VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DNSSEC ONLINE WEBOVÁ MAPA

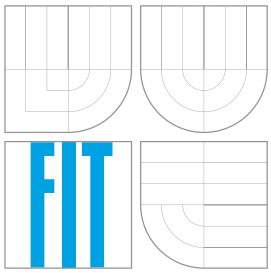
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETER SCHERFEL

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DNSSEC ONLINE WEBOVÁ MAPA

DNSSEC ONLINE WEB MAP

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETER SCHERFEL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2011

Abstrakt

Táto práca sa zaoberá problematikou zabezpečenia systému DNS pomocou technológie DNSSEC. Má za úlohu preštudovať systém podpisovania záznamov DNSSEC a možnosti získavania informácií o stave zabezpečenia domén. Hlavným cieľom práce je vytvoriť systém, ktorý bude automaticky sledovať zabezpečenia domén TLD a SLD. Informácie o zabezpečení bude priebežne zbierať a automaticky aktualizovať. Z voľne dostupného systému Whois, získa geografické adresy, ktoré pomocou API Google Maps prevedie na geografické súradnice. Získané dáta ukladá do databázy a pomocou API Google Maps vykresľuje na geografickú mapu svet.

Abstract

This master's thesis is focusedurity extension DNSSEC of DNS system. It will go trough the basics of securing DNS by DNSSEC and possibilities of collecting information about the state of DNSSEC in present. Its main goal is to design application, which will be monitoring security state of TLD and SLD domains. Implemented system is gathering information from public Whois database. Gathered information are transformed to geographical coordinates by Google Maps API. All gathered information are stored in database and placed on geographical map of the world by Google Maps Api.

Klíčová slova

DNSSEC, DNS, geografická lokalizácia, Google Maps

Keywords

DNSSEC, DNS, geographical mapping, Google Maps

Citace

Peter Scherfel: DNSSEC online webová mapa, diplomová práce, Brno, FIT VUT v Brně, 2011

DNSSEC online webová mapa

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter Scherfel
24. května 2011

Poděkování

Rád by som sa poďakoval svojmu vedúcemu práce Ing. Petru Matouškovi Ph.D. za jeho ochotný prístup a odbornú pomoc pri vypracovávaní tejto práce. Ďalej by som sa chcel poďakovať Ondrejovi Surému zo združenia CZ.NIC za odporné konzultácie pri vytváraní tejto práce.

Obsah

Kapitola 1

Úvod

Všetky aplikácie, ktoré zabezpečujú komunikáciu medzi počítačmi pripojenými do siete, používajú k identifikácií sieťových uzlov IP adresy. Pre bežného užívateľa by bolo veľmi náročné pamätať si IP adresy počítačov (webové servery, mailové servery a podobne), ku ktorým sa každodenne pripájajú a využívajú ich služby. Užívateľom siete je jednoduchšie si zapamätať slovné pomenovanie počítača - *doménové meno*, namiesto číselnej adresy. Aj pri vývoji sieťových aplikácií je výhodnejšie používať doménové meno pri adresovaní počítačov. Ak sa zmení IP adresa počítača v sieti, tak jeho doménové meno ostane.

Väzby medzi IP adresami a doménovými menami sú uložené v databázy DNS (Domain Name System). DNS je celosvetovo distribuovaná databáza, ktorú dnes využívajú takmer všetky aplikácie, ktoré komunikujú cez sieť, obzvlášť internet. Užívatelia používajúci internet si väčšinou neuvedomujú že využívajú služby DNS a aké riziká to so sebou prináša.

Prvá implementácia DNS vznikla v roku 1983, ktorej špecifikácia bola zverejnená v RFC 882 a RFC 883. V roku 1987 bola upravená a špecifikácia, ktorá je platná dodnes sa nachádza v RFC 1034[11] a 1035[12]. Tento návrh DNS bol zameraný na to aby bol systém distribuovaný a škáľovatelný, ale nebol braný ohľad na bezpečnosť. Dodnes samotné DNS neposkytuje možnosť zabezpečenia systému a vytvára priestor pre útoky, ktoré nezabezpečenie využívajú. Príkladom takéhoto útoku je *phising* (podvrhnutie falošnej internetovej stránky, za účelom získať citlivé informácie od návštevníkov).

Z tohto dôvodu bolo navrhnuté rozšírenie systému DNS, ktoré rozširuje možnosti DNS a podporuje zabezpečenie v podobe digitálneho podpisovania záznamov v DNS databázy - *DNSSEC (Domain Name System Security Extensions)*[13].

1.1 Motivácia

DNSSEC je technológia, ktorá je v súčasnosti nasadzovaná do praxe. Toto nasadzovanie je postupné a dokonca niektoré organizácie odmietajú použiť DNSSEC. Dôvody prečo niektoré organizácie odmietajú použiť DNSSEC budú spomenuté v tejto práci. Motiváciou tejto práce je poskytnúť prehľad o súčasnom stave nasadenia technológie DNSSEC v praxi.

1.2 Cieľ práce

Cieľom tejto práce je vytvoriť nástroj, ktorý sleduje nasadenie technológie DNSSEC v praxi a jeho nastavenie. Webová mapa, ktorá bude zobrazovať domény zabezpečené pomocou technológie DNSSEC. Webová aplikácia bude slúžiť hlave administrátorom domén a dávať

im prehľad o súčasnom stave. Na druhú stranu môže byť motiváciou pre subjekty, ktoré ešte nepoužívajú zabezpečenie DNS pomocou technológie DNSSEC.

Výsledná webová mapa je vytváraná pre organizáciu *CZ.NIC* a bude mapovať nasadenie technológie DNSSEC medzi TLD doménami a SLD doménami `.cz`. Výsledok tejto práce by mal byť univerzálny a teda použiteľný aj pre iné SLD domény ako `.cz`.

1.3 Štruktúra dokumentácie

V prvej časti si položíme teoretický základ pre prácu s technológiou DNSSEC. V stručnosti vysvetlíme základy systému DNS a podrobne popíšeme aké rozšírenia zavádza do systému DNS zabezpečenie DNSSEC. V poslednej časti kapitoly o DNSSEC popíšeme akým spôsobom DNSSEC funguje a akým spôsobom prebieha overovanie pravosti záznamov v DNS.

Druhá časť práce sa zaoberá možnosťami získavania informácií potrebných pre vytvorenie webovej mapy. Popíšeme si služby ktoré sú potrebné k získaniu všetkých potrebných informácií na vytvorenie mapy a nástroje, pomocou ktorých je možné získavať informácie z týchto služieb.

Tretia časť práce sa zaoberá návrhom aplikácie. Zadefinujeme informácie, ktoré sú potrebné k vytvoreniu aplikácie a principiálne popíšeme postupy na získavanie dát pre vytvorenie webovej mapy.

V štvrtej časti sa budeme zaoberať samotnou implementáciou aplikácie. Uvedieme konkrétne postupy na získavanie informácií a implementujeme opatrenia, aby sa zamedzilo zneužitiu systému na získanie citlivých informácií.

Piata časť je venovaná testovaniu vytvoreného systému. Testovať budeme základnú funkcionalitu a zabezpečenie proti zneužitiu. V závere tejto časti zhodnotíme systém ako celok.

Kapitola 2

DNSSEC

V dobe keď systém DNS vznikal, bol počet užívateľov internetu nízky a tak hrozba zneužitia nebola až taká vysoká. DNS vo svojej pôvodnej implementácií (RFC 1034[11] a RFC 1035[12]) neobsahovalo žiadne bezpečnostné opatrenia proti zneužitiu a bolo navrhnuté hlavne ako škálovatelný a distribuovaný systém.

V súčasnosti sú bezpečnostné rizika vyššie ako kedysi a na preklad doménových mien pomocou DNS sa spolieha väčšina aplikácií komunikujúcich cez internet. Aplikácie v sebe zahŕňajú rôzne formy zabezpečenia, ale spoliehajú na DNS, ktoré ochranu proti falšovaniu údajov získaných z DNS neposkytuje. Túto ochranu poskytuje až rozšírenie DNS a to DNSSEC (Domain Name System Security Extensions). Cieľom DNSSEC je rozšíriť pôvodné DNS o mechanizmy, aby bol DNS klient schopný overiť, či poskytnuté informácie z DNS sú pravdivé a môže im dôverovať. Je dôležité poznamenať, že DNSSEC chráni len údaje uložené v databázy DNS a nestará sa o to ako sú informácie prenášané. DNSSEC neposkytuje žiadne obmedzenie prístupu, autentizáciu užívateľov, alebo šifrovanie odpovedí z DNS. Poskytuje len overenie pravosti získaných údajov z databázy DNS.

V prvej časti tejto kapitoly si v skratke vysvetlíme základy DNS a zadefinujeme pojmy, ktoré budeme neskôr používať. V druhej časti si podrobnejšie popíšeme aké rozšírenia DNSSEC prináša do súčasného DNS, teda pozrieme sa na nové typy záznamov v DNS, ktoré implementuje DNSSEC a vysvetlíme si základné princípy jeho fungovania.

2.1 Základy DNS

DNS (Domain Name System) je hierarchický systém, postavený na distribuovanej databázy, ktorý udržuje informácie spojené s doménami. Hlavnou funkciou je preklad doménového mena na IP adresu. Poskytuje ale aj opačný preklad (IP adresa na doménové meno), či adresu mailového serveru pre danú doménu ([6]). V tejto práci sa nezaoberáme systémom DNS, ale jeho rozšírením DNSSEC, preto si vysvetlíme len veci potrebné k práci s DNSSEC. Ďalšie informácie ohľadom DNS je možné nájsť v RFC [12].

2.1.1 Zóna

Menný priestor DNS je definovaný v [11] a je reprezentovaný ako strom doménových mien. Koreň tohto stromu je najvyššia doména označovaná pomocou bodky .. Zóna je podstrom doménového stromu, ktorého administrácia bola delegovaná na iný DNS server (jedná sa o distribuovaný systém).

2.1.2 RR - Resource records

Celá DNS databáza pozostáva z jednoduchých záznamov - *Resource records (RR)*. Každý záznam RR pozostáva z následných častí:

- Meno vlastníka (názov domény, ku ktorej záznam patrí).
- Typ záznamu.
- Trieda záznamu.
- Doba po ktorú môže byť záznam uložený v cache pamäti.
- Dĺžka dát v zázname.
- Dáta záznamu, sú označované ako RDATA.

Viacej informácii o RR sa dočítame v [12]

2.1.3 Ako DNS funguje

DNS systém používa pre komunikáciu model *klient-server*. Klient sa svojho DNS serveru dotáže a ten odpovie (dotaz/odpoveď). V systéme DNS rozoznáva 4 typy serverov :

- Primárny name server udržuje dáta o svojej zóne v databázy na disku. Zmeny v databázy ma zmysel robiť len na primárnych name serveroch. Primárne servery sú autoritou pre svoje domény, tzn. ich dáta pre príslušnú zónu sa považujú za autoritatívne.
- Sekundárny name server si v pravidelných časových intervaloch kopíruje informácie o zóne z primárneho servera. Preto nemá význam robiť akékoľvek zmeny na sekundárnom servery, lebo budú prepísané pri následnom kopírovaní. Informácie o zóne poskytnuté od sekundárneho servera, sú rovnako autoritatívne ako od primárneho.
- Root name server je name server, ktorý obsahuje informácie o root doméne. V súčasnosti je celkovo 13 root serverov rozmiestnených po celom svete. Každý root server je primárny name server.
- Caching only server nie je autoritou pre žiadnu zónu (nie je primárny, ani sekundárny name server). Dáta, ktoré ním prechádzajú si dočasne ukladá vo svojej pamäti. Ak príde dotaz, na ktorý sa v jeho pamäti nachádza odpoveď, tak ju poskytne, ale ako neautoritatívnu. Túto vlastnosť, má každý name server, akurát *caching only server* neudržuje lokálne žiadne informácie o zónach, okrem tých, ktoré ním prechádzajú.

Systém DNS používa model klient-server. Typy serverov sme si popísali v predchádzajúcej časti. Na strane klienta sa nachádza takzvaný *DNS resolver*. Ten je zodpovedný za vytvorenie a odoslanie dotazu na server DNS. Server DNS môže vyriešiť dotaz dvoma spôsobmi:

- *Rekurzívne* - v tomto prípade klient (*DNS resolver*) odošle svoj dotaz na server DNS. Server DNS vykoná kompletne celý preklad a klientovi je predaný až záverečný výsledok. Dotazovanie ostatných server DNS, ktoré sú potrebné na získanie odpovede na dotaz, je v réžií serveru DNS, ktorému bol zaslaný doraz. Tento spôsob je pre server výpočetne náročnejší. Preto po každom dotaze, je odpoveď uložená v pamäti cache a odpoveď na ďalší rovnaký dotaz je braná z cache pamäti.

- *Nerekurzívne* - pri nerekurzívnom vykonávaní dotazu, je dotazovanie ostatných serverov na strane klienta. Server DNS, ktorému bol zaslaný dotaz, len odpovie kde má (*DNS resolver*) hľadať ďalšie informácie k úspešnému vykonaniu dotazu. Tento spôsob je výpočetne menej náročný ako rekurzívne vybavovanie dotazov.

Zvyčajne servery DNS, ktoré používajú rekurzívny spôsob, sú tie ktoré sú najbližšie k užívateľovi ktorý využíva služby DNS. Vybavenie dotazu je výpočetne náročnejšie a výsledky sú ukladané do pamäti cache. Servery DNS, ktoré sú vyššie v hierarchii, používajú nerekurzívny spôsob. Teda v odpovediach poskytujú len informácie, kde sa má ďalej hľadať. Len v prípade ak poznajú odpoveď na dotaz, tak pošlú odpoveď.

Pri preklade domény na IP adresu, sa server najprv pozrie do svojej cache pamäti. Pokiaľ v nej nájde odpoveď, tak odpovie. Ak v cache nenájde odpoveď, tak začne prehľadávať menný priestor DNS a začína od root name serverov. Preklad domény `fit.vutbr.cz` nám ilustruje obrázok 2.1.

1. Klient odošle dotaz na preklad domény `fit.vutbr.cz` na IP adresu svojmu serveru DNS (adresu serveru DNS má klient nastavenú, alebo pridelenú z DHCP). Ak server pozná odpoveď na dotaz, alebo odpoveď na tento dotaz je uložená v pamäti cache, tak server klientovi pošle odpoveď.

2. Ak server nepozná odpoveď musí ju zistiť. Každý server DNS má vo svojej konfigurácii uložené adresy *root* serverov, ktoré obsahujú informácie o doménach typu *TLD*. Server DNS, pošle dotaz na *root* server. Ten odpovie, že odpoveď nepozná, ale pošle adresu servera DNS, ktorý ma informácie o doméne `cz`.

3. Následne je serveru, ktorý má informácie zaslaný dotaz. Ten odpoveď nepozná a posiela informácie o tom kde sa má ďalej hľadať. V odpovedi je addresa serveru DNS, ktorý má informácie o doméne `vutbr.cz`.

4. Dotaz je poslaný na server DNS, ktorý ma informácie o doméne`vutbr.cz`. Ten pozná odpoveď na dotaz a tak posiela odpoveď s požadovanými informáciami.

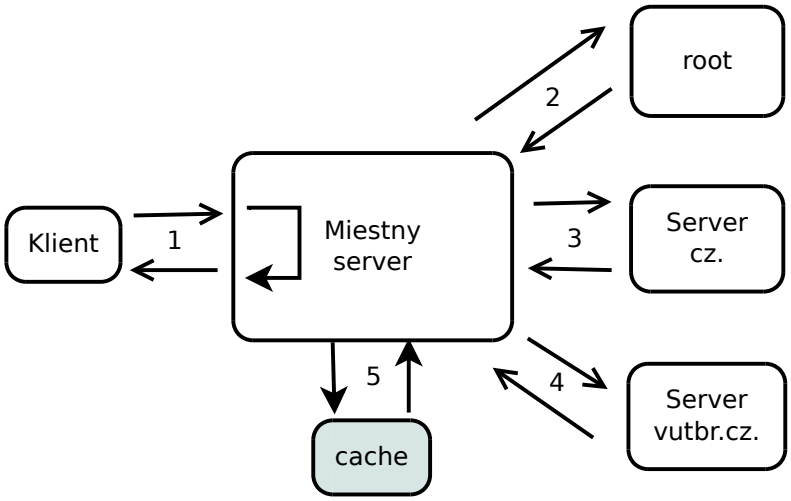
5. Klientov server DNS obdržal odpoveď na klientov dotaz. Odpoveď je odoslaná klientovi a uložená do pamäti cache. Ak by prišiel znova rovnaký dotaz, tak by prebehol len krok 1.

Podrobnejšie informácie o fungovaní a princípoch DNS nájdete v [6], alebo v RFC [12] a [11].

2.2 Hrozby v DNS

Keďže DNSSEC je rozšírenie, ktoré prináša do DNS prvky zabezpečenia, popíšeme akým spôsobom sa dá napadnúť nezabezpečené DNS. Najčastejší a najnebezpečnejší spôsob zneužitia DNS je *DNS cache poisoning*.

Ako bolo spomenuté v kapitole 2.1.3, každý DNS server si dočasne ukladá dáta, ktoré ním prechádzajú, do svojej lokálnej cache pamäti. Ak mu príde dotaz, na ktorý sa odpoveď nachádza v cache, tak túto informáciu z cache použije a pošle ju naspäť klientovi. Originálna implementácia DNS neumožňuje overenie zdroja, od ktorého dáta prišli. Táto skutočnosť dáva priestor útočníkom.



Obrázek 2.1: Preklad domény fit.vutbr.cz.

