3. Systém DNS

Adresování a překlad adres je nezbytná součást Internetu, bez které se žádná komunikace mezi uživateli neobejde. Že nefunguje překlad doménových jmen, zjistíme během krátkého okamžiku, kdy nejsme schopni načíst jedinou webovou stránku nebo poslat email.

Tato kapitola se zabývá architekturou systému DNS (Domain Name System). Seznámíme se se základními stavebními prvky systému, způsobem uložení dat a přístupu k nim. Popíšeme si také základní typy záznamů, které systém DNS spravuje. Popis systému je rozšířen o některé základní bezpečnostní problémy DNS a ukazuje, jak zajistit DNS proti zneužití. Zmíníme se také o využití DNS pro IP telefonii či lokalizaci a sdílení zdrojů.

3.1 Služba DNS

Základním úkolem služby DNS je mapování (převod) doménových adres (např. pcmatousek.fit.vutbr.cz) na IP adresy (147.229.12.101). Doménové adrese se často říká také doménové jméno (domain name). V první kapitole jsme si řekli, že každé síťové rozhraní v Internetu obsahuje jednoznačný identifikátor, jímž je IP adresa. Pro uživatele je identifikace počítačů pomocí IP adres nevhodná. Málokterý uživatel Internetu si pamatuje nějakou IP adresu. Na druhou stranu není výhodné, aby síťová zařízení pracovala s textovými řetězci. 32-bitové číslo (či 128-bitové u IP adres verze 6) je velmi vhodný formát pro porovnávání či prefixové vyhledávání.

Už při zavedení IP adres v 70. letech 20. století se začaly používat jmenné ekvivalenty, které se vždy překládaly na IP adresu. Právě systém DNS obsahuje celosvětovou databázi IP adres a jejich srozumitelných ekvivalentů, doménových jmen.

Služba DNS tedy obsahuje databázi všech doménových adres a příslušných IP adres. Definuje také, jak budeme přistupovat k těmto datům. Protože jde o velmi rozsáhlou databázi, je distribuovaná na více počítačů, kde běží speciální servery, kterým se říká *nameservery*, doménové servery (jmenné servery) nebo také servery DNS. IP adresu zjišťujeme z doménového jména dotazem na server DNS. Proces vyhledávání v systému DNS nazýváme rezoluci či rozlišení jména (*name resolution*).

Systém DNS využívají všechny aplikační protokoly (např. HTTP, SMTP, FTP) pro překlad doménových adres na IP adresy. To je vždy první aktivita, kterou aplikace udělá poté, co ji požádáme o připojení na vzdálenou službu a zadáme adresu v podobě doménového jména. Služba si nejprve požádá o překlad na IP adresu. Teprve poté může dojít k navázání spojení.

Mezi základní služby, které poskytuje systém DNS, patří:

- překlad doménových adres na IP adresy (s využitím záznamů typu A, AAAA),
- překlad IP adres na doménové adresy (s využitím záznamů typu PTR),
- překlad aliasů počítačů, překlad na tzv. kanonická jména (s využitím záznamů CNAME),
- určení poštovního serveru pro danou doménu (s využitím záznamů typu MX),
- podpora rozložení zátěže mezi více aplikačních serverů (rotace záznamů),
- sdílení informace v rámci globálního prostoru jmen či
- delegování správy domén na jednotlivé subjekty (pomocí záznamů typu NS).

O jednotlivých typech překladů si podrobněji povíme v kapitole 3.3.

3.1.1 Historie

Pro mapování doménových jmen na IP adresy se původně používal jeden soubor HOSTS.TXT, který spravovala organizace NIC (Network Information Center). Uložená data (mapování doménová adresa – IP adresa) se šířila na další počítače pravidelnými aktualizacemi pomocí služby FTP. S rozšířením lokálních sítí se administrace záznamů přenesla na lokální úroveň. Aby byly změny viditelné i na Internetu, musely se lokálně mapované adresy předávat do NIC. Postupně vzniká koncept distribuované databáze založené na doménových serverech.

