol UDP s portom 514.

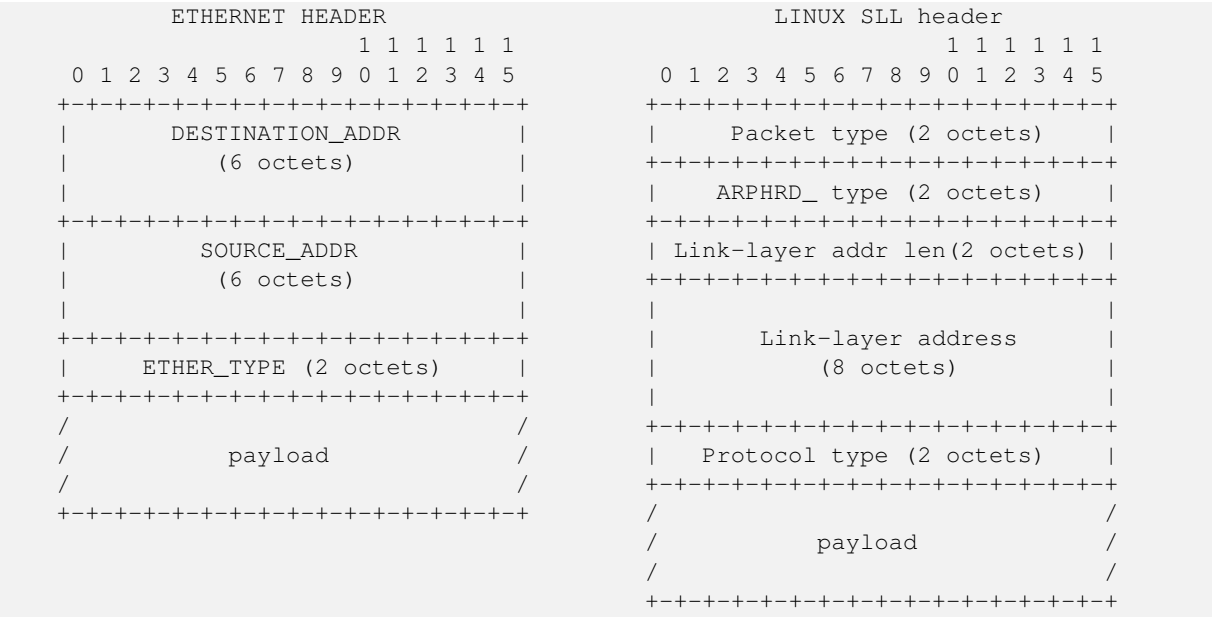
3.4 Zaujímavé časti implementácie

- **TCP segmentácia:** V prípade komunikácie cez TCP *(Transmission Control Protocol)* program zachytáva a ukladá jednotlivé segmenty. V momente prijatia všetkých segmentov daného TCP streamu dôjde k poskladaniu do jedného bloku, ktorý bude predložený k ďalšiemu spracovaniu.

  TCP segmenty sa ukladajú do mapy, kde kľúčom je sekvenčné číslo prípadného nasledujúceho segmentu a hodnotou je vektor TCP segmentov patriacich do daného TCP streamu spolu s ich veľkosťami v bytoch.

```
std::map <unsigned int,std::vector <std::pair <const u_char *,unsigned int>>>tcp_map;
```

- **Default interface:** Pri online režime aplikácie nieje nevyhnutné špecifikovať rozhranie, vtedy sa bude načúvať na všetkých dostupných rozhraniach a využívať tzv. *Linux cooked capture encapsulation*.
  Pre zaujímavosť porovnanie LINUX SLL s klasickou ETHERNET hlavičkou:



- **Multitasking**: Pri odosielaní dát na logovací server či ich výpis na štandardný výstup sa vykoná `fork()` hlavného procesu. Ten nestráca čas týmito vedľajšími úlohami, pokračuje v spracovávaní prichádzajúcich paketov a detský proces sa využije na výpis alebo zaslanie štatistík a ukončí sa.
  Vytvorenie nového procesu pre výpis, v prípade chyby výpis vykonáva hlavný proces:
  `stats.cc`: *line 29-45*

```
void print_handler(int signo){
    (void)signo;

    /* fork */
    pid_t f_pid;

    /* On success parent process leaves the func., and child will continue.
       On failure parent will continue */
    if((f_pid = fork()) > 0){
        return;
    }
    print_stats();

    /* exit child process */
    if(f_pid == 0)
        exit(0);
}
```

- **IP fragmentácia**: nieje v tomto programe riešená.