Princíp tohto útoku je, že útočník podvrhne falošné údaje o danej doméne do cache pamäti servera, alebo klienta (O spôsoboch ako podvrhnúť falošné informácie sa môžeme viac dočítať v [16]). Pred každým dotazom sa nahliadne do cache pamäti, či sa hľadaná informácia v nej nenachádza. Ak vyhľadávanie v cache je úspešné, tak dotazovanie ďalších serverov neprebieha a informácia uložená v pamäti cache sa poskytne klientovi. Pokiaľ sa v pamäti cache nachádza falošná informácia, podvrhnutá útočníkom, tak práve táto je poskytnutá klientovi. Takto útočník dokáže svoju obeť, bez jej vedomia, presmerovať na stanicu s inou IP adresou, akoby užívateľ očakával. Tento útok je základom iných útokov ako napr. *phising*. Cieľom tohto útoku je užívateľa dostať na falošnú internetovú stránku, ktorá vyzerá identicky so skutočnou. Jej cieľom je získať od užívateľa citlivé informácie, ako napríklad prihlasovacie údaje.

Proti tomuto typu útoku v systéme DNS neexistuje účinná ochrana. Až rozšírenie DNSSEC implementuje do DNS mechanizmy, ktoré dokážu úspešne odvrátiť hrozbu *DNS cache poisoning* pomocou digitálneho podpisovania záznamov RR v DNS.

2.3 Rozšírenie záznamov RR

Cieľom Domain Name System Security Extenstion (DNSSEC) je definovať mechanizmy, ktoré klientovi umožnia overiť, či informácie ktoré dostal z DNS sú pravdivé. Druhou úlohou je poskytnúť overenú informáciu o tom, že požadované dáta sa v DNS nenachádzajú.

Tieto mechanizmy vyžadujú zaviesť určité zmeny do DNS protokolu. DNNSEC pridáva do DNS niekoľko nových záznamov RR (resource records):

- DNSKEY
- RRSIG
- DS
- NSEC
- NSEC3

DNSSEC rozširuje aj hlavičku DNS správy o nové byty, aby systém DNS mohol informovať klienta, či poskytnuté údaje sú overené a aby klient bol schopný v dotaze indikovať, že vyžaduje autentizované dáta.

DNSSEC je v praxi nasadzované postupne, preto je dôležité, aby bolo spätne kompatibilné s DNS.

2.3.1 Záznam DNSKEY

DNSSEC používa asymetrickú kryptografiu na podpisovanie a overovanie záznamov RR. Privátny kľúč je uložený na servery, pokiaľ možno mimo DNS. A verejný kľúč je uložený práve v zázname DNSKEY.

Klient tak môže použiť tento kľuč z DNSKEY na overenie podpisov, ktorými sú podpísane iné záznamy (napr. A záznam) v zóne a tak overiť ich pravosť.

RDATA pre DNSKEY záznam obsahujú následné položky:

- **Príznaky** - pole príznakov má dĺžku 16 bitov. V súčasnosti sa používajú len dva, bit 7 a 15. Ostatné sú rezervované a musia byť nastavené na 0.

  Bit 7 označuje *Zónový kľúč*. Ak je tento byt nastavený na 1, potom DNSKEY záznam obsahuje kľúč na podpisovanie záznamov v danej zóne a vlastník tohto záznamu musí byť názov zóny.

- **Protokol** - pole pre protokol je veľkosti 8 bitov. Toto pole musí mať vždy hodnotu 3. Ak je jeho hodnota iná ako 3, tak podľa RFC [15] je záznam DNSKEY považovaný ako nevalidný pri overovaní podpisov.

- **Algoritmus** - 8 bitové pole, ktoré určuje aký kryptografický algoritmus bol použitý pri vytváraní verejného kľúča a v akom formáte je kľúč uložený.

- **Kľúč** - toto pole obsahuje samotný verejný kľúč na overenie podpísaných záznamov. Formát kľúča závisí od algoritmu, akým bol kľúč vytvorený.

Následný príklad nám ukazuje záznam DNSKEY pre doménu `fit.vutbr.cz`.

```
fit.vutbr.cz.          14400 IN DNSKEY 256 3 5 (
                       AwEAAcyccLng09xZedVqpf/0DkSqLwngQXS1e9FeOdV6
                       D4hcrljjzHMlsD5l7UCdcXdnyJnS6UZwHlV/l2U4v7jn
                       ii4UB7GEi4pbCVdH1xToOxCg5osszn0wBK4tJ2AmISm/
                       z4IfBwx9l2fCO2Y8Y/v0pK5jWZ4KjPwpUsd3TR5EYudD
                      ) ; key id = 35421
```

Prvé štyri polia určujú vlastníka, TTL, triedu a ty záznamu RR (v tomto prípade DNSKEY). Hodnota 256 indikuje že je nastavený bit *zónového kľúča* (bit 7). Hodnota 3 je pevná pre protokol a hodnota 5 určuje šifrovací algoritmus. V tomto prípade 5 označuje algoritmus RSA/SHA1. Ostatný text je verejný kľúč zakódovaný pomocou *Base64*. Na konci sa nachádza ešte jeden udaj a to id kľúča, alebo *key tag*. Je to 16-bitové číslo, ktoré slúži k rýchlejšej identifikácií kľúča.

2.3.2 Záznam RRSIG

DNSSEC pomocou asymetrickej kryptografie overuje pravosť záznamov v DNS. Verejný kľúč pre overenie, ako bolo spomenuté v kapitole 2.3.1, je uložený v zázname typu DNSKEY.

Digitálne podpisy samotných záznamov RR, sú uložené v zázname typu RRSIG. Aby sme boli presnejší, tak RRSIG nepodpisuje samotné záznamy RR, ale RR sety. RR set je množina záznamov, ktoré majú rovnaké meno, triedu a typ. Napríklad pre jednu doménu môže byť DNS obsahovať viacej MX (adresy mailových serverov) záznamov. Všetky tieto MX záznamy sú podpísané práve jedným záznamom RRSIG.

Záznam RRSIG obsahuje podpis pre RR set s určitým menom, triedou a typom. Určuje dobu platnosti podpisu, algoritmus použitý pri podpise, meno podpisujúceho a id kľúča (2.3.1). Id kľúča identifikuje záznam DNSKEY, ktorý obsahuje verejný kľúč, ktorý môže klient použiť na overenie podpisu. Slúži hlavne na zrýchlenie hľadania záznamu DNSKEY, ktorým bol príslušný RRSIG vytvorený.

Pretože každý RR set v zóne musí byť ochránený digitálnym podpisom, tak RRSIG záznam musí byť prítomný aj pre záznam typu CNAME. Toto je zmena od originálnej implementácie DNS [11]. V tej je uvedené, že ak je pre nejaké meno prítomný CNAME záznam, tak CNAME je jediný typ povolený pre dané meno. V DNSSEC záznam typu RRSIG musí byť prítomný aj pre záznam CNAME.

RDATA pre DNSKEY záznam obsahujú následné položky:

- **Typ záznamu** - záznam podpísaný pomocou RRSIG.
- **Algoritmus** - označuje algoritmus, ktorý je použitý pri vytváraní podpisu. Číslom 5 je označený algoritmus RSA/SHA1. Každá implementácia DNNSEC je povinná podporovať RSA/SHA1.
- **Label field**
- **Doba životnosti podpisovaného záznamu** - je to hodnota ktorá je uvedená pri tomto zázname v autoritatívnej zóne.
- **Doba platnosti podpisu** - TTL
- **Dátum ukončenia platnosti podpisu**- RRSIG zázname nesmie byť použitý pre overenie záznamu, ak mu skončila platnosť. Tento dátum je reprezentovaný ako 32-bitové číslo označujúce počet sekúnd od 1.1.1970 00:00:00 UTC. V zónovom súbore je reprezentovaný reťazcom vo formáte *RRRRMMDDhhmmss*.
- **Dátum zahájenia platnosti podpisu** - je reprezentovaný a používaný v zónovom súbore rovnako ako dátum ukončenia platnosti.
- **Id kľúča** - odkazuje do záznamu DNSKEY. Odkazuje na kľuč, ktorý overuje tento podpis. Id kľúča slúži na rýchlejšie nájdenie kľúča pre overenie a nie je povinné.
- **Meno podpisujúceho** - označuje vlastníka záznamu DNSKEY, ktorý má klient použiť pri overovaní podpisu. Meno podpisujúceho musí obsahovať názov zóny, do ktorej patrí záznam ktorý je podpísaný týmto záznamom RRSIG.
- **Podpis záznamu** - je vo formáte, ktorý závisí od algoritmu, ktorým bol podpis vytvorený a v kódovaní *Base64*.

Následný príklad ukazuje podpis A záznamu pre dotaz na server s adresou www.nic.cz.

```
www.nic.cz.             1800 IN A 217.31.205.50
www.nic.cz.             1800 IN RRSIG A 5 3 1800 20110118140302 (
```

```
                                    20110104140302 3362 nic.cz.
                                    k1/XscRQZkVygOEFhr6cgqk2vohuFtGMmo8GItwbtwZq
                                    DSjlIOWPrSom5FP4GKxDrbs8s46mAMxO6WyUQV9CX+F0
                                    t2aCtmKdPZnnKQRu+owVBaMM5nhkondRoFcOPmUScKZI
                                    MZrz7n8KQWVxmWeV/AafVZzn3OTYrwsDQGmsEzo= )
```

2.3.3 Záznam DS

Záznam typu DS je spojený s DNSKEY a je použitý pri overovaní záznamu DNSKEY. Záznam DS je spojený s DNSKEY pomocou id kľúča, typu algoritmu a kontrolným súčtom kľúča uloženého v DNSKEY. Kontrolný súčet by bol dostatočný na spojenie záznamu DS a DNSKEY, ale pomocou id kľúča a typu algoritmu je toto spárovanie rýchlejšie.

Záznam DS a DNSKEY záznam majú rovnakého vlastníka, ale sú uložené v rôznych lokalitách. Záznam DS je uložený v zóne o úroveň vyššie ako DNSKEY záznam ktorý overuje. Napríklad záznam DS pre doménu `fit.vutbr.cz` je uložený v zóne `vutbr.cz`. K nemu príslušný DNSKEY záznam je uložený v zóne `fit.vutbr.cz`. RDATA pre záznam DS obsahujú následné položky:

- **Id kľúča** - toto id je identické s id kľúča použitého v zázname RRSIG a slúži na spojenie s kľúčom uloženým v zázname DNSKEY.
- **Typ algoritmu** - algoritmus použitý k vytvoreniu podpisu. Je identický s typom algoritmu uvedeného v záznamoch typu RRSIG a DNSKEY.
- **Typ algoritmu** - algoritmus použitý na vytvorenie kontrolného súčtu z kľúča v DNSKEY.
- **Kontrolný súčet** - súčet vytvorený zo záznamu DNSKEY. Pre kontrolný súčet je použitý vlastník DNSKEY záznamu a RDATA DNSKEY záznamu. Výpočet kontolného súčtu vyzerá následne (Znak + značí zreťazenie) :

```
hash = hash_algoritmus(DNSKEY vlasník + DNSKEY RDATA);
DNSKEY RDATA = FLAGY + PROTOKOL + ALGORITMUS + KĽÚČ
```

Následný príklad ukazuje záznam DNSKEY a k nemu príslušný DS záznam.

```
fit.vutbr.cz. 86400 IN DNSKEY 256 3 5 ( AQOeiiR0GOMYkDshWoSKz9Xz
                                        fwJr1AYtsmx3TGkJaNXVbfi/
                                        2pHm822aJ5iI9BMzNXxeYCmZ
                                        DRD99WYwYqUSdjMmmAphXdvx
                                        egXd/M5+X7OrzKBaMbCVdFLU
                                        Uh6DhweJBjEVv5f2wwjM9Xzc
                                        nOf+EPbtG9DMBmADjFDc2w/r
                                        ljwvFw==
                                        ) ;  key id = 60485

fit.vutbr.cz. 86400 IN DS 60485 5 1 ( 2BB183AF5F22588179A53B0A
                                       98631FAD1A292118 )
```

2.3.4 Záznam NSEC

Doteraz spomenuté záznamy sa týkali hlavne podpisovania záznamov existujúcich v DNS databázy. Pokiaľ proces prekladu zlyhá, to znamená že požadovaná doména sa v DNS databázy nenachádza, tak v DNS odpovedi je nastavený návratový kód `RCODE` na hodnotu `NXDOMAIN`. Správa, ktorá neobsahuje žiadne dáta nemôže byť ani podpísaná. Preto DNSSEC zavádza záznam typu NSEC, ktorý poskytuje overiteľnú odpoveď o neexistencii domény.

Záznam NSEC obsahuje názov ďalšieho vlastníka, ktorý obsahuje overiteľné dáta, alebo deleguje na iný server pomocou NS záznamu. K tejto informácií poskytuje ešte zoznam typov záznamov, ktoré sú dostupné pre daného vlastníka. Záznamy NSEC týmto spôsobom vytvárajú reťazec všetkých autoritatívnych vlastníkov v zóne. Pomocou takto vytvoreného reťazca sa dá overiť, že požadovaná doména neexistuje a poskytnúť o tom overiteľnú odpoveď. Pokiaľ klient požiada o vyhľadanie neexistujúcej domény, tak mu je vrátený NSEC záznam, ktorý obsahuje abecedne najbližšiu doménu, pre ktorú existujú dáta.

Skutočnosť, že klient pri neexistencii domény dostane NSEC záznam, ktorý obsahuje ďalšiu najbližšiu doménu, otvára priestor pre ďalšie zneužitie. Tým je možnosť získania kompletného obsahu zóny.

Ak by klient poslal dotaz na preklad domény `ucebna.fit.vutbr.cz`, ktorá neexistuje, tak ako odpoveď dostane NSEC záznam pre meno `ucebna-301.fit.vutbr.cz`, v ktorom sa dozvie, že za touto doménou nasleduje `ucebna-304.fit.vutbr.cz`. Z tohto síce vie overiť, že požadovaná doména `ucebna.fit.vutbr.cz` skutočne neexistuje, ale zároveň takýmto spôsobom je schopný prečítať celý obsah zóny.

Existencia tohto spôsobu prečítania obsahu celej zóny spôsobila, že niektoré inštitúcie odmietli implementovať DNSSEC. Skutočnosť, že je možné zistiť obsah ich domény bola vnímaná, ako veľké bezpečnostné riziko. Preto vznikol novší typ záznamu NSEC3, ktorý je predmetom ďalšej časti tejto práce.

RDATA pre NSEC záznam obsahujú následné položky:

- **Meno následnej domény** - doména ktorá obsahuje dáta, alebo deleguje na ďalší server.
- **Bitová mapa** - identifikuje záznamy, ktoré sú prítomne v zóne pre danú doménu.

Následný príklad ukazuje NSEC záznam pre doménu `ucebna-301.fit.vutbr.cz`:

```
ucebna-301.fit.vutbr.cz. 86400 IN NSEC ucebna-304.fit.vutbr.cz. (
                                    A RRSIG NSEC )
```

2.3.5 Záznam NSEC3

Záznam NSEC3 vychádza z novšej špecifikácie [1]. Slúži na rovnaký účel ako NSEC, na overenie neexistencie domény v systéme, ale neumožňuje získať informácie o celej zóne ako predchádzajúci záznam NSEC.

NSCE3 funguje na podobnom princípe ako NSEC. Vytvára reťazec domén v zóne, pre ktoré existujú záznamy. Na základe tohto reťazca je klient schopný overiť, že doména na ktorú sa pýtal skutočne neexistuje. Na rozdiel od NSEC, NSEC3 nepoužíva názov domény, ale jej hash. To znamená, že reťazec nie je pospájaný pomocou názvov domén, ale pomocou ich hashov. Toto zabezpečuje, že už nie je možné pomocou NSEC3 záznamu zistiť ďalšiu doménu v zóne.

Použitie NSEC3 je náročnejšie na výpočet oproti NSEC. Pri dotaze na neexistujúcu doménu je potreba vyrátať hash z domény, ktorá je obsiahnutá v dotaze. To znamená, že

pre každý dotaz na NSEC3 sa počíta hash domény. Tu vzniká hrozba útoku typu DoS, kedy je na server zaslané veľké množstvo dotaz na NSEC3 s neexistujúcimi doménami. V praxi sa to rieši tak, že dotazy na NSEC3 majú nižšiu prioritu ako ostatné.

RDATA pre NSEC3 záznam obsahujú následné položky:

- **Algoritmus** - algoritmus použitý na vytvorenie hashu.
- **Príznaky** - príznaky použité v NSEC3. V špecifikácii [1] je zadefinovaný len 8-mi bit (celkovo 8 bitov). Ten je použitý pre Opt-Out príznak. Ostatné musia byť nastavené na 0.
- **Počet iterácií** - počet iterácií pri výpočte hashu. Čím viac iterácií bolo použitých, tým je NSEC3 záznam odolnejší proti slovníkovým útokom.
- **Dĺžka reťazca** - dĺžka reťazca, ktorý je pridaný k doméne pred počítaním hashu.
- **Reťazec** - je pridaný k doméne pred počítaním hashu. Zvyšuje ochranu proti slovníkovým útokom.
- **Dĺžka hashu**
- **Hash následnej domény v zóne**
- **Bitová mapa** - identifikuje záznamy, ktoré sú prítomne v zóne pre danú doménu.

Následný príklad ukazuje NSEC3záznam pre doménu cz:

```
L30GJ03SLD0V0KNMPKRE262BLN2V6G2Q.cz.  900 IN NSEC3 1 0 10 89A9724CA3554264
                                      L30H4U4CEFVA4KVR903VH28P9EVJR9JS
                                      NS SOA RRSIG DNSKEY NSEC3PARAM
```

2.3.6 Záznam NSEC3PARAM

Záznam typu NSECPARAM je špecifikovaný v RFC 5155 [1]. 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. Ak sa v zóne nachádza záznam NSECPARAM s príznakom nastaveným na hodnotu 0, tak pre každé doménové meno je záznam NSEC3 vytvorený rovnakým algoritmom, používa rovnaký počet iterácii pri vytváraní hashu a rovnaký reťazec bol pridaný k doméne pred počítaním hashu. To znamená, že zóna musí obsahovať záznamy NSEC3, ktoré boli vytvorené s rovnakými parametrami.

Záznam NSECPARAM je používaný len autoritatívnymi servermi, pre danú doménu a nie je používaný pri overovaní podpísaných záznamov.

RDATA pre NSECPARAM sú zhodné s prvými 4 položkami ako pri zázname NSEC3.

- **Algoritmus** - identifikuje kryptografický algoritmus použitý pri vytváraní hashu v .
- **Pole príznakov** - Opt-Out príznak je nastavený na hodnotu 0. Ostatné príznaky sú rezervované pre budúce použitie a sú nastavené na hodnotu 0.

- **Počet iterácií** - čím viac iterácií bolo použitých, tým je NSEC3 záznam odolnejší proti slovníkovým útokom.
- **Dĺžka reťazca** - dĺžka reťazca, ktorý je pridaný k doméne pred počítaním hashu.
- **Reťazec** ktorý bol pridaný k doméne pred počítaním hashu. Zvyšuje ochranu proti slovníkovým útokom.
- **Dĺžka hashu**

Následný príklad ukazuje NSEC3PARAM záznam pre doménu `cz`:

```
cz.             0 IN NSEC3PARAM 1 0 10 89A9724CA3554264
```

2.4 Ako funguje DNSSEC

DNSSEC rozširuje DNS o zabezpečenie záznamov a overenie platnosti týchto záznamov. Zabezpečenie záznamov je implementované ich digitálnym podpisovaním. Pri podpisovaní je použitá asymetrická kryptografia. To znamená použitie súkromného a verejného kľúča. Na vytvorenie digitálneho podpisu pre záznam je použitý súkromný kľuč a na overenie tohto podpisu je použitý verejný kľúč, ktorý je každému užívateľovi verejne dostupný.

Každý administrátor zóny si vytvorí dvojicu kľúčov. Súkromný kľúč a k nemu príslušný verejný kľúč. Súkromný kľuc ma uložený v tajnosti (súkromný kľúč je odporúčané držať mimo sieť) a verejný kľúč je dostupný z DNS. Je uložený v zázname typu DNSKEY.

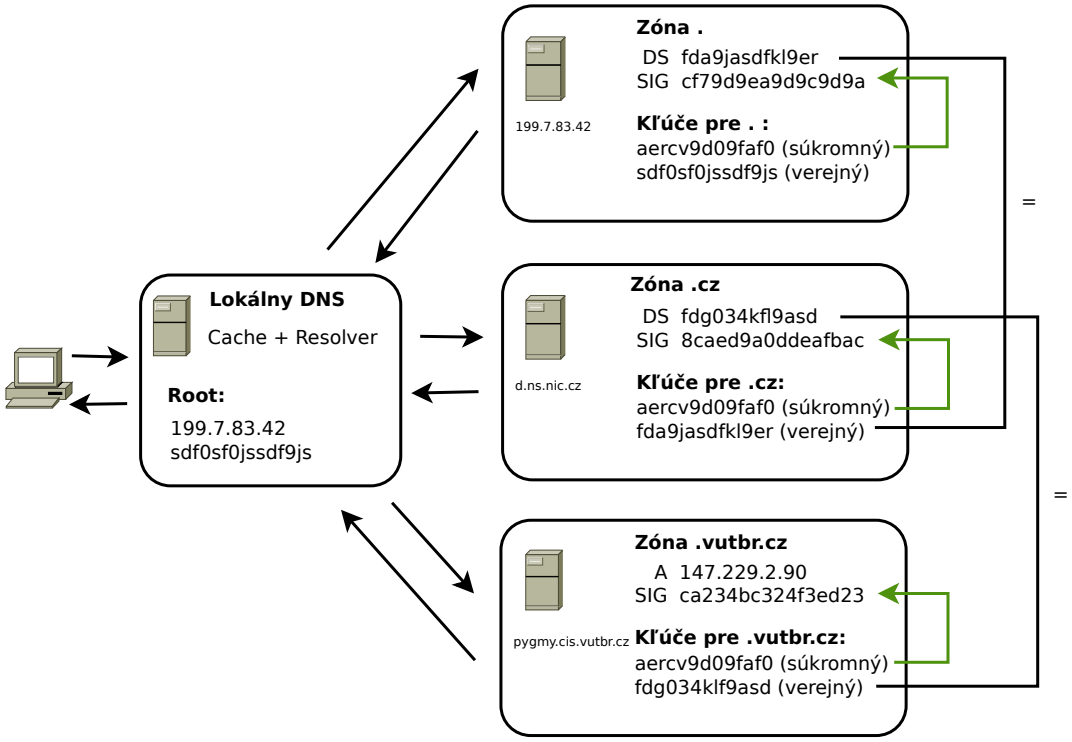
V DNSSEC rozoznávame dva typy verejných kľúčov:

- ZSK (Zone signing key) - kľúč slúžiaci na podpisovanie záznamov v zóne.
- KSK (Key signing key) - kľuč slúžiaci na podpisovanie kľúčov(ZSK). Tento kľúč je uložený je overený u vyššej autority.

ZSK môžu byť menené pomerne často (sú v správe administrátora domény), preto sú kratšie a slabšie ako KSK, aby overovanie netrvalo zbytočne dlho. Naopak KSK je silnejší a dlhší kľúč. Musí byť schválený od vyššej autority. Tento proces schvaľovania je zložitejší ako obmena kľúčov v rámci zóny. Preto majú KSK dlhšiu životnosť a sú silnejšie. Použitie ZSK a KSK nie je nutnosťou, ale je to odporúčané riešenie. Môže existovať len jeden kľúč, ktorý podpíše sám seba.

Tieto kľúče vytvárajú *reťazec dôvery*. Podpis záznamov v zóne je overiteľný pomocou ZSK. ZSK sú overiteľné pomocou kľúča KSK a ten je overený v zóne o stupeň vyššie. Pre lepšie pochopenie si uvedieme príklad. Záznamy v doméne `vutbr.cz` sú podpísane pomocou súkromného kľúča, ktorý drží správca v tajnosti. Tieto podpisy idú overiť pomocou ZSK (párový kľúč k súkromnému kľúču), ktorý je uložený priamo v doméne `vutbr.cz`. ZSK je tak isto podpísaný a jeho podpis je možné overiť pomocou KSK. ZSK aj KSK sú uložené v zázname DNSKEY v `vutbr.cz`. KSK je ale možné overiť až v doméne `cz`. Tam sa nachádza záznam DS, pomocou ktorého je možné overiť KSK vo `vutbr.cz`.

Ako sa vytvára reťazec dôvery si popíšeme na príklade, ktorý ilustruje obrázok 2.2. Klient obdžal zo systému DNS odpoveď na žiadosť o preklad domény `www.vutbr.cz`. Postup overovania pravosti obdržaných informácií bude následovný:



Obrázek 2.2: Príklad podpisovania kľúčov a reťazca dôvery.

1. DNS resolver má nakonfigurovaný ako bod dôvery koreňový server s kľúčom `sdf0sf0jssdf9js`. DNS resolver v prvom kroku overí platnosť získaného záznamu na servery DNS, ktorý má na starosti zónu `vutbr.cz`. V tomto prípade ide o záznam typu `A`. Na servery je tento záznam podpísaný súkromným kľúčom (zelená šípka). Tento podpis je overiteľný pomocou verejného kľúča (uložený v zázname typu `DNSKEY`).

2. Verejný kľúč, ktorý je uložený v zóne `vutbr.cz` je overiteľný v zóne `cz`. V zóne `cz` je v zázname typu `DS` uložený hash verejného kľuča pre doménu `vutbr.cz` (na obrázku pre lepšiu názornosť nepoužívame hash, ale priamo celý kľúč). Takto je overiteľný verejný kľúč pre `vutbr.cz`.

3. Tretí krok je podobný ako druhý. Opäť je verejný kľúč overiteľný pomocou jeho hashu, ktorý je uložený v koreňovej doméne pomocou záznamu `DS`. V koreňovej doméne, v zázname `DNSKEY`, je uložený verejný kľúč. Ten je zhodný s kľúčom, ktorý ma nakonfigurovaný DNS resolver, ako bod dôvery. Tak vznikol reťazec dôvery a odpoveď, ktorú dostal DNS resolver zo systému DNS je overená a považovaná za pravdivú.

Bližšie informácie o fungovaní a implementácií nájdete v RFC [13], [15] a [14]

2.5 Zhrnutie

V tejto kapitole sme sa oboznámili s novými typmi záznamov, ktoré DNSSEC zavádza do DNS a mechanizmami na overovanie platnosti záznamov v DNS databázy. DNSSEC prináša účinnú ochranu proti *DNS cache poisoning*. Na druhú stranu objem zónových súborov

narástol(o podpisy záznamov) a preklad je výpočetne náročnejší (rátanie hash a pod). Tým sa DNS stáva náchylnejšie napríklad na útoky typu *DoS* (Denial of service).

Kapitola 3

Získavanie informácií pre online mapu

Na vytvorenie webovej mapy, ktorá zobrazuje súčasný stav implementácie zabezpečenia DNS - DNSSEC, je potreba získavať informácie z externých zdrojov. V tejto časti si popíšeme dáta, ktoré sú pre nás potrebné na vytvorenie mapy a služby, z ktorých ich budeme získavať. Uvedieme si aj nástroje pomocou ktorých budeme získavať informácie z týchto externých zdrojov.

Služby, ktoré budeme sú všetky verejné a voľne dostupné, ale niektoré sú obmedzované určitými limitmi. Pre vytvorenie mapy budeme potrebovať informácie z nasledujúcich služieb :

- **DNS** - zo systému DNS budeme získavať hlavne informácie o stave zabezpečenia konkrétnych domén. V prípade zabezpečených domén budeme ďalej získavať podrobnosti o zabezpečení, ako napríklad typy použitých algoritmov pri podpisovaní.
- **Whois** - služba Whois [4] nám poskytne hlavne administratívne informácie o doménach, ako geografickú adresu vlastníka domény. Na základe tejto adresy budeme schopný umiestniť doménu na mapu.
- **Google Maps** - **Google Maps** budeme používať aj na samotné vykreslenie dát na mapu, ale ešte predtým služby od Googlu použijeme na geografickú lokáciu adries získaných zo služby Whois.

3.1 Informácie z DNS

Systém DNS nám poskytne najdôležitejšie informácie potrebné pre mapu. A to informáciu o tom, či je daná doména zabezpečená pomocou technológie DNSSEC. Z DNS budeme zisťovať aj ďalšie informácie okrem samotného zabezpečenia domény. Pôjde hlavne o informácie o nastavení DNSSEC pre danú doménu. Budeme sledovať aké algoritmy boli použité pri vytváraní podpisov a ich doby platnosti. Zaujímavou informáciou bude sledovať aj to, či daná doména používa pre poskytnutie odpovedi o neexistencii subdomény záznam NSEC, alebo NSEC3. Použitie záznamov typu NSEC, umožňuje listovanie zón, preto bude zaujímave sledovať ktoré z TLD domén, ale aj SLD, používajú ktorý typ záznamu. Práve záznam NSEC, kvôli listovaniu zón, odrádza organizácie od nasadzovania technológie DNSSEC.

Informácie ktoré sú obsiahnuté v DNSSEC, nie je možné získať z každého DNS servera. Napríklad ak náš lokálny DNS server nepodporuje DNSSEC a my pošleme dotaz, v ktorom

požadujeme zabezpečenú odpoveď na doménu, o ktorej vieme že je podpísaná pomocou technológie DNSSEC, dostaneme odpoveď, z ktorej nie je možné určiť či sa jedná o autorizovanú odpoveď. Ak pomocou nástroju *dig* sa dotážeme na doménu `vutbr.cz` lokálneho DNS serveru, ktorý nepodporuje DNSSEC dostaneme následujúcu odpoveď:

```
dig vutbr.cz NS +dnssec

;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32973
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; ANSWER SECTION:
vutbr.cz. 86400 IN NS rhino.cis.vutbr.cz.
vutbr.cz. 86400 IN NS pygmy.cis.vutbr.cz.
vutbr.cz. 86400 IN NS sloth.vutbr.net.
```

Ale ak použijeme DNS server, ktorý podporuje DNSSEC dostaneme následujúcu odpoveď:

```
dig vutbr.cz NS +dnssec @217.31.204.130

;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16779
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; ANSWER SECTION:
vutbr.cz. 86400 IN NS sloth.vutbr.net.
vutbr.cz. 86400 IN NS pygmy.cis.vutbr.cz.
vutbr.cz. 86400 IN NS rhino.cis.vutbr.cz.
vutbr.cz. 86400 IN RRSIG NS 5 2 86400 20110528135846 20110428135846 39756
        vutbr.cz. 8/RarasYb0v48HJq5nsMacfpALN8NwRSFz7R/mh4qWQGyp3+WBMCEhJ4
        KxbRQq0fgzfmV1xStHg7gm/J7pTUk/oPQrMDRIONZI+KOsmq3qQnDuoZ
        xKZ2TawEWTC472wWKmNGcmPHsGlpYSj3VVY17J2gnCKxnUcY3KMcuoVJ 75g=
```

Dotaz sme poslali na server s IP adresou `217.31.204.130`. Je to verejný server združenia CZ.NIC, ktorý podporuje DNSSEC. V hlavičke odpovedi sa už objavil príznak `ad` (*authentic data*), ktorý indikuje že ide o autorizovanú odpoveď a môžme tejto informácií dôverovať, že je pravdivá. Odpoveď už neobsahuje len záznamy `NS` ale aj podpis tejto sady záznamov `RRSIG`.

Pre našu aplikáciu bude potreba vybrať DNS server, ktorý podporuje zabezpečenie DNSSEC. Server organizácie CZ.NIC však obsahuje obmedzenia na počet dotazov v určitom časovom intervale. Preto pre získavanie informácií budeme môžme použiť server spoločnosti Google na adrese `8.8.8.8`, alebo vytvoriť si vlastný server DNS, ktorý podporuje DNSSEC.

3.1.1 Typy záznamov potrebné pre online mapu

Všetky informácie v systéme DNS sú uložené v záznamoch RR. V tejto časti si vyčleníme, ktoré budeme sledovať pohľadu našej aplikácie.

Záznam DS

Každá doména zabezpečená pomocou technológie DNSSEC musí obsahovať záznam DS. Pokiaľ by pre danú doménu neexistoval záznam DS, nebolo by možné vytvoriť reťazec dôvery a tak overiť platnosť podpisov.

Záznam DNSKEY

Verejný kľúč na overovanie podpisov je uložený v zázname RR typu `DNSKEY`. Okrem samotného kľúča sú v tomto zázname uložené aj informácie o platnosti kľúča a algoritme akým bol vytvorený. Ďalšia podstatná informácie je, o aký typ kľúča sa jedná, či o KSK, alebo ZSK. Rozdiel medzi týmto dvoma typmi je popísaný v 2.4. Podľa RFC 4043 [13] a RFC 4035 [14] nie je nutné používať oba typy kľúčov. Použitie oboch typov kľúčov je na administrátorovi domény a ich použitie zlepšuje výkonnosť pri overovaní podpisov.

Platnosť záznamu `DNSKEY` je dôležitým údajom z pohľadu aplikácie, ktorá je predmetom tejto práce. Pokiaľ vyprší platnosť kľúča a nie je predĺžená, tak aj celá táto doména sa stáva nezabezpečenou. Preto bude potrebné sledovať tieto informácie, aby bola mapa aktuálna a odpovedala skutočnému stavu DNSSECu.

Aplikácia, ktorá je predmetom tejto práce, je hlavne učená pre administrátorov domén, aby dávala prehľad o aktuálnom nasadení a nastavení DNSSECu. Základom zabezpečenia DNSSECu sú kryptografické algoritmy. Niektoré algoritmy, ktoré sú v špecifikácii DNSSECu, sa v súčastnosti už nepoužívajú a ani nie je odporúčané aby sa používali, ako napríklad algoritmus `RSA/MD5`. Aplikácii, ktorá mapuje stav DNSSECu, bude sledovať aj použité algoritmy pri vytváraní kľúčov a tak poskytne administrátorom lepší prehľad o nastavení DNSSECu.

Záznam NS

Každá doména musí obsahovať záznam `NS`. Tento záznam určuje adresu servera, ktorý pre danú doménu obsahuje autoritatívne dáta. Informácie uložené v zázname `NS` budeme ukladať a neskôr používať, ako dodatočné informácie k doméne, pri vykresľovaní na mapku.

Záznam NSEC a NSEC3

Obidva typy záznamov slúžia na poskytnutie overenej informácie o neexistencii danej domény. `NSEC` je v prvej špecifikácii DNSSEC. Neskôr sa ukázalo, že pomocou tohto záznamu je možné prečítať obsah celej zóny. Tento fakt je považovaný za bezpečnostné riziko a preto mnohé organizácie odmietli zaviesť DNSSEC práve kvôli tomuto riziku. `NSEC3` vznikol aby riešil práve problém, ktorý vzniká pri používaní záznamu `NSEC`, listovanie zón. `NSEC3` je pomerne nový a je špecifikovaný v RFC 5155 [1]. Záznam `NSEC3` je podporovaný až v najnovších verziách softwaru pre server DNS.

V aplikácií budeme sledovať aj stav týchto dvoch záznamov v DNSSECu.

RRSIG

V DNS s podporou DNSSEC musí pre každý set záznamov, záznamy rovnakého typu, existovať záznam `RRSIG`. V ňom je uložený podpis záznamov, ktorý je použitý pri overovaní pravosti záznamu. Pokiaľ by pre nejaký záznam chýbal podpis `RRSIG`, tak by nebolo možné overiť pravosť záznamu. V zázname `RRSIG` je uložená aj platnosť tohto podpisu a práve tá bude zaujímavá z pohľadu našej aplikácie.