4 Návod na použitie

Pred prvým spustením je nutné projekt preložiť príkazom **`make`** alebo **`make all`**.
Aplikácia sa spúšťa nasledovne:
`$ ./dns-export [-r file.pcap] [-i interface] [-s syslog-server] [-t seconds]`
Pozn. Na použitie online režimu aplikácie je potrebné pracovať ako *root*.

Podrobnejší návod k spusteniu aplikácie možno zobraziť pomocou prepínača `-h`:
`$ ./dns-export -h`

4.1 Spracovanie štatistík zo súboru

Jedná sa o tzv. offline režim aplikácie a nevyhnutným argumentom je cesta k súboru `*.pcap`.
Existuje niekoľko možností použitia:

1. Po analýze a spracovaní súboru `dns.pcap` štatistiky vypíše na STDOUT
   - **`$ ./dns-export -r dns.pcap`**
2. Štatistiky sa pošlú na syslog server s adresou `192.168.60.1`
   - **`$ ./dns-export -r dns.pcap -s 192.168.60.1`**

4.2 Spracovanie online sieťovej prevádzky

V tomto režime program načúva na rozhraní definovanom vstupným argumentom *interface*, resp. dokáže načúvať na všetkých rozhraniach ak argument používateľ nezadá vôbec alebo ako argument rozhrania použije reťazec *"any"*. Existuje niekoľko možností použitia:

1. Program načúva na sieťovom rozhraní `eth0`, štatistiky vypíše na STDOUT po prijatí signálu `SIGUSR1`
   - **`$ sudo ./dns-export -i eth0`**
2. Štatistiky sa budú posielať na logovací server s adresou `192.168.60.1` každých 60 sekúnd (default)
   - **`$ sudo ./dns-export -i eth0 -s 192.168.60.1`**
3. Štatistiky sa budú posielať na logovací server každých `20` sekúnd
   - **`$ sudo ./dns-export -i eth0 -s 192.168.60.1 -t 20`**

Všetky tieto kombinácie je možné použiť aj pre načúvanie na všetkých rozhraniach, definovaním rozhrania ako *"any"* alebo vynechania tohoto argumentu

1. - **`$ sudo ./dns-export`**
   - **`$ sudo ./dns-export -i any`**
2. - **`$ sudo ./dns-export -s 192.168.60.1`**
   - **`$ sudo ./dns-export -i any -s 192.168.60.1`**
3. - **`$ sudo ./dns-export -s 192.168.60.1 -t 20`**
   - **`$ sudo ./dns-export -i any -s 192.168.60.1 -t 20`**

4.3 Príklady štatistickej interpretácie dát DNS záznamov

Dáta DNSSEC záznamov sú prezentované podľa formátu v príslušnom RFC v podkapitole *The <TYPE> RR Presentation Format*. Informácie z ostatných DNS záznamov sú interpretované zhodne ako pri použití nástroja **`dig`**.