3.1.2 Nástroje na získavanie informácií o DNSSEC

V tejto časti si uvedieme niekoľko nástrojov, ktoré budeme používať pri vytváraní aplikácie. Ako sme si uviedli, informácie o DNSSEC nie je možné získavať z akéhokoľvek servera DNS. Server DNS musí podporovať DNSSEC. Ak lokálny server DNS nepodporuje DNSSEC, je potreba použiť iný server, alebo použiť DNS resolver, ktorý funguje nerekurzívne a vie validovať podpisy záznamov.

Domain Information Groper

Domain Information Groper, skrátene *dig*, je nástroj zo softwarového balíku *BIND*. Je to software pre servery DNS. Tento nástroj nebudeme priamo používať v našej aplikácii, ale bude používaní pri implementácií na overovanie výsledkov, získaných pomocou iných nástrojov.

Domain Information Groper je hlavne nástroj na prácu s DNS. Aby podporoval DNSSEC, treba použiť parameter `+dnssec`. Viacej o použití nástroja Domain Information Groper sa dá dozvedieť v [3].

Príklad v ktorom požadujeme preklad domény `www.fit.vutbr.cz` na IP adresu a k tomuto prekladu chceme použiť server na adrese `8.8.8.8` vyzerá následovne :

```
dig www.fit.vutbr.cz A +dnssec +multiline @8.8.8.8
; <<>> DiG 9.7.3-RedHat-9.7.3-1.fc14 <<>>
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29887
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;www.fit.vutbr.cz. IN A

;; ANSWER SECTION:
www.fit.vutbr.cz. 7783 IN A 147.229.9.23
www.fit.vutbr.cz. 7783 IN RRSIG A 5 4 14400 20110612123028 (
                              20110513123028 35421 fit.vutbr.cz.
                              lcn8CBH4POlrii1S/bqnKP5Vruu9L+pwecid3uFACH+
                              1K8DFYUuoTKCBcT1fCuKvFucErlJKR5AXerLaho= )
```

Unbound

Unbound je DNS resolver vyvinutý organizáciou *NLnet Labs*. Je implementovaný v jazyku C a plne podporuje validáciu pre DNSSEC. Súčasťou softwaru Unbound je aj knižnica *libunbound*. Ide o pomerne jednoduchú knižnicu na používanie, ktorá poskytuje všetky potrebné funkcie na prácu s DNS.

Knižnica používa štruktúru typu `struct ub_ctx` na uloženie kontextu medzi volaniami jednotlivých funkcií. Na začiatku každého programu, ktorý používa funkcie z knižnice *libunbound* je teda potrebné inicializovať štruktúru `struct ub_ctx`.

Následné je možné ďalej volať funkcie z knižnice *libunbound* ako napríklad funkciu `ub_resolve`. Funkcia `ub_resolve` slúži na vytvorenie a zaslanie dotazu do systému DNS. Odpoveď na dotaz je uložená v štruktúre `struct ub_result`, ktorá obsahuje informácie jak o úspechu, alebo neúspechu, tak aj prijaté informácie zo systému DNS. Jednoduchý preklad domény `www.fit.vutbr.cz` na IP adresu vyzerá následovne:

```
struct ub_ctx* ctx;
struct ub_result* result;

ctx = ub_ctx_create();
ub_resolve(ctx, 'www.fit.vutbr.cz', 1, 1, &result);
```

Po tomto zavolaní štruktúra `struct ub_result` obsahuje výslednú IP adresu prekladu.

Knižnica *libunbound* podporuje aj overovanie platnosti záznamov získaných z DNS. Funkcia `ub_resolve` sa automaticky pokúša o overenie informácií, ktoré získala z DNS pomocou technológie DNSSEC. V predchádzajúcom prípade overenie bude neúspešné, lebo nebol nastavený bod dôvery. Ide o autoritu v reťazci dôvery, ktorej dôverujeme. Pokiaľ sa pri overovaní dostane k tejto autorite, tak odpoveď považuje za overenú.

Aby bolo možné pomocou knižnice *libunbound* overiť záznam pomocou DNSSEC je potrebné pred použitím funkcie `ub_resolve` načítať informácie o bode dôvery. V podstate ide o záznam typu `DNSKEY` domény, ktorej dôverujeme. Dôležite aby kľúč uložený v `DNSKEY`, ktorý použijeme ako bod dôvery, bol typu KSK. V opačnom prípade nebude možné overiť odpovede z DNS.

Pridanie bodu dôvery pred použitím funkcie `ub_resolve` vyzerá následovne:

```
if( (retval=ub_ctx_add_ta_file(ctx, 'ta.key')) != 0) {
        printf('error adding keys: %s\n', ub_strerror(retval));
}
```

Súbor `ta.key` môže obsahovať aj viacej bodov dôvery. Pre náš účel budeme potrebovať najvyšší bod v strome DNS a to koreňovú doménu '.'. V aplikácií budeme sledovať domény TLD a doménu `.cz`. Vystačili by sme aj len s koreňovou doménou '.' ako bodom dôvery, ale ak použijeme aj doménu `.cz`, tak overovanie domén typu SLD pre `.cz` bude rýchlejšie.

Priklad súboru `ta.key` je následovný:

```
. 86400 IN DNSKEY 257 3 8 AwEAAb5gVAzK59YHDxf/DnswfO1RmbRZ6W16JfhFecfI+E
      47t2FHaKyMMEROapL5SZ8HiCzlO5lORZGGdN37WY7fkv55rs+kwHdVRSrQdl81fUn
```

Po správnom nastavení bodu dôvery a odoslaní dotazu pomocou funkcie `ub_resolve` je knižnica *libunbound* schopná overiť, či prijatá odpoveď je správna a nebola podvrhnutá. Výsledok o overení je uložený v položke štruktúry `struct ub_result` s názvom `secure`.

Práca s knižnicou *libunbound* je pomerne jednoduchá a viacej informácií o tom ako vyzerajú jej štruktúry a aké funkcie obsahuje sa môžme dočítať v [10].

Knižnica ldns

Knižnica *ldns* je podobne ako *libunbound* implementovaná v jazyku C. Vďaka implementácií v čistom C je veľmi rýchla a optimalizovaná na čo najlepší výkon. Cieľom knižnice *ldns* je zjednodušiť programovanie aplikácií, ktoré využívajú služby systému DNS. Unbound, ktorý sme spomínali v predchádzajúcej časti, je implementovaný pomocou knižnice *ldns*.

Knižnica *ldns* je univerzálnejšia ako *libunbound*. Už len preto, že Unbound je implementovaný pomocou *ldns*. Avšak pre účel tejto práce je jednoduchšie používať knižnicu *libunbound*, lebo je jednoduchšia na použitie a dostatočná pre účely tejto práce. Na účely, kedy *libunbound* nebude dostatočná budeme používať práve knižnicu *ldns*.

Knižnica *ldns* je pomerne veľká knižnica a obsahuje funkcie na tie najzákladnejšie operácie potrebné pre DNS. Príklady si uvádzať nebudeme, lebo ide o rozsiahlu knižnicu a na vytvorenie požiadavku je potrebných viacero funkcií. Príklady použitia ako aj ďalšie informácie môžu byť nájdené v [9].

3.2 Informácie z databázy Whois

Whois je databáza, ktorá ukladá informácie o majiteľoch domén a IP adries. Na prácu s Whois databázou slúži protokol s rovnakým názvom Whois. Špecifikácia tohto protokolu je v RFC 3912 [5]. Tento protokol je typu dotaz / odpoveď a je pomerne jednoduchý. Jeho hlavnou úlohou je doručiť údaje informácie v podobe čitateľnej pre človeka. Z pohľadu užívateľa je to pohodlné, ale z pohľadu programátora ide o problematickú záležitosť, keďže nie je presne zadefinovaná syntax, ani sémantika informácií v tomto protokole.

Naša aplikácie bude používať databázu Whois na získavanie administratívnych geografických adries vlastníkov domén. Pri získavaní geografických adries môžme naraziť na rôzne formáty uloženia v databázy Whois. Nástroj, ktorý je cieľom tejto práce, má sledovať domény TLD a SLD domény `cz`. Štruktúra databáz národných domén TLD je hierarchická, to znamená že každý národný registrátor spravuje svoj vlastný server Whois. Keďže národný registrátory spravujú svoje vlastné servery Whois, tak informácie o SLD doménach `cz` budeme získavať z rovnakého servera Whois. Tým pádom môžme očakávať geografické adresy v rovnakom formáte. Aj keď ostatné SLD domény nie sú predmetom tejto práce, tak na základe krátkeho prieskumu sa dá predpokladať, že formát adries z Whois, ktoré patria pod RIPE,je zhodný ako v `cz` doméne. To však už neplatí o doménach napríklad v Afrike.

Z databázy Whois je možné získať aj priamo informácie o stave DNSSEC pre danú doménu. Avšak tieto informácie sú v databázy Whois nepovinné a platnosť týchto údajov by sa ťažko overovala. Pre domény `cz`, ktoré sú zabezpečené pomocou DNSSEC, Whois obsahuje aj záznam `DS`. Nás bude zaujímať len geografická adresa z databázy Whois.

Formát dát z databázy Whois vyzerá následovne (uvádzame len informácie, ktoré nás zaujímajú z pohľadu tejto práce):

```
whois vutbr.cz
[Querying whois.nic.cz]
contact:      SB:VUTBR-CZ
org:          Vysoke uceni technicke v Brne
name:         Vysoke uceni technicke v Brne
address:      Antoninska 548/1
address:      Brno
address:      601 90
address:      CZ
e-mail:       vit.slama@vutbr.cz
registrar:    REG-GENREG
created:      10.08.2001 22:13:00
changed:      10.02.2011 13:30:09
```

3.2.1 Nástroje na prácu s Whois

Protokol Whois patrí medzi jednoduché protokoly a slúži hlavne ako zdroj informácií pre administrátorov, ale aj pre verejnosť.

Nástroj whois

Najrozšírenejší a pravdepodobne najpoužívanejší, je nástroj, ktorý je súčasťou UNIXových systémov, *whois*. Nástroj `whois` sa spúšťa z príkazového riadka. Tento nástroj v našej apli-

kácii nebudeme používať a uvádzame ho len z toho dôvodu, lebo patrí medzi najrozšírenejšie pre prácu s Whois.

Trieda phpWhois

Trieda *phpWhois* je vyvinutá spoločnosťou *easyDNS Technologies Inc.* Je implementovaná v skryptovaciom jazyku PHP. Na to aby fungovala je potreba PHP preprocesor minimálne verzie 4.3.0.

Trieda *phpWhois* okrem toho, že sa vie dotazovať systému Whois, vie získané informácie aj parsovať. Protokol Whois, ako bolo spomenuté, vracia informácie v podobe čitateľnej pre človeka. Trieda *phpWhois* vie tieto informácie rozparsovať a uložiť do štruktúry, s ktorou môžme ďalej pracovať. Ako sme spomenuli v 3.2, tak každý server Whois môže poskytovať informácie v inom formáte. Trieda *phpWhois* rieši aj tento problém. Informácie o doménach SLD rovnakého typu (napríklad všetky domény `.cz`), sú obsiahnuté na rovnakom servery Whois. Trieda *phpWhois* obsahuje parsre pre väčšinu domén TLD. Vďaka tomu pri použití triedy *phpWhois* sa naša aplikácia nemusí zameriavať len na domény `cz`, ale môže podporovať aj ostatne domény TLD, tie ktoré je trieda *phpWhois* schopná prečítať.

Viacej o triede *phpWhois*, ako pre ktoré všetky domény TLD je implementovaná podpora, je možné a dočítať v [7]. Z tohto zdroju je možné triedu aj stiahnuť pod licenciou GPL.

3.3 Informácie o geolokáci

Výsledkom tejto práce má byť geografická mapa, ktorá zobrazuje súčasný stav DNSSEC v praxi. Doteraz sme si uviedli ako získať informácie o tom či je daná doména zabezpečená, alebo nie a z kade a ako získať geografické adresy. Poslednou potrebnou informáciou na vykreslenie dát na mapu sú geografické súradnice.

Geografické súradnice je možné získavať dvoma spôsobmi a to:

- Na základe IP adresy
- Ná základe geografickej adresy

Geolokáciu na základe IP adresy je možné použiť na IP adresu autoritatívneho serveru DNS pre danú adresu. Pri doménach TLD ešte bolo možné použiť geolokáciu na základe IP adresy, lebo ich nieje veľký počet (v čase písanie tejto práce ich počet je 292) a každá TLD doména má svoj vlastný autoritatívny server DNS. Geolokácia na základe IP adresy je nepresnejšia ako na základe geografickej adresy, lebo neexistuje databáza, ktorá by obsahovala presné informácie o lokácií danej IP adresy, keďže IP adresy sú prideľované internetovým providerom po celých blokoch.

Pri doménach SLD už nie je možné použiť geolokáciu na základe IP adresy autoritatívneho server DNS. Domény SLD sú registrované pomocou registrátorov. Registrátory sú spoločnosti na ktoré je delegované právo registrovať domény od registrátora národnej domény. V Českej Republike je to združenie CZ.NIC. Registrátory vlastnia svoje vlastné servery DNS, ktoré sa stávajú autoritatívnymi servermi pre domény, ktoré sú cez nich zaregistrované. Domén SLD je výrazne väčší počet ako TLD (v čase písania tejto práce je ich približne 750 000) a veľký počet domén zdieľa rovnaký server DNS. Geolokácia na základe IP adresy tak nie je možná, lebo na výslednej mape by bolo veľa domén na jednom mieste a

tieto lokácie by neodpovedali skutočným lokáciam ich vlastníkov. Preto pri doménach SLD je potreba použiť geografickú lokalizáciu na základe geografickej adresy vlastníka domény.

V aplikácií, ktorá je predmetom tejto práce budeme používať geolokáciu na základe geografickej adresy. Tento spôsob je presnejší a z databázy Whois máme k dispozícií geografické adresy.

3.3.1 Nástroje a služby pre geolokáciu

V tejto časti práce si popíšeme pomocou akých nástrojov a akých služieb budeme získavať informácie o geolokácií domén, ktoré sú zabezpečené pomocou DNSSEC.

GeoLite City od Maxmind

Väčšina nástrojov pre geolokáciu je v podobe webových stránok, alebo ako API prístupné cez web. Nástroj *GeoLite City* od spoločnosti *Maxmind* je dostupný ako balík, ktorý obsahuje databázu a API pomocou ktorého je možné s databázou pracovať. *GeoLite City* je dostupný po licenciou GPL. Ide o nástroj, ktorý slúži na určenie geografickej polohy na základe IP adresy. *Maxmind* ponúka aj platenú verziu tohto nástroja, ktorá je presnejšia ako tá, ktorá je volne dostupná.

Použitie nástroja *GeoLite City* je pomerne jednoduché. V aplikácií, ktorá bude mapovať stav DNSSEC, bude použitá len okrajovo a to na geolokáciu domén ku ktorým sa nepodarilo získať geografickú adresu z databázy Whois.

Viac informácií o *GeoLite City* a jeho použití je možne nájsť v [8].

Google Maps

Google Maps už nie je nástroj, ktorý sa dá stiahnuť a lokálne používať, ale ide o službu, ktorá je prístupná cez web, alebo API napísane v jazyku JavaScript[2]. Služba *Google Maps* neslúži len na geolokáciu na základe geografickej adresy. *Google Maps* sú hlavne zamerané na vytváranie geografických máp a prezentáciu dát na mape. K tomu patrí aj geolokácia na základe geografickej adresy.

Geolokácia pomocou *Google Maps* je prístupná dvoma spôsobmi :

- API napísané v jazyku JavaScript
- Protokol HTTP

Výstup je rovnaký z obidvoch spôsobov použitia tejto služby. Formát výstupu je možné zadefinovať pomocou parametra a *Google Maps* podporuje tieto dva formáty:

- XML
- JSON

Na získavanie geografických súradníc pre danú geografickú adresu, budeme používať *Google Maps* cez protokol HTTP. Toto rozhranie je prístupné cez protokol HTTP na internetovej adrese `http://maps.googleapis.com/maps/api/geocode/xml`. Táto adresa je pre výstupný formát XML. Pokiaľ by sme požadovali výstup vo formáte JSON, tak v adrese namiesto `xml` použijeme `json`. Následne v URL parametroch predáme geografickú adresu. Na výstupe bude XML štruktúra, ktorá bude obsahovať informácie o danej lokalite. Výhodou *Google Maps* je, že sa nemusíme starať o formát adresy, ktorú predávame parametrom, ale *Google Maps* vo svojej réžií sa postará o správne rozpoznanie adresy.

Uvedieme si príklad geolokácie adresy *Božetechová 2, Brno, Česká Republika*. V príklade uvedieme aj rôzne formáty adresy, pre ktoré *Google Maps* vráti rovnaký výsledok. Predanie adresy v parametre vyzerá následovne :

```
maps.googleapis.com/maps/api/geocode/xml?address=Bozetechova 2,Brno
maps.googleapis.com/maps/api/geocode/xml?address=Brno, Bozetechova 2
maps.googleapis.com/maps/api/geocode/xml?address=Bozetechova 2 CZ
```

Pre všetky prechádzajúce volania API *Google Maps* a aj iné kombinácie adries bude výsledok rovnaký. Výsledok obsahuje XML štruktúru, ktorej sú detailné informácie o tejto polohe. Nás budú zaujímať len geografické súradnice.

3.4 Zhrnutie

V tejto kapitole sme si stručne popísali služby, ktoré budeme používať k vytvoreniu aplikácie, ktorá je predmetom tejto práce. Vrámci týchto služieb sme si vymedzili informácie, ktoré sú z nášho pohľadu potrebné pre vytvorenie webovej mapy.

Uviedli sme si nástroje, pomocou ktorých budeme s týmito službami pracovať a v krátkosti uviedli ich použitie.

Kapitola 4

Návrh aplikácie

V kapitole 3 sme si uviedlo zdroje informácií pre vytvorenie webovej mapy, ktorá bude mapovať stav DNSSEC v praxi. V rámci týchto zdrojov sme vymedzili informácie, ktoré sú z pohľadu našej aplikácie potrebné a uviedli sme si nástroje pomocou ktorých je možné tieto informácie získavať.

V tejto kapitole kapitole si popíšeme princípy ako bude celé získavanie informácií a ich zobrazovanie fungovať. Niektoré zo zdrojov, z ktorých budeme čerpať informácie obsahujú limity, s ktorými je potrebné sa vysporiadať. Ako sa s nimi budeme vysporadúvať bude popísané v tejto kapitole.

Všetky zdroje informácií, ktoré sme uviedli v kapitole 3 sú voľne dostupné pre širokú verejnosť. Ale informácia o tom aké domény sledovať nie je voľne dostupná. Združenie CZ.NIC je národným registrátorom a tak má k dispozícií zoznam všetkých domén SLD pre doménu `cz`. Táto práca vzniká v spolupráci so združením CZ.NIC a pre účel tejto práce poskytlo zoznam zaregistrovaných domén `cz`. Je potreba navrhnúť mechanizmy, ktoré zabránia tomu aby aplikácia tieto informácie poskytovala ďalej, alebo aby bolo možné nejakým spôsobom tieto informácie znej vyčítať. Čo sa týka domén TLD, ich zoznam je voľne dostupný na internete.

Celá aplikácia bude rozdelená do dvoch častí:

- **Serverová časť** - táto časť aplikácie bude zbierať informácie potrebné na vykreslenie do mapy.
- **Prezentačná časť** - táto časť bude zobrazovať informácie získané pomocou serverovej časti.

4.1 Serverová časť

Serverová časť bude mať na starosti zbieranie dát a ich aktualizáciu. Serverová aplikácia bude bežať na pozadí systému na ktorom bude aplikácia nainštalovaná a priebežne bude zbierať nové informácie a sledovať zmeny v stávajúcich dátach. Všetky tieto dáta budú uložené v databázy, s ktorou bude potom pracovať aj prezentačná časť, ktorá má na starosti samotné vykresľovanie dát do mapy.

4.1.1 Ukladanie dát

Všetky dáta, s ktorými bude aplikácia pracovať, budú uložené v databázy. Predtým ako začneme s návrhom databázy je potrebné si vybrať databázový systém, na ktorom aplikácia-

ciu vystaviame. Pre implementáciu tejto aplikácie som zvažoval dva databázové systémy, *MySQL* a *SQLite*.

- *MySQL* je relačná databáza ktorá beží na servery. To znamená, že pre využívanie tohto databázového systému je potreba aby na systéme, na ktorom je databáza nainštalovaná, na pozadí bežal démon ktorý obsluhuje databázu.
- *SQLite* narozdiel od *MySQL* nepotrebuje žiadnu podporu od systému, na ktorom databáza beží. Databáza pre konkrétnu aplikáciu je uložená v jednom súbore a funkcie na prácu s databázou sú vložené do programu v čase kompilácie. Zo strany operačného systému nie sú za behu potrebné žiadne knižnice ani externé programy.

Databáza pre aplikáciu bude vytváraná v dlhších časových intervaloch, kvôli obmedzeniam niektorých služieb, z ktorých budeme získavať informácie, ako *Google Maps*. Hlavne z tohto dôvodu som pre implementáciu zvolil databázu *SQLite*. Databáza bude vždy uložená v jednom súbore a pri presune na iný systém aplikácia v získavaní a aktualizovaní dát bude pokračovať, tam kde skončila na predchádzajúcom systéme. Ďalšia výhoda, že databáza nebude závislá na systéme kde momentálne pracuje.

Dáta, ktoré budeme ukladať do databázy, budu domény a získané informácie o nich. Teda bude to doména s informáciami o vlastníkovi, stave domény a geografickej lokácií a k nej budeme ukladať záznamy RR z DNS, ktoré sme určili v 3.1.1. Tieto údaje je potrebné ukladať aby sme mohli filtrovať údaje na mape podľa zvolených kritérií a tak isto pre vytváranie štatistík v budúcnosti. Záznamy RR môžme do databázy ukladať dvoma spôsobmi:

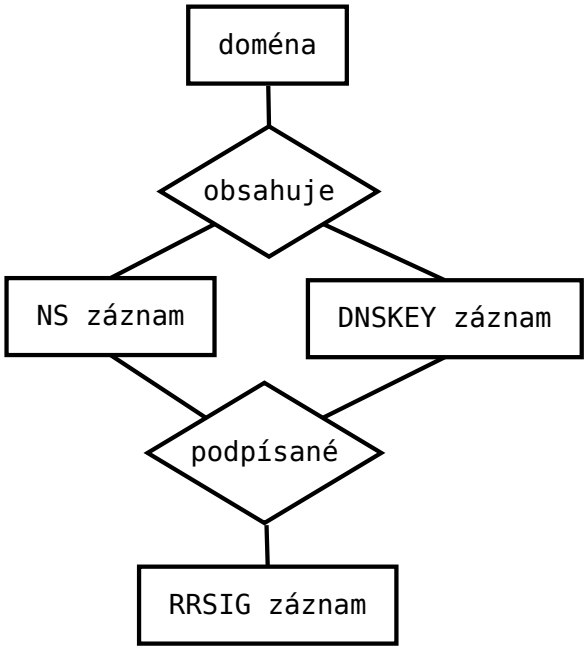
1. Prvý spôsob je navrhnúť databázovú štruktúru, do ktorej by bolo možné uložiť každý typ záznamu RR. Keďže záznam RR má presne definovanú štruktúru, tak databázová štruktúra by bola pomerne jednoduchá.
2. Druhý spôsob je definovať samostatnú databázovú tabuľku pre každý záznam RR ukladaný v databázy.

V kapitole 3.1.1 sme určili, ktoré všetky záznamy RR budeme ukladať v databázy. Pre implementáciu som zvolil druhý spôsob, vytvorenie zvlášť tabuľky pre každý ukladaný záznam RR, lebo poznáme dopredu typy záznamov, ktoré budeme ukladať. Druhý spôsob návrhu je aj rýchlejší oproti prvému z pohľadu vyhľadávania. Databáza bude pre doménu `cz` obsahovať približne 700 000 domén a je potrebné, aby vyhľadávanie bolo čo najrýchlejšie. Dáta pre databázu (zoznam SLD domén) bude načítaný zo zoznamu, ktorý pre účel tejto práce poskytlo združenie CZ.NIC.

Na obrázku 4.1 je ER diagram, ktorý zobrazuje vzťahy v databázy.

4.1.2 Informácie o stave zabezpečenia

Prvotná informácia, ktorú systém obsahuje, je názov domény. V prvom kroku získavania dát je potreba zistiť stav o zabezpečení domény. Túto skutočnosť je možné získať dotázaním sa databázy DNS na existenciu záznamu DS. Ak je doména zabezpečená pomocou technológie DNSSEC, záznam DS musí byť prítomný. Ak by nebol prítomný, nebolo by možné pri overovaní platnosti záznamov vytvoriť reťazec dôvery a tak overiť platnosť získaných záznamov pre danú doménu. Prítomnosť záznamu DS ešte neznamená, že doména je zabezpečená. Aj samotný záznam DS je potreba overiť u nadriadenej autority a zistiť, či je možné vytvoriť reťazec dôvery až po bod, ktorému dôverujeme.



Obrázek 4.1: ER diagram zobrazujúci vzťahy v databázy.

Na získavanie informácií o zabezpečení budeme používať knižnicu *libunbound* zo softwarového balíka *Unbound*. Túto knižnicu sme si popísali v časti 3.1.2. Pokiaľ určíme aj bod dôvery, tak je schopná vytvoriť reťazec dôvery a skontrolovať platnosť záznamov získaných zo systému DNS.

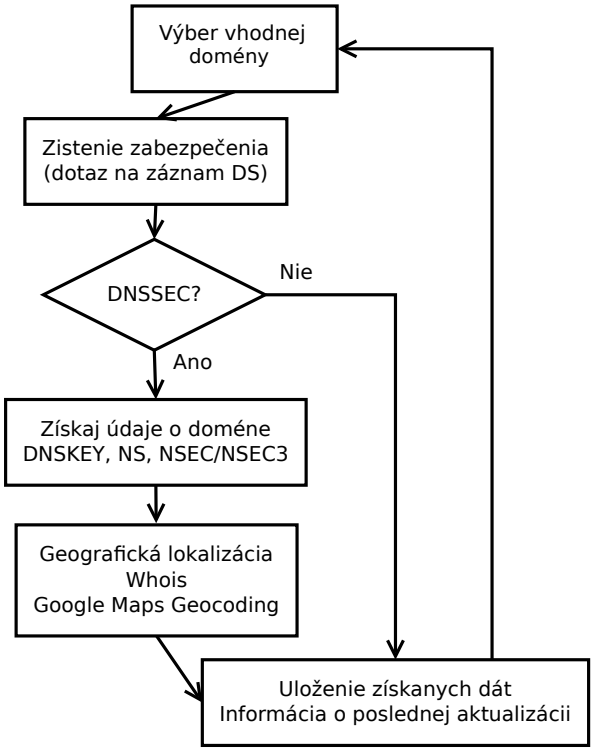
Program v nekonečnom cykle bude kontrolovať zabezpečenie domén. Každá doména v databázy obsahuje príznak, ktorý určuje či je zabezpečená, čas kedy bola zaznamenaná informácia o zabezpečení (kvôli štatistikám do budúcna) , revízne číslo, ktoré udáva koľko krát bola daná doména kontrolovaná na zabezpečenie a čas poslednej kontroly. Na základe týchto troch údajov sa vyberie vhodná doména na kontrolu stavu zabezpečenia. Prednosť dostávajú domény, ktoré sú nezabezpečené a domény s najnižším revíznym číslom.

Niektoré verejné servery DNS, ktoré podporujú technológiu DNSSEC, obsahujú obmedzenia ako napríklad počet dotaz na určitý časový interval. Databáza obsahuje pomerne veľa záznamov a tieto limity by sa rýchlo vyčerpali. Ideálne je aby lokálny server DNS podporoval technológiu DNSSEC. Pre vývoj aplikácie budeme používať server DNS, ktorý je nainštalovaný na localhoste a podporuje technológiu DNSSEC.

4.1.3 Určenie geografickej polohy domény

Informácie o zabezpečení domény máme k dispozícií z predchádzajúceho kroku. Pre vytvorenie mapy, ktorá zobrazuje stav technológie DNSSEC, je potrené spojiť názov domény s geografickou polohou. Na to nám poslúži databáza Whois, ktorá obsahuje geografickú adresu vlastníka domény.

Databáza Whois môže obsahovať neaktuálne informácie, alebo geografickú adresu ktorá neodpovedá skutočnému vlastníkovi domény (napríklad adresa je uvedená na registrátora domény, nie na vlastníka) a tak spôsobiť nepresnosti v zobrazení na mape. Treba si uvedomiť, že predmetom tejto práce nie je vytvoriť čo najpresnejšiu geolokáciu domén, ale



Obrázek 4.2: Vývojový diagram serverovej časti aplikácie.

mapovať stav zavedenia technológie DNSSEC do praxe. Preto sú pre nás dôležitejšie informácie zo systému DNS, ako z Whois a nepresnosti spôsobené nepresnými informáciami z Whois budeme tolerovať.

4.1.4 Geografická lokácia adresy

Posledným krokom, aby sme získali kompletné informácie na vytvorenie mapy, je geografická lokácia adries, ktoré sme získali z databázy Whois. Geografická lokácia (podľa štandardu WGS 84), tak isto ako geografická adresa jednoznačne identifikuje lokalitu. Súradnice sa skladajú z dvoch čísiel typu float, kde jedno reprezentuje zemepisnú dĺžku a druhe zemepisnú šírku. Geografické súradnice budeme získavať pomocou služby *Google Maps*, ktorá obsahuje nástroj na geografickú lokalizáciu.

Google Maps obsahuje limit na počet prekladu geografickej adresy na geografické súradnice. Tento limit je 2500 prekladov na jednu IP adresu za posledných 24 hodín. Je potrebné vhodne rozvrhnúť použitie tejto služby, aby sme nevyčerpali limit.

4.1.5 Vývojový diagram serverovej časti

Na obrazku 4.2 je znázornený vývojový diagram serverovej časti, ktorá má za úlohu zbierať a aktualizovať infomácie o doménach a ich stave zabezpečenia technológiou DNSSEC.

Obrázek 4.3: Zobrazenie väčšieho počtu dát - klastrovanie.

4.2 Zobrazovanie dát na mape

V 4.1 sme popísali ako budú dáta uložené a ako budú získavané. Druhá časť aplikácie sa bude starať o vizualizáciu dát. Zobrazovanie dát bude v podobe webovej mapy, ktorá bude zobrazovať dáta na geografickú mapu. Táto časť sa bude skladať z dvoch menších častí:

1. **Prezentačná časť** bude zobrazovať dáta na geografickej mape a reagovať na požiadavky užívateľa (filtrovanie dát).

2. **Dátova časť** sa bude starať o získavanie dát z databázy, ktorá bola vytvorená serverovou časťou, ktorú sme popísali v 4.1.

Tieto dve časti budú spolu komunikovať pomocou asynchronného posielania dotazov z JavaScritpu (skrátene AJAX).

Pri zobrazovaní dát na mapu je potreba vyriešiť aj formu ich zobrazovania. Keďže budeme zobrazovať veľký počet dát (bodov na geografickej mape) je potreba riešiť problém, keď ich bude na malom priestore veľa. Napríklad v prípade keď máme zobrazenú na mape celú Českú Republiku a sú vykreslené všetky údaje pre dané zobrazenie mapy. Tak vznikne veľký zhluk bodov na mape a mapa tak bude neprehľadná. Tento problém budeme riešiť zoskupovaním bodov, ktoré sú blízko seba. Výsledok takéhoto zobrazenia ilustruje obrázok 4.3.

4.2.1 Zabezpečenie dát

Pri zobrazovaní dát pre užívateľa je potrebné zabezpečiť aby cez túto aplikáciu nebolo možné získať citlivé dáta. Všetky informácie, s ktorými aplikácia pracuje, sú získane zo zdrojov, ktoré sú voľne dostupné. Jedine zoznam domén SLD, ktorý pre účely tejto práce

poskytlo združenie *CZ.NIC*, nie je voľne dostupný a preto treba zamedziť tomu, aby bolo možné tento zoznam domén získať.

Prvý spôsob zabezpečenia bude ošetrenie toho, aby dátová časť neposielala do prezentačnej časti zoznam domén. Vždy budú zaslané len body (geografické súradnice), na ktorých sa nachádzajú nejaké dáta. Až po vyžiadaní užívateľa o dáta nejakého bodu, budú zaslané dáta s doménami, ktoré sa nachádzajú na danom bode. Takto zaistíme aby sa neprenášal celý zoznam domén.

Druhý spôsob bude obmedzenie na počet dotazov na jednu IP adresu za časový interval. V tomto prípade ide o to aby nebolo možné iteratívnym dotazovaním prečítať obsah databázy.

V aplikácií budeme implementovať obidva spôsoby súčasne. Tie zamedzia tomu, aby bolo možné prečítať zoznam domén pomocou bežných skriptov.

Mapa bude implementovaná pomocou služby *Google Maps*. Tá obsahuje API, ktoré je implementované v jazyku JavaScript, to znamená že je interpretované na strane internetového prehliadača. Teoreticky existuje spôsob, kedy by útočník implementoval bota, ktorý by vedel používať užívateľské rozhranie internetového prehliadača a vedel rozoznávať body na mape, ktoré označujú zobrazované dáta. V dostatočne dlhom časovom intervale by bol postupným prechádzaním všetkých bodov na mape schopný získať zoznam zabezpečených domén. Po konzultácii s pánom Ondrejom Surým zo združenia CZ.NIC, predpokladáme, nie je tak veľká motivácia k prevedeniu takto časovo a implementačne náročného útoku.

4.3 Zhrnutie

Predmetom tejto kapitoly bolo navrhnúť aplikáciu, ktorá bude zbierať a aktualizovať dáta o súčasnom stave nasadenia technológie DNSSEC. Uviedli sme si ako budeme používať zdroje informácií uvedených v kapitole 3. Navrhli sme bezpečnostné mechanizmy, ktoré zabraňujú tomu, aby bolo možné cez aplikáciu získať citlivé informácie (zoznám domén poskytnutý združením *CZ.NIC*).

Kapitola 5

Implementácia

V tejto kapitole sa budeme zaoberať samotnou implementáciou aplikácie. Kapitola bude rozdelená na dve hlavné časti. Prvá sa bude zaoberať implementáciou serverovej časti, teda uložením a získavaním dát. Druhá časť sa bude zaoberať implementáciou prezentačnej časti, ktorá má za úlohu zobrazovať získané dáta a zabezpečiť ich tak, aby nebolo možné ich neoprávnene získať.

5.1 Databáza na ukladanie informácii o DNSSEC

Na obrázku 4.1 je znázornený ER diagram. Na základe týchto vzťahov navrhneme tabuľky v databázy, ktoré budú ukladať všetky potrebné dáta. V tejto časti práce si vytvoríme iba tú najhlavnejšiu tabuľku `domains`. Ostatné je možné nájsť v prílohe C.

5.2 Implementácia serverovej časti aplikácie

Pracovanie serverovej časti môžme rozdeliť do viacerých samostatných častí:

1. Výber domény z databázy, pre ktorú sa budú aktualizovať údaje.
2. Zistenie stavu zabezpečenia pre danú doménu pomocou technológie DNSSEC.
3. Získanie geografických údajov o lokácií danej domény.

domains		
`domain`	TEXT	názov domény
`secured`	INTEGER	príznak zabezpečenia domény
`lastUpdate`	INTEGER	čas poslednej aktualizácie
`lat`	REAL	zemepisná šírka
`lng`	REAL	zemepisná dĺžka
`revision`	INTEGER	počet aktualizácií danej domény
`level`	INTEGER	informácia o type domény
`distributed`	INTEGER	príznak o distribúcií na geografickú lokáciu

Tabulka 5.1: Štruktúra databázovej tabuľky `domains`.