- **A** záznam, RDATA = *(ADDRESS)*
  `www.google.com A "216.58.208.196" 1`
- **AAAA** záznam, RDATA = *(IPv6 ADDRESS)*
  `www.google.com AAAA "2a00:1450:4007:80d::2004" 1`
- **CNAME** záznam, RDATA = *(CNAME)*
  `www.bbc.co.uk CNAME "www.bbc.net.uk" 2`
- **MX** záznam, RDATA = *(PREFERENCE, EXCHANGE)*
  `feec.vutbr.cz MX "10 kos.feec.vutbr.cz" 2`
- **NS** záznam, RDATA = *(NSDNAME)*
  `vutbr.cz NS "rhino.cis.vutbr.cz" 1`
- **SOA** záznam, RDATA = *(MNAME, RNAME, SERIAL, REFRESH, RETRY, EXPIRE, MINIMUM)*
  `ni.cz SOA "ns1.vultr.com dnsadm.choopa.com 1472492563 10800 3600 604800 3600" 1`
- **TXT** záznam, RDATA = *(TXT-DATA)*
  `vutbr.cz TXT "MS=ms21627876" 1`
- **PTR** záznam, RDATA = *(PTRDNAME)*
  `196.208.58.216.in-addr.arpa PTR "par10s21-in-f4.1e100.net" 1`
- **SRV** záznam, RDATA = *(PRIORITY, WEIGHT, PORT, TARGET)*
  `_sip._tcp.cesnet.cz SRV "100 10 5060 cyrus.cesnet.cz" 1`
- **DNSKEY** záznam, RDATA = *(FLAGS, PROTOCOL, ALGORITHM, PUBLIC KEY)*
  `ietf.org DNSKEY "256 3 5 AwEAAdD...Bbhk=" 1`
- **DS** záznam, RDATA = *(KEY TAG, ALGORITHM, DIGEST TYPE, DIGEST)*
  `ietf.org DS "45586 5 1 D0FDF996D1AF2CCDBDC942B02CB02D379629E20B" 1`
- **NSEC** záznam, RDATA = *(NEXT DOMAIN NAME, RRSET TYPES)*
  `feec.vutbr.cz NSEC "_sipfederationtls._tcp.feec.vutbr.cz NS SOA MX RRSIG NSEC DNSKEY" 1`
- **NSEC3PARAM** záznam, RDATA = *(HASH ALG, FLAGS, ITERATIONS, SALT)*
  `sshfp.net NSEC3PARAM "1 0 20 80637D8AF055B5EECA2A621EDAAA3C5E" 1`
- **RRSIG** záznam, RDATA = *(TYPE COVERED, ALGORITHM, LABELS, ORIGINAL TTL, SIGNATURE EXPIRATION, SIGNATURE INCEPTION, KEY TAG, SIGNER NAME, SIGNATURE)*
  `www.fit.vutbr.cz RRSIG "A 5 4 14400 20181209094549 20181109094549 35421 fit.vutbr.cz ULAD...uqXo=" 1`
- Ak záznam **nepatrí** do množiny podporovaných typov DNS záznamov menovaných v kapitole 3.2, aplikácia dokáže zistiť o aký typ ide, no dáta ignoruje a vypíše reťazec *"Unsupported DNS record type"*:
  `google.com CAA "Unsupported DNS record type" 1`

Literatúra

[1] Aké typy DNS záznamov existujú ? online, 2016.
URL https://www.websupport.sk/support/index.php?type=page&urlcode=769240&title=Ak%C3%A9-typy-DNS-z%C3%A1znamov-existuj%C3%BA

[2] Arends, R.; Austein, R.; Larson, M.; aj.: Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), Březen 2005, updated by RFCs 4470, 6014.
URL https://tools.ietf.org/html/rfc4034

[3] Caletka, O.: Principy a správa DNS. online, 2015.
URL https://xn--ondej-kcb.caletka.cz/dl/slidy/20150915-CESNET-Principy_a_sprava_DNS.pdf

[4] Gerhards, R.: The Syslog Protocol. RFC 5424 (Proposed Standard), March 2009.
URL https://tools.ietf.org/html/rfc5424

[5] Grégr, M.: Network and Services logging. online, 2018.
URL https://wis.fit.vutbr.cz/FIT/st/cfs.php?file=%2Fcourse%2FISA-IT%2Flectures%2Fisa-logovani.pdf&cid=12786

[6] Gulbrandsen, A.; Vixie, P.; Esibov, L.: A DNS RR for specifying the location of services (DNS SRV). RFC 2782 (Proposed Standard), February 2000.
URL https://tools.ietf.org/html/rfc2782

[7] Kitterman, S.: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. April 2014.
URL https://tools.ietf.org/html/rfc7208

[8] Laurie, B.; Sisson, G.; Arends, R.; aj.: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. RFC 5155 (Proposed Standard), March 2008.
URL https://tools.ietf.org/html/rfc5155

[9] Mockapetris, P.: Domain names - implementation and specification. RFC 1035 (Standard), November 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343.
URL https://tools.ietf.org/html/rfc1035

[10] NES@FIT: Laboratorní manuál pro cvičení z předmětu ISA – Síťové aplikace a správa sítí. online, 2018.
URL https://wis.fit.vutbr.cz/FIT/st/cfs.php?file=%2Fcourse%2FISA-IT%2Fhwlabs%2Fmanual.pdf

[11] Thomson, S.; Huitema, C.; Ksinant, V.; aj.: DNS Extensions to Support IP Version 6. RFC 3596 (Draft Standard), October 2003.
URL https://tools.ietf.org/html/rfc3596