5.2.1 Výber vhodnej domény

Server funguje v nekonečnom cykle. Na začiatku každého cyklusu je potreba vybrať doménu, pre ktorú budeme získavať informácie. Je potreba zabezpečiť, aby počas programu boli v pravidelných intervaloch aktualizované informácie o všetkých doménach, ktoré sú obsiahnuté v databázy.

V tabuľke C sme zadefinovali, ako bude vyzerať hlavná tabuľka databázy, ktorá ukladá základné informácie o doméne. Pri výbere vhodnej domény nás budu zaujímať hlavne tri položky z tejto tabuľky:

1. `secured` - informácia o tom či je doména zabezpečená
2. `lastUpdate` - čas poslednej aktualizácie
3. `revision` - počet aktualizácií pre danú doménu.

Najprv je potrebné zadefinovať interval v akom sa budú dáta pre domény aktualizovať. Po konzultácií s pánom Ondrejom Surým zo združenia *CZ.NIC* sme sa zhodli, že raz za dva týždne bude postačujúce. Interval pre aktualizáciu teda bude každých 1209600 sekúnd. Interval uvádzame v sekundách z toho dôvodu, lebo v databázy máme čas poslednej aktualizácie uložený v podobe Unixového času, to je počet sekúnd od 1. januára 1970 (Unix epoch).

Výber domény na spracovanie bude záležať na čase od poslednej aktualizácie, teda či spadá do intervalu dvoch týždňov. Ďalej záleží od počtu, koľko krát bola daná doména aktualizovaná. Domény s menšou hodnotou `revision` majú prednosť. Tým sa zabezpečí aby sa v pravidelných intervaloch kontrolovali všetky domény a nenastalo vyhladovanie, teda aby určité domény (napríklad tie ktoré sú stále nezabezpečené) nedostávali prednosť pred ostatnými.

Posledná podmienka je informácia o stave zabezpečenia. Domény, ktoré zatiaľ nie sú evidované ako zabezpečené dostávajú prednosť na aktualizáciu pred doménami, ktoré už sú zabezpečené pomocou technológie DNSSEC.

Dotaz na výber domény vhodnej na spracovanie vyzerá následovne:

```
SELECT * FROM domains WHERE lastUpdate < (strftime('%s','now') - 3600 )
ORDER BY revision ASC, secured ASC
```

Dôležité je si všimnúť podmienku zoraďovania, kde záleží na poradí v akom sú zoraďované. Podstatné je, aby zoraďovanie podľa počtu aktualizácií malo prednosť pred tým, či je doména zabezpečená, alebo nie je.

5.2.2 Zisťovanie stavu zabezpečenia

Skôr ako začneme overovať stav zabezpečenia domény, ktorú sme vybrali v predchádzajúcej časti, implementujeme dve triedy, ktoré budú zjednodušovať prácu so systémom DNS. Knižnice, ktoré budeme používať pre prácu s DNS, *lunbound* a *ldns*, sú napísane v jazyku C, teda obsahujú funkcie na prácu s DNS. My implementujeme triedy, ktoré tieto obsahujú funkcie a dáta (štruktúry) na prácu s DNS.

Trieda DNSSEC_Resolver

Trieda *DNSSEC_Resolver* implementuje základnú prácu s DNS, teda dotazovanie sa systému DNS a overovanie platnosti výsledkov dotazovania. Metódy triedy *DNSSEC_Resolver* sú implementované pomocou knižnice *lunbound*.

Prvá metóda implementovaná v *DNSSEC_Resolver* je na zasielanie dotazov do systému DNS. Prijíma dva parametre. Prvý je doména, na ktorú sa dotazujeme a druhý je požadovaný typ záznamu RR.

```
DNS_packet queryRR(string domain,RRTYPE type);
```

Ako návratová hodnota z metódy je objekt typu *DNS_packet*, ktorú si popíšeme v nasledujúcej časti práce.

Druhou podstatnou metódou je overovanie platnosti záznamov. Trieda ma zapamätaný kontext z prechádzajúceho dotazu do systému DNS. Metóda *isValid* na základe podpisov, verejných kľúčov a zadaného bodu dôvery overí platnosť záznamov získaných v predošlom volaní metódy *queryRR*.

```
bool isValid();
```

Trieda DNSSEC_packet

Trieda *DNSSEC_packet* slúži na prácu so samotným DNS paketom. Sú v nej implementované hlavne metódy na získavanie informácií z DNS paketu, ktorý bol zaslaný ako odpoveď na dotaz. Na prácu s paketom sú vo vnútri triedy použité funkcie z knižnice *ldns*.

Trieda obsahuje celý paket, ktorý bol získaný ako odpoveď na dotaz a následne pomocou metód je možné získavať jednotlivé informácie z odpovedi. Uvedieme si len zopár metód, ostatné sú používané analogicky.

- `char* getRRSIGalgorithm()` - vráti typ algoritmu, ktorým bol záznam RRSIG (podpis iného záznamu) vytvorený.
- `char* getDNSKEY(int index)` - vráti verejný kľúč, uložený v zázname DNSKEY, ktorým je možné overiť podpisy záznamov. V odpovedi sa môže nachádzať viacej záznamov DNSKEY a parameterom `index` je možné určiť, ktorý záznam požadujeme.
- `char* getNS(int index)` - vráti hodnotu uloženú v zázname NS, teda adresu autoritatívneho servera DNS pre danú doménu. Podobne ako v prechádzajúcom prípade, odpoveď môže obsahovať viacej záznamov NS a cez parameter `index` je možné určiť ktorý požadujeme.

Overovanie zabezpečenia

Pri overovaní, či je daná doména zabezpečená, sa budeme dotazovať na záznam typu DS. Ten musí byť obsiahnutý pre každú doménu, ktorá je zabezpečená technológiou DNSSEC. Pokiaľ by záznam DS nebol prítomný, tak nie je možné overiť podpisy záznamov, lebo by nebolo možné vytvoriť reťazec dôvery. Prítomnosť samotného záznamu DS ešte neznamená, že doména je zabezpečená pomocou technológie DNSSEC. Je potrebné overiť platnosť tohto záznamu.

Overenie zabezpečenia domény v programe vyzerá následovne:

```
resolver.queryRR(domain,DS_RR);
if(resolver.isValid()) {
      ...
}
```

Ak je doména zabezpečená, získame pre ňu ďalšie údaje, ktoré budeme pre domény sledovať. Dodatočné informácie, ktoré sledujeme pre každú doménu sme si zadefinovali v 3.1.1. Ako príklad si uvedieme získanie všetkých verejných kľúčov uložených v zázname DNSKEY, ale len typu KSK.

```
packet = resolver.queryRR(domain, DNSKEY_RR);
packet.loadByType(LDNS_RR_TYPE_DNSKEY);
for(int i = 0; i < packet.getCountAnswer(); i++) {
    if(packet.getDNSKEYflag(i) == '257') {     // 257 je označenie KSK kľúču
          packet.getDNSKEY(i);
          packet.getTtl(i);
    }
}
```

Takýmto spôsobom získane informácie uložíme do databázy. Podobným spôsobom budeme postupovať pri získavaní ostatných záznamov RR, ktoré sme si vymedzili v kapitole 3.1.1.

Pokiaľ doména nie je zabezpečená, tak sa nezískavajú dodatočné informácie, ale sa aktualizuje počet aktualizácií pre danú doménu a čas poslednej aktualizácie.

5.2.3 Geografická lokácia domény

Posledným krokom v činnosti serverovej časti je geografická lokácia domény. Tá sa dá rozdeliť do dvoch častí.

1. Získanie geografickej adresy z databázy Whois.
2. Geografická lokácia na základe adresy získanej z databázy Whois.

Získavanie geografickej adresy

Na získavanie geografickej adresy vlastníka domény použijeme databázu Whois. Ako sme už spomínali, problémom je, že databáza Whois môže poskytovať dáta v rôznych formátoch a nieje špecifikovaný jednotný formát. V implementácií aplikácie budeme používať už vytvorenú triedu *phpWhois* [1], ktorá implementuje parsre pre rôzne servery Whois a informácie získane z Whois ukladá do štruktúry. Vďaka *phpWhois* aplikácia nebude zameraná len domény `cz`, ale bude podporovať aj väčšinu národných domén.

Z programu napísaného v jazyku C++ budeme triedu *phpWhois* (ktorá je napísaná v skriptovacom jazyku Php) pomocou funkcie `popen`[2]. Z pohľadu aplikácie nás bude zaujímať len jeden údaj a to geografická adresa vlastníka domény. Preto môžme vytvoriť skript, napísaný v jazyku Php, ktorý prijíma ako parameter názov domény a jeho výstupom je geografická adresa pre túto doménu. Tento skript na získanie adresy používa triedu *phpWhois*.

Celá práca s databázou Whois je implementovaná v triede `Whois`. Príklad použitia tejto triedy je následovný:

[1] Triedu je možné získať na adrese `http://www.phpwhois.org/`

[2] Informácie o `popen` je možné nájsť na http://pubs.opengroup.org/onlinepubs/007908799/xsh/popen.html

```
address = whois.getAddress(domain, QUERY_SLD);
```

Metóda `getAddress` prijíma dva parametre. Prvým parametrom je názov domény, ktorej chceme získať adresu. Druhým je typ domény na aký sa dotazujeme. Rozoznávame domény TLD a SLD.

Získanie geografickej lokácie

Posledná potrebná informácia o doméne je jej geografická lokácia. Tú budeme získavať na základe geografickej adresy získanej z databázy Whois.

Na geografickú lokalizáciu budeme používať službu *Google Maps*. Tá ma cez protokol HTTP dostupný nástroj na geografickú lokalizáciu na základe adresy. Služba je dostupná na adrese `maps.googleapis.com/maps/api/geocode/`. Cez parameter prijíma adresu. Adresa môže byť v ľubovolnom formáte a o správne rozpoznanie adresy sa stara *Google Maps* samostatne. Po detailnom testovaní rôznych kombinácií adresy, môžem konštatovať, že služba funguje spoľahlivo. Výsledok môže vyť vrátený v dvoch rôznych formátoch, XML a JSON. My budeme používať formát XML.

Práca s API pre *Google Maps* je implementovaná pomocou triedy *GoogleAPI*. Volanie nástroju na geografickú lokalizáciu je implementované v metóde `getLocation`. Tá prijíma ako parameter geografickú adresu, ktorú chceme previesť na geografické súradnice. Metóda odošle na adresu API požiadavok a ako návratová hodnota metódy je XML, ktoré obsahuje odpoveď. Samotné odoslanie HTTP požiadavku na adresu API je realizované pomocou knižnice *libcurl*[3]. Príklad odpovede od služby *Google Maps* vyzerá následovne:

```
<GeocodeResponse>
 <status>OK</status>
 <result>
  <formatted_address>
       Božetěchova 1/2, 612 00 Královo Pole, Czech Republic
  </formatted_address>
  <address_component>
   <long_name>Božetěchova</long_name>
  </address_component>
  <address_component>
   <long_name>Královo Pole</long_name>
   <type>sublocality</type>
  </address_component>
  <address_component>
   <long_name>Brno</long_name>
   <type>locality</type>
  </address_component>
  <address_component>
   <long_name>South Moravia</long_name>
  </address_component>
  <address_component>
   <long_name>Czech Republic</long_name>
  </address_component>
```

[3]Informácie o knižnici nájdete na `http://curl.haxx.se/libcurl/`

```
  <address_component>
   <long_name>Brno 12</long_name>
  </address_component>
  <geometry>
   <location>
    <lat>49.2265440</lat>
    <lng>16.5971216</lng>
   </location>
  </geometry>
 </result>
</GeocodeResponse>
```

Z XML štruktúry, získanej ako odpoveď od *Google Maps*, získame dáta geografickej polohy, na ktorej sa doména nachádza. Na prácu s XML štruktúrou použijeme XML parser *TinyXml* [4].

Získanie geografických súradníc, potrebných pre aplikáciu, implementuje metóda *getLocationCoordinates*. Tá prijíma ako parameter geografickú adresu a ukazovatele na dve premenné, do ktorých sa uloží zemepisná dĺžka a zemepisná šírka. Príklad použitia triedy *GoogleAPI* je následovný:

```
google.getLocationCoordinates(address,&lat,&lan);
```

Služba *Google Maps* obsahuje obmedzenie na počet preložení geografickej adresy na geografické súradnice. Tento limit je 2500 prekladov na jednu IP adresu za posledných 24 hodín. Aby sme zabezpečili, že tento limit sa nevyčerpá a aplikácia bude plynulo fungovať, zavedieme časový interval, po ktorý bude aplikácia čakať po každom volaní služby *Google Maps*. Interval po aký ma aplikácia čakať stanovíme následovne:

```
24hodín * 60minút * 60sekúd = 86400 sekúnd za deň
86400sekúnd / 2500dotazov = 34.56
```

Interval je teda stanovený na 35 sekúnd. To znamená, že každých 35 sekúnd sa zavolá API služby *Google Maps*. Takýmto spôsobom by získavanie geografických údajov pre databázu, ktorá obsahuje napríklad 100 000 domén zabezpečených technológiou DNSSEC, trvalo približne 40 dní. Treba poznamenať, že geografické údaje sa získavajú len pri doménach, ktoré systém práve zaregistroval ako zabezpečené. To znamená, ak doména v systéme je evidovaná ako zabezpečená a má priradené geografické súradnice, tak pri jej aktualizácií sa nebude znova vykonávať geografická lokácia. Aktualizácia geografickej lokácie, pri zabezpečených doménach, sa bude vykonávať jeden krát za 10 aktualizácií.

Distribuovaný výpočet geografickej lokácie

Problém nastáva ak sa do systému pridá na jeden krát veľké množstvo nových domén (napríklad celá nová národná doména). V takomto prípade by trvalo dlhý čas, pokiaľ by sa vytvorila aktuálna databáza. Limit služby *Google Maps* je zavedený na IP adresu. Preto je možné geografickú lokalizáciu distribuovať na viacej počítačov, ktoré majú rôzne IP adresy.

Distribúcia výpočtu bude implementovaná pomocou jednoduchej webovej stránky. Ako sme už uviedli, službu *Google Maps* je možné používať cez API implementované v jazyku JavaScript.

[4] Dokumentácia aj nástroj je dostupný na `http://www.grinninglizard.com/tinyxml/`

Samotná distribúcia výpočtu bude v podobe distribuovania odkazu na stránku. Kód stránky obsahuje skript napísaný v jazyku JavaScript, ktorý zo serveru načíta domény, ktoré obsahujú geografickú adresu, ale neobsahujú geografické súradnice. Pomocou API služby *Google Maps* získajú pre dané domény geografické súradnice a tie následne odošlú naspäť na server, kde sa aktualizujú v databázy.

Takýmto spôsobom je každému klientovi priradených 2000 domén na geografickú lokalizáciu a vytvorenie úplnej databázy je podstatne rýchlejšie a záleží na počte užívateľov zapojených do výpočtu.

Príklad geografickej lokácie v jazyku JavaScript:

```
geocoder = new google.maps.Geocoder();
geocoder.geocode( { 'address': address}, function(results, status) {
   if (status == google.maps.GeocoderStatus.OK) {
      sendToServer(results[0].geometry.location);
   }
}
```

Aby tento spôsob výpočtu mohol fungovať, je potrebné aby server vedel zbierať informácie bez geografickej lokalizácie. Táto možnosť bude implementovaná pomocou parameteru pri spúšťaný démona, ktorý zbiera informácie o stave domén.

5.3 Implementácia webovej mapy

Druhou časťou aplikácie je webová aplikácia, ktorá zobrazuje získané dáta pomocou serverovej časti, na geografickú mapu sveta. Ako sme spomenuli časti 4.2, ktorá sa zaoberala návrhom aplikácie, bude prezentačná časť zložená z dvoch častí. Dátová a prezentačná časť. Tieto dve časti spolu komunikujú pomocou technológie AJAX. Hlavnou výhodou tohto prístupu je, že dáta sú na stranu užívateľa načítane na pozadí. To znamená, že pri načítaní nových dát nieje potreba načítať znova celú webovú stránku.

5.3.1 Dátová časť

Dátová časť bude poskytovať dáta pre prezentačnú časť, na základe toho čo si užívateľ vyžiadal (napríklad filtrovaním). Táto časť aplikácie pracuje s rovnakou databázou ako serverová časť. Implementovaná bude v skriptovacom jazyku Php.

Získavania dát

Skriptu, ktorý obsluhuje databázu, sú na vstupe predané parametre od klienta pomocou metódy POST. Skript na základe týchto požiadavkou vytvorí dotaz do databázy. Užívateľ bude mať možnosť pomocou filtra si vybrať dáta. Skript dátovej časti bude vybavovať tri typy dotazov:

1. Zoznam bodov, ktoré obsahujú dáta. Tento dotaz je prevedený vždy pri načítaní webovej stránky, ktorá zobrazuje mapu. Príklad dotazu na výber domén typu SLD zabezpečených pomocou algoritmu RSA/SHA-1 (kód pre šifrovací algoritmu RSA/SHA-1 v záznamoch RR je 5) bude vyzerať následovne:

   ```
   SELECT lat,lng FROM domains
   INNER JOIN rrDNSKEY USING(domain)
   ```

```
INNER JOIN rrSIG USING(idRRdnskey)
WHERE secured AND lat!=0 AND lng!=0 AND
      level=2 AND rrdnskey.algorithm = 5
```

Tento dotaz vráti všetky súradnice domén, ktoré vyhovujú daným kritériam. Je dôležité si všimnúť, že z databázy nevyťahujeme názov domény. Do klientskej časti sú zaslané len súradnice bodov, na ktorých sa nachádza nejaká doména, alebo domény (na jednom mieste môže byť viacej domén).

2. Zoznám domén, ktoré majú rovnaké geografické súradnice, teda sa nachádzajú na rovnakom bode na mape. Tieto dáta sú klientovi poskytnuté v momente, keď si vyberie jeden konkrétny bod zobrazený na mape. V tomto prípade, narozdiel od prechádzajúceho prípadu sa posielajú už aj názvy domén. Príklad takéhoto výberu dát z databáze je následovný:

   ```
   SELECT domain FROM domains
   WHERE lat=17.497713 AND -88.186654  AND secured
   ```

3. Informácie o jednej konkrétnej doméne. Tieto informácie sú klientovi poskytnuté na dotaz na konkrétnu doménu. V tomto prípade sa z databázy vyberajú všetky informácie, ktoré sú k danej doméne dostupné.

Dáta uložené v databázy sú pomerne dlhodobé, to znamená že nie sú aktualizované často. Po získaní dát z databázy, je výsledok uložený v pamäti cache. Takže pri následujúcom dotaze od klienta nie sú dáta načítane z databázy, ale z pamäte cache, pokiaľ ich obsahuje.

Všetky dáta získané z databázy sú klientovi zaslané vo formáte JSON.

Ochrana databázy

Ochrana databázy bude spočívať v obmedzení počtu dotazov za časovú jednotku na IP adresu. Ide o podobný limit ako používa služba *Google Maps*. Po konzultácií s pánom Ondrejom Surým zo združenia CZ.NIC sme určili tento limit na 2000 dotazov na jeden deň. Limit je nastaviteľný v konfiguračnom skripte pre serverovú časť. Po prekročení limitu, je administrátorovi zaslaný email, ktorý sa môže rozhodnúť a danú IP adresu zablokovať pernamentne.

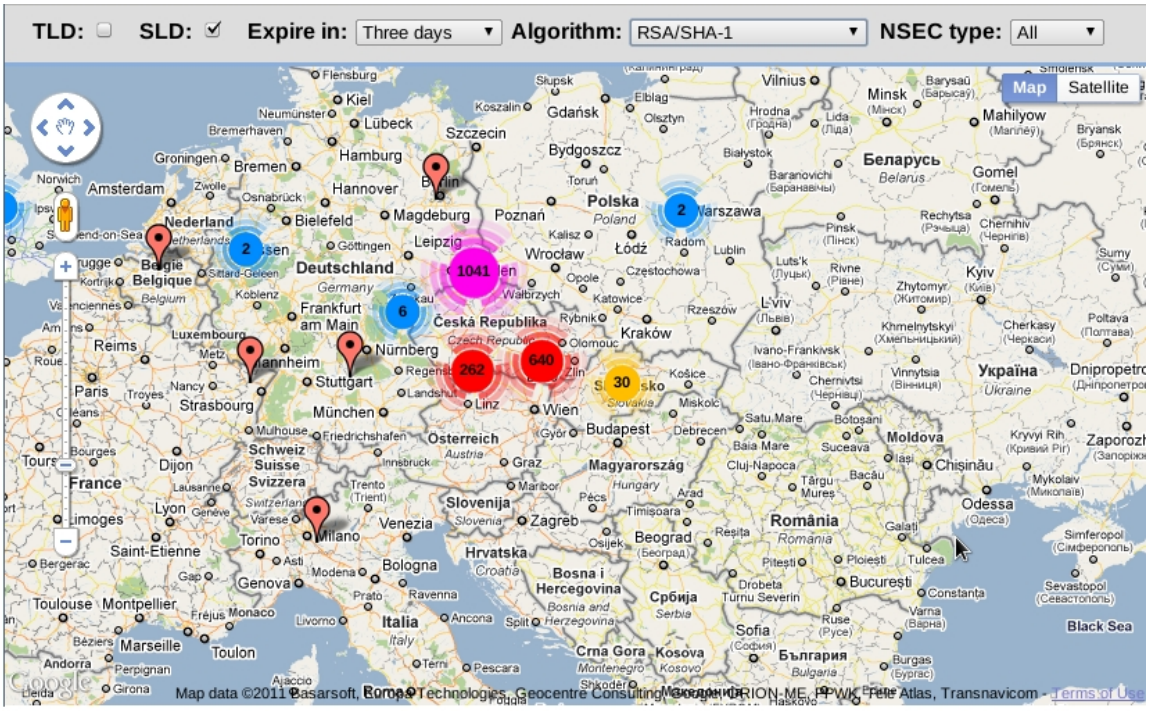
V skripte dátovej časti budeme počítať jednotlivým IP adresám počet prístupov za posledných dvadsaťštyri hodín. IP adresa z ktorej je užívateľ pripojený je v jazyku Php dostupná v premennej `$_SERVER['REMOTE_ADDR']`. Po pre

5.3.2 Prezentačná časť

Dáta sú prezentované užívateľovi pomocou geografickej mapy. Pre vytvorenie mapy a na zobrazenie dát na nej, je použitá služba *Google Maps*. Tá obsahuje API napísané v jazyku JavaScript. Narozdiel od nástroja, ktorý sme použili pre geografickú lokalizáciu adries, neobsahuje žiadne obmedzenia použitia.

Celá webová aplikácia je implementovaná v HTML 4.0 a JavaScripte. Pri implementácií bol použitý JavaScriptový framework jQuery [5], ktorý podstatne zjednodušuje prácu s webovými technológiami.

[5] jQuery, ako aj dokumentácia, je dostupný na adrese `http://jquery.com/`



Obrázek 5.1: Webová aplikácia na zobrazovanie informácií o stave technológie DNSSEC.

Pre vstup užívateľa slúži formulár v hornej časti aplikácie, pomocou ktorého je možné zadefinovať aké dáta sa majú zobraziť na mape. Ako filter pre dáta je možné zvoliť:

- Typ domény - TLD, alebo SLD.
- Doba platnosti verejného kľúča, ktorý slúži na overenie podpisov záznamov (ak jeho platnosť vyprší, nie je možné overiť podpisy).
- Typ algoritmu, ktorým boli podpisy vytvorené.
- Typ zázname pre poskytnutie overiteľnej odpovede o neexistencii domény.

Po každej zmene vo formulári sú nové dáta automaticky načítané. Načítavanie prebieha na pozadí a tak nieje potrebné znova načítať celú stránku. Počas načítavania dát zo servera, je užívateľ informovaný o tom, že prebieha prenos dát.

Na mape sa zobrazuje pomerne veľké množstvo dát a je potrebné ich zobrazovať tak, aby mapa bola čitateľná. Na spravovanie veľkého počtu bodov používame nástroj *MarkerClusterer* [6]. *MarkerClusterer* sa stará o to, že ak sa nachádza blízko seba veľa bodov, tak ich nahradí väčším bodom a zobrazí ich počet.

Na mape sú zobrazené body, ktoré obsahujú nejaké informácie o doménach. Po vybratí jedného konkrétneho bodu, sa zobrazí zoznam domén, ktoré sa nachádzajú na danom mieste. Pre každú doménu je možné zobraziť aj detailné informácie. Po vybratí jednej konkrétnej domény sa zobrazí okno s informáciami, ktoré obsahuje:

[6] `http://gmaps-utility-library.googlecode.com/svn/trunk/markerclusterer/1.0/docs/reference.html`

- Verejné kľúče pre danú doménu spolu s ich platnosťami, algoritmami a identifikátorom kľúča.
- Adresy autoritatívnych serverov DNS pre danú doménu.
- Typ záznamu pre neexistenciu subdomény - NSEC/NSEC3.
- Geografické informácie.

5.4 Zhrnutie

V tejto časti práce sme sa zaoberali hlavne implementáciou aplikácie, ktorá mapuje stav nasadenia technológie DNSSEC v praxi a súčasný stav zobrazuje na geografickej mape sveta.

V prvej časti sme sa zaoberali implementáciou serverovej časti. Tá sa stará o zbieranie nových informácií a aktualizovanie dát, ktoré sú už obsiahnuté v databázy. Povedali sme si akým spôsobom budú implementované jednotlivé časti, ktoré sa starajú o zbieranie a aktualizovanie dát. Do serverovej časti sme implementovali zabezpečenia, aby nebolo možné získať zoznam domén, ktoré pre účel tejto práce poskytlo združenie *CZ.NIC*.

V druhej časti kapitoly sme popisovali implementáciu samotnej webovej aplikácie, ktorá zobrazuje dáta na geografickú mapu svet. Na vytvorenie samotnej mapy sme použili službu *Google Maps* a jej API implementované v jazyku JavaScript.

Návod na použitie a konfiguráciu sa nachádza v prílohe B.

Kapitola 6

Testovanie systému

V tejto kapitole sa budeme venovať testovaniu systému vytvoreného v predchádzajúcich kapitolách. Prvej časti sa budeme venovať testovaniu serverovej časti a to získavaniu dát o stave zabezpečenia domén pomocou technológie DNSSEC.

Druhá časť sa bude venovať testovaniu celkovej aplikácie a to z pohľadu užívateľa. Budeme zisťovať, či je možné nejakým spôsobom získať všetky domény, ktoré sú obsiahnuté v databázy. Zoznam domén je jediná informácia, ktorá nie je voľne dostupná.

V poslednej časti tejto kapitoly, si zhrnieme výsledky použitia aplikácie a niektoré zaujímavé fakty ohľadom zabezpečenia domén pomocou technológie DNSSEC.

6.1 Testovanie získavania dát

Hlavnou úlohou je sledovanie stavu technológie DNSSEC v systéme DNS. Pre overenie fungovania spustíme démona s parametrom `-v`, ktorý spustí kontrolné výpisy.

```
./dnssec-daemon -v -g
[mojecaffe.cz.] not secured
[obecskvoretice.cz.] secured : NSEC : 49.3042663 : 14.4732797
[alfavariagroup.cz.] not secured
[hosteskybrno.cz.] secured : NSEC : 49.1946057 : 16.6065806
[combair.cz.] not secured
[koalicepronaturu.cz.] not secured
```

Z výstupu je vidieť, že domény `obecskvoretice.cz` a `hosteskybrno.cz` sú zabezpečené pomocou technológie DNSSEC. Ako odpoveď na neexistujúcu subdoménu používajú záznam NSEC. Keď sa pozrieme do databázy napríklad na doménu `hosteskybrno.cz`, tak vidíme, že démon uložil aj podrobnejšie informácie, ktoré sú dostupné zo systému DNS.

Výsledky je možné overiť pomocou dotazu do databázy:

```
SELECT domains.domain,nsEntry,lat,lng,dnskeyEntry FROM domains
INNER JOIN rrDNSKEY USING(domain)
INNER JOIN rrNS USING(domain)
WHERE domains.domain='hosteskybrno.cz.';
hosteskybrno.cz.|alfa.ns.active24.cz.|49.1946057|16.6065806|AQO1gtiQNJnn/9e..
hosteskybrno.cz.|beta.ns.active24.cz.|49.1946057|16.6065806|AQO1gtiQNJnn/9e..
```

Aplikácia je navrhnutá tak, aby zvládala aj iné SLD domény ako `cz`. Pridáme doménu `ul.pt`, o ktorej sme pomocou nástroju *dig* zistili, že je zabezpečená pomocou technológie DNSSEC. Doménu do systému pridáme pomocou parametra `-a`

```
./dnssec-daemon -a ul.pt.
Domain ul.pt. Added
```

```
./dnssec-daemon -v -g
[ul.pt.] secured : NSEC : 38.7069320 : -9.1356321
```

Nové informácie v databázy je možné overiť rovnakým spôsobom ako v predchádzajúcom prípade. Je potrebné poznamenať, že systém podporuje len domény SLD, pre ktoré je implementovaný parser v triede *phpWhois*. Zo systému Whois sa získava základná informácia pre geografickú lokalizáciu. Pokiaľ táto informácia v systéme chýba, systém vie overiť len stav zabezpečenia, ale nevie ju umiestniť na geografickú mapu.

6.2 Testovanie systému

Na sledovanie komunikácie medzi webovou aplikáciou a serverovou časťou budeme používať rozšírenie do prehliadača *Firefox* [1] pre debugovanie, *Firebug*[2].

6.2.1 Bezpečnosť aplikácie

V pri načítaní webovej aplikácie je na server zaslaný dotaz o dáta. Dotaz po načítaní webovej aplikácie je zobrazený pomocou *Firebugu* na obrázku 6.1. Dáta prijaté ako odpoveď servera vyzerajú následovne:

```
[{'lat':-4.6826693,'lng':55.480396},{'lat':32.387172,'lng':-90.044045},
{'lat':33.836081,'lng':-81.1637245},{'lat':34.1174411,'lng':-118.3258023},
{'lat':39.103086,'lng':-84.506344},{'lat':39.6790286,'lng':-75.7660466},
{'lat':39.7906246,'lng':-75.5317568},{'lat':40.6331249,'lng':-89.3985283},
{'lat':44.7626091,'lng':-109.0284742},{'lat':45.3132319,'lng':9.4881702}]
```

V odpovedi sú len súradnice bodov, na ktorých sa nachádzajú nejaké dáta. Odpoveď je zaslaná vo formáte JSO

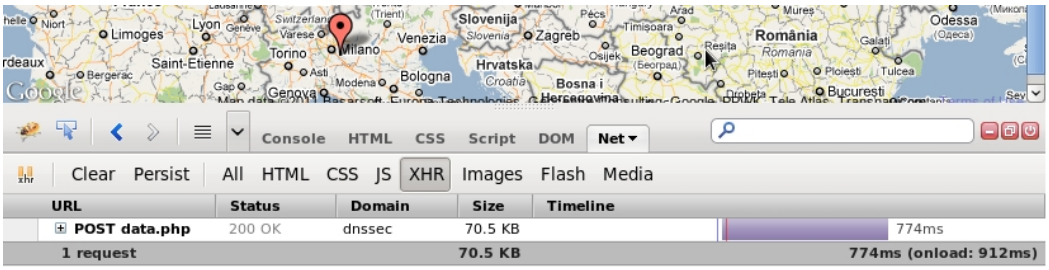
Na následnom obrázku D.1 je znázornená celá topológia aplikácie, ktorá sleduje stav technológie DNSSEC a zobrazuje dáta na geografickú mapu.

V okamihu, kedy si užívateľ zvolí jeden konkrétny bod na mape, tak je na sever zaslaný dotaz so súradnicami bodu a ako odpoveď už je názov domény. Dotaz zaslaný na server je zobrazený pomocou *Firebugu* na obrázku6.2. Odpoveď od servera na tento dotaz je následovná:

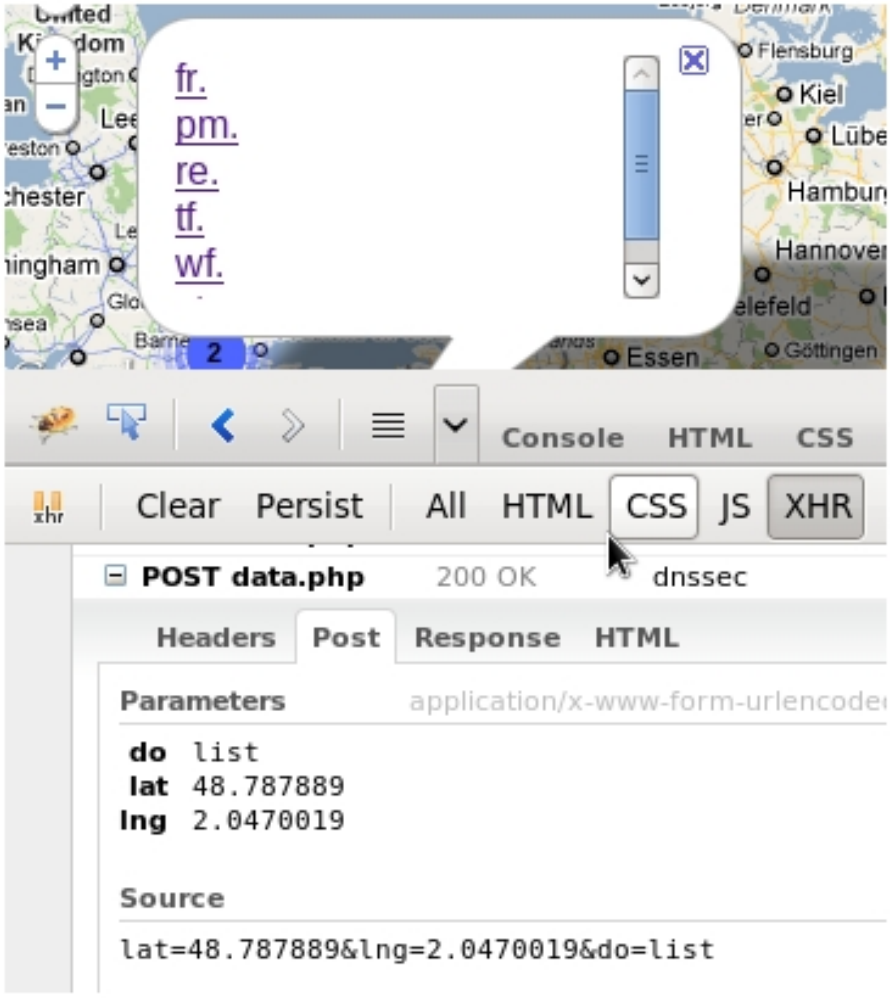
```
['fr.','pm.','re.','tf.','wf.','yt.']
```

V tomto prípade je už užívateľovi zaslaný zoznam domén pre daný bod na mape. Toto otvára možnosť postupne inkrementovať súradnice a iteratívne sa dotazovať serveru. Takýmto spôsobom by bolo možné získať všetky celý zoznam domén registrovaných pre doménu `cz`.

[1] `http://www.mozilla.com/en-US/firefox/new/`
[2] `http://getfirebug.com/`



Obrázek 6.1: Načítanie dát.



Obrázek 6.2: Dotaz o zaslanie domén nachádzajúcich sa na rovnakom bode.

Obrázek 6.3: Odmietnutie služby po prekročení limitu.

	Celkovo	**Zabezpečené**
TLD .	271	57
SLD cz.	772907	101234

Tabulka 6.1: Počet zabezpečených domén technológiou DNSSEC.

Proti takémuto prečítaniu celej mapky, sme implementovali obmedzenie počtu dotazov na dvetisíc na jednu IP adresu v časovom intervale dvadsaťštyri hodín. Pokiaľ je tento limit prekročný, užívateľovi sú odopreté služby aplikácie na dvadsaťštyri hodín a administrátorovi systému je zaslaný výstražný e-mail. Administrátor môže rozhodnúť o trvalom odopretí prístupu pre danú IP adresu. Informácie o konfigurácií je možné nájsť v prílohe B.

Pre testovacie účely sme limit znížili na desať dotazov. Po prekročení tohto limitu sa pri načítaní webovej aplikácie zobrazí hláška „Limit exceeded“. Prekročenie limitu je znázornené na obrázku 6.3.

Následne je možné IP adresu zablokovať. Prístup z blokovanej IP adresy je znázornený na obrázku 6.4

6.3 Zhodnotenie výsledkov

Výsledkom tejto práce má byť geografická mapa, ktorá mapuje stav nasadenia technológie DNSSEC do praxe. Základom celej tejto aplikácie je databáza, ktorá uchováva informácie o doménach a ich zabezpečení. Tieto informácie je možné použiť aj na vytvorenie niekoľkých štatistík.

Databáza v čase písania tejto práce obsahuje celkovo 772907 domén. Počet zabezpečených je zobrazený v tabuľke 6.1.

Je vidieť, že technológia sa stále zavádza. Napríklad v Európe, je stále veľká väčšina národných domén nezabezpečená. Rovnaká je situácia aj v Českej Republike, kde väčšina domén nie je zabezpečená pomocou technológie DNSSEC.

Obrázek 6.4: Blokovanie prístupu z určitej IP adresy

	NSEC	**NSEC3**
TLD .	11	46
SLD cz.	100553	681

Tabulka 6.2: Použitie záznamov NSEC a NSEC3.

Prvá špecifikácia technológie DNSSEC, používa záznam NSEC ako odpoveď na dotaz o neexistujúcej subdoméne. V časti 2.3.4 sme spomenuli, ako je možné pomocou záznamu NSEC, prečítať celý obsah zóny. Z tohto dôvodu niektoré organizácie odmietli zaviesť DNSSEC. Až nástupca NSEC3 rieši tento problém. Tabuľka 6.2 znázorňuje použitie záznamov NSEC a NSEC3.

Zaujímavé je, že niektoré TLD domény stále používajú NSEC. Najväčšia z nich je národná doména `us`. To znamená, že je možné získať kompletný obsah zóny `us`. Takto získané údaje sa dajú použiť napríklad na rozosielanie spamu.

6.4 Zhrnutie

V tejto kapitole sme sa venovali testovaniu aplikácie. Otestovali sme možnosti získavania informácií o technológií DNSSEC. Otestovali sme zabezpečenie aplikácie proti získaniu obsahu celej zóny `cz` a v závere sme zhodnotili výsledky, ktoré dosiahla aplikácia.

Zaujímavé sú informácie v akom pomere je počet zabezpečených a nezabezpečených domén v súčasti vzhľadom k súčasnej rastúcej kriminálnej aktivite na internete.

Kapitola 7

Záver

Cieľom tejto práce bolo navrhnúť a vytvoriť systém, ktorý by automaticky získaval a aktualizoval informácie a stave technológie DNSSEC v praxi a získané informácie zobrazoval na geografickej mape sveta.

Súčasťou tejto práce je prehľad technológie DNSSEC, ktorý sa hlavne zaoberá spôsobom podpisovania záznamov v systéme DNS. Obsahuje popis všetkých nových záznamov, ktoré DNSSEC zavádza do DNS a popis princípu zabezpečenia systému DNS pomocou technológie DNSSEC.

Vytvorený systém sa skladá z dvoch častí. Prvá časť je implementovaná v jazyku C++ a má za úlohu priebežne sledovať stav DNSSEC pri doménach TLD a SLD. Systém momentálne sleduje všetky domény TLD a z SLD len doménu `cz`. Systém je navrhnutý tak, aby vedel sledovať aj ďalšie národné domény. Informácie o doménach, ich zabezpečení a geografickej lokácií získava zo systému DNS, databázy Whois a služby Google Maps.

Druhá časť slúži na prezentáciu získaných na geografickej mape sveta. Implementovaná je ako webová aplikácia, ktorá pomocou Google Maps vykresľuje získané dáta na geografickú mapu sveta. Zobrazené informácie je možné filtrovať na základe niekoľkých parametrov. Webová aplikácie je napísaná v jazykoch PHP, HTML a JavaScript.

Poslednou časťou práce je testovanie systému. V testovaní sme sa zamerali hlavne na bezpečnosť. Zdroje informácií, ktoré systém používa sú voľne dostupné, až na počiatočný zoznam domén, ktoré sa majú sledovať. Tento zoznam poskytlo združenie CZ.NIC. Pri testoch sme sa pokúšali získať zoznam domén, ktoré systém obsahuje. Vo výsledku sme zistili, že technikami nám známym, nie je možné sa dostať k citlivým informáciam. Presnosť geolokácie je pomerne vysoká, ale obsahuje chyby. Nepresnosti tolerujeme, lebo cieľom tejto práce nie je nástroj na presnú geolokáciu domén, ale nástroj na sledovanie stavu zabezpečenia DNS pomocou technológie DNSSEC.

Výsledkom tejto práce je autonómny systém, ktorý sleduje stav nasadenia technológie DNSSEC do praxe. Systém bol vyvinutý v spolupráci so združením CZ.NIC a v súčasnosti je prevádzkovaný na adrese `http://dnssec-map.cz`.

Ďalší vývoj systému by mohol zahrňovať komplexnejšie sledovanie informácií o doménach a implementovanie ďalších parserov pre informácie získané z databázy Whois, aby podporoval čo najviac domén typu SLD.

Literatura

[1] Ben Laurie, Geoffrey Sisson, Roy Arends, David Blacka: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. RFC 5155, 2008, 2008-02-01 [cit. 2011-01-09].

[2] Google: Google Maps Javascript API V3 Reference. `http://code.google.com/apis/maps/documentation/javascript/reference.html`.

[3] Internet System Consortium: BIND 9.5 Administrator Reference Manua. `https://www.isc.org/software/bind/documentation/arm95#man.dig`.

[4] Joan Gargano, Ken Weiss: Whois and Network Information Lookup Service. RFC 1834, 1995, 1995-08-01 [cit. 2011-05-10].

[5] Leslie Daigle: WHOIS Protocol Specification. RFC 3912, 2004, 2004-09-01 [cit. 2011-05-08].

[6] Libor Dostálek, A. K.: *Velký průvodce protokoly TCP/IP a systémem DNS*. Computer Press, Praha, 2000, iSBN 80-7226-323-4.

[7] Mark Jeftovic: phpWhois -base class to do whois queries with php. `http://www.phpwhois.org/`.

[8] Maxmind: GeoLite City. `http://www.maxmind.com/app/geolitecity`.

[9] NLnet Labs: Ldns library documentation. `http://nlnetlabs.nl/projects/ldns/doc/index.html`.

[10] NLnet Labs: Unbound documentation. `http://www.unbound.net/documentation/libunbound.html`.

[11] P. Mockapetris : DOMAIN NAMES - CONCEPTS AND FACILITIES. RFC 1034, 1987, 1987-11-01 [cit. 2010-01-04].

[12] P. Mockapetris : DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, 1987, 1987-11-01 [cit. 2011-01-03].

[13] Roy Arends, Rob Austein, Matt Larson, Dan Massey, Scott Rose: DNS Security Introduction and Requirements. `http://tools.ietf.org/html/rfc4033`, 2005-03-01 [cit. 2011-01-06].

[14] Roy Arends, Rob Austein, Matt Larson, Dan Massey, Scott Rose: Protocol Modifications for the DNS Security Extensions. RFC 4035, 2005, 2005-03-01 [cit. 2011-01-06].

[15] Roy Arends, Rob Austein, Matt Larson, Dan Massey, Scott Rose: Resource Records for the DNS Security Extensions. RFC 4034, 2005, 2005-03-01 [cit. 2011-01-06].

[16] Tom Olzak: DNS Cache Poisoning: Definition and Prevention. `http://adventuresinsecurity.com/Papers/DNS_Cache_Poisoning.pdf`, 2006-03-01 [cit. 2010-01-04].

Dodatek A

Obsah CD

- `/src` - obsahuje zdrojové súbory aplikácie.
- `/doc/projekt.pdf` - technická správa vo formáte PDF.

Dodatek B

Inštalácia a použitie

Požiadavky na systém

Pre úspešné nainštalovanie aplikácie je potrebné aby systém obsahoval:

- PHP 5.3, ale vyššie
- Knižnicu *libunbound*
- Knižnicu *ldns*
- Knižnicu *curl*
- Knižnicu *sqlite3*

Inštalácia

Adresárová štruktúra:

- `/src`
  - `bin` - po konfigurácií bude obsahovať binárne súbory
  - `depend` - obsahuje PHP triedu *phpWhois* a databázu
    - `db` - zložka s databázov
    - `whois` - zložka s triedou na prácu s Whois
    - `keys` - súbor s bodom dôvery
  - `src` - obsahuje zdrojové súbory
  - `web` - obsahuje zdrojové súbory pre webovú aplikáciu

V zložke `/src` sa nachádzajú zdrojové súbory. Pomocou nástrojov *Autotools* bol vytvorený konfiguračný skript, ktorý po spustený skontroluje prítomnosť potrebných knižníc a vytvorí Makefile.

V zložke `/src` spustíme následujúcu sekvenciu príkazov:

- `./configure`
- `make`
- `./setup`

Posledný krok, pustenie skriptu `setup`, nie je povinné. Po preklade pomocou `make` je v zložke `/src/src` vytvorený binárny súbor `daemon`. Skript `setup` prekopíruje výsledné binárne súbory do zložky `bin` a spolu s nimi aj ďalšie potrebné časti ako databáza a PHP trieda *phpWhois*.

Inštalácia webovej aplikácií spočíva v skopírovaní zložky `web` do koreňovej zložky webu. Webová aplikácia obsahuje skript `config.php`, v ktorom je potreba nakonfigurovať hlavne cestu k databázy. Konfigurácia webovej aplikácie je vysvetlená v skripte `config.php`.

Použitie aplikácie

Serverovú časť je možné spúšťať s niekoľkými parametrami:

- -h - vytlačí nápovedu
- -d cesta_k_db - cesta k databázi, defaultná `db/db`
- -k cestak_ku_keys - cesta k súboru s kľúčmi. Defaultná `keys`.
- -a domena - vloží novú doménu do databázi.
- -r domena - odstráni doménu z databázi.
- -g - zapne geolokáciu.
- -v - zapne testovacie výpisy.

Príklad použitia je následovný:

```
./dnssec-daemon -g -k ta.keys -d databas
```

Zložka `web` obsahuje skript `distribute.php`, ktorý slúži k distribuovaniu geolokácie na viacerých klientov. Použitie je jednoduché a treba rozoslať odkaz na tento skript užívateľom. Takto je geolokácia distribuovaná na viacej uzlov s rôznymi IP adresami a pre každý platí limit od Google Maps 2500 dotazov za dvadsaťštyri hodín na jednu IP adresu. Príklad použitia tohto skriptu je následovný:

```
http://dnssec-map.cz/distribute.php
```

Po načítaní sú klientovi pridelené adresy na geolokáciu a výsledky geolokácie sú zasielané naspäť na server.

V súbore `ip-block`, je možné nastaviť IP adresy, ktoré majú zamedzený prístup k službe. IP adresy sú na zadané na zvlášť riadkoch.

Dodatek C

Popis databázových tabuliek

V tejto časti sú popísané databázové tabulky použité v aplikácií.

domains		
`domain`	TEXT	názov domény
`secured`	INTEGER	príznak zabezpečenia domény
`lastUpdate`	INTEGER	čas poslednej aktualizácie
`lat`	REAL	zemepisná šírka
`lng`	REAL	zemepisná dĺžka
`revision`	INTEGER	počet aktualizácií danej domény
`level`	INTEGER	informácia o type domény
`distributed`	INTEGER	príznak o distribúcií na geografickú lokáciu

Tabulka C.1: Hlavná tabuľka na ukladanie domén - `domains`.

rrDNSKEY		
idRRdnskey	INTEGER	primárný kľúč
domain	TEXT	doména ku ktorej patrí DNSKEY
ttl	INTEGER	doba platnosti záznamu
dnskeyEntry	TEXT	verejný kľúč
flag	INTEGER	príznak kľúča KSK, alebo ZSK
algorithm	INTEGER	typ algoritmu

Tabulka C.2: Štruktúra databázovej na ukladanie verejných kľúčov `rrDNSKEY`.

rrNS		
idRRns	INTEGER	primárný kľúč
domain	TEXT	doména ku ktorej patrí DNSKEY
ttl	INTEGER	doba platnosti záznamu
nsEntry	TEXT	autoritatívny server DNS

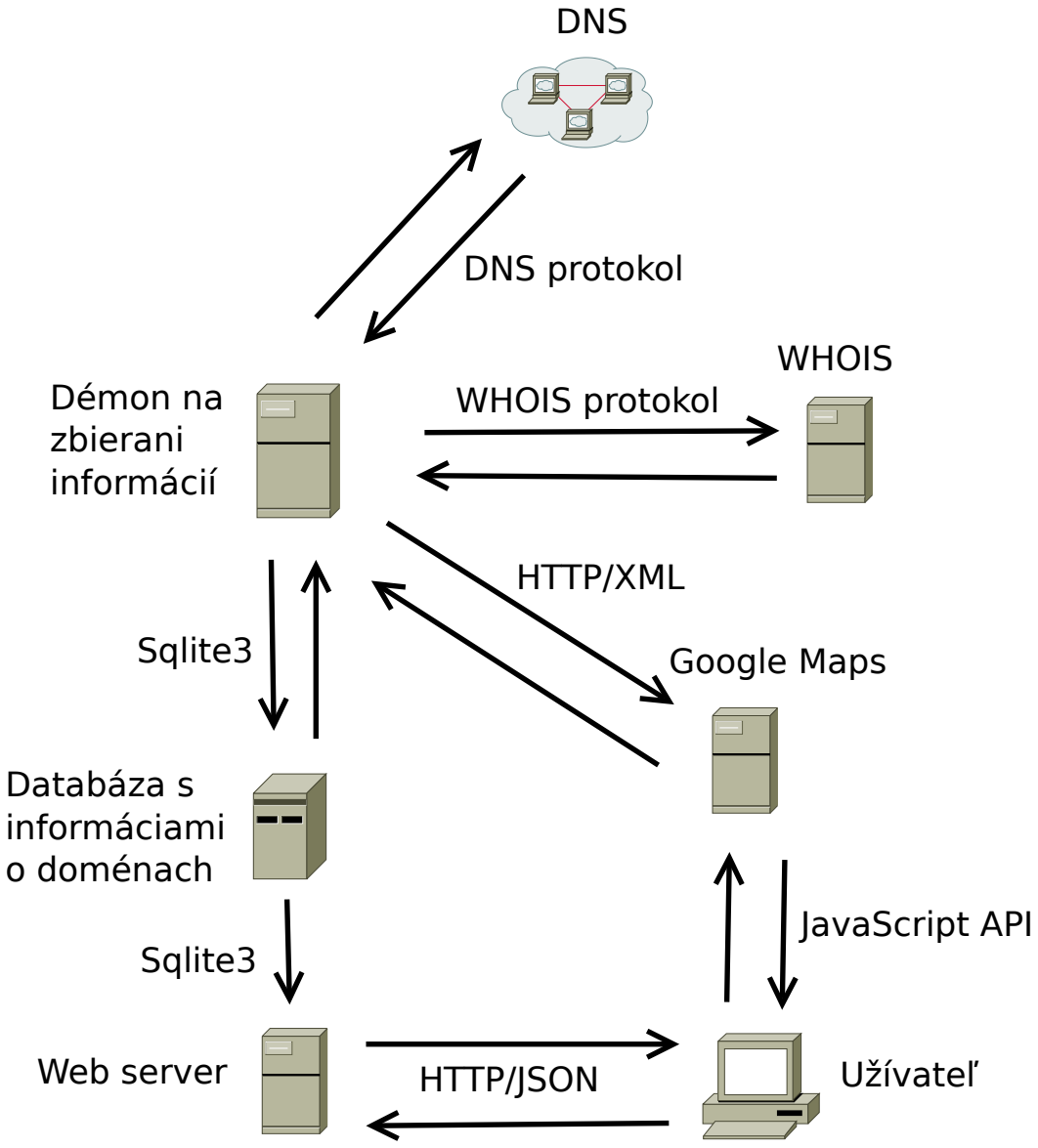
Tabulka C.3: Štruktúra databázovej na ukladanie záznamov NS `rrNS`.

rrRRSIG		
idSig	INTEGER	primárný kľúč
idRR	TEXT	kľúč do tabuľky rrDNSKEY a rrNS
ttl	INTEGER	doba platnosti záznamu
validUntil	TEXT	doba platnosti
validFrom	TEXT	príznak kľúča KSK, alebo ZSK
idKey	INTEGER	id kľúča ktorým bol vytvorený
algoritm	INTEGER	algoritmus, ktorým bol podpis vytvorený

Tabulka C.4: Štruktúra databázovej na ukladanie podpisov `rrRRSIG`.

Dodatek D

Diagram aplikácie



Obrázek D.1: Architektúra aplikácie.