Primárním cílem DNS bylo vytvořit konzistentní prostor jmen (name space), který se použije pro odkazování na síťové zdroje. Databáze měla být distribuovaná s možností ukládání lokálních kopií pro zvýšení výkonu systému. Systém DNS byl navržen tak, aby byl přístupný nejen pro IP, ale i pro jiné rodiny protokolů a aplikace. Hlavním úkolem systému DNS je tedy zajistit snadný přístup k informacím uloženým v DNS ze všech počítačů připojených k síti. Následující body shrnují požadavky na databázi DNS:

- Velikost databáze závisí na počtu připojených počítačů, který v budoucnu poroste.
- Většina dat v databázi se nemění často. Systém však musí provádět drobné změny celkem rychle (během několika minut).
- Přístup k datům je důležitější než jejich aktuálnost. Když není možné data aktualizovat, je třeba zajistit alespoň běh služby.
- Originální data se udržují lokálně v takzvaných primárních záznamech (master files). Odtud se šíří na další počítače pomocí systému DNS. Formát záznamů je textový, spravuje je lokální administrátor.

Systém DNS obsahuje seznam všech registrovaných doménových adres a jejich mapování na IP adresy. V prostoru IPv4 to může být až 2^{32} adres, což je hodně. Samozřejmě není potřeba vytvářet pro všechny IP adresy odpovídající doménovou adresu. Například pro IP adresy třídy D a E to nemá smysl. Také pokud je IP adresa použita pouze pro systémové služby, nemá smysl k ní vymýšlet doménové jméno. Naopak, pro některé IP adresy může naopak existovat více doménových jmen, jak si ukážeme v kapitole 3.3.

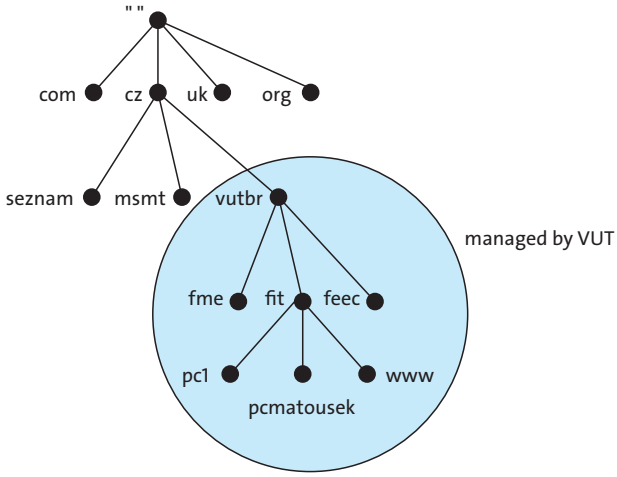
3.2 Architektura systému DNS

Systém DNS se skládá ze tří hlavních komponent – prostoru doménových jmen, serverů DNS a resolveru [27]. Prostor doménových jmen zahrnuje strukturu, uspořádání a přístup k datům v systému DNS. Servery DNS ukládají tato data ve svých lokálních databázích. Resolver slouží pro přístup k datům v systému DNS. Jednotlivé komponenty systému DNS si nyní podrobněji popíšeme.

3.2.1 Prostor doménových jmen (domain name space)

Jak jsme si už řekli, doménových jmen i IP adres je poměrně hodně. Z důvodu uložení a efektivního vyhledání se logický prostor všech doménových jmen ukládá v systému DNS speciálním způsobem. Systém DNS tvoří databáze hierarchicky uspořádaná jako kořenový strom doménových jmen, viz obrázek 3.1. Tato stromová struktura připomíná

uspořádání souborového systému v Unixu. Z pohledu algebry se jedná o acyklický graf – strom, který obsahuje uzly a hrany.



Obrázek 3.1: Hierarchické uspořádání doménových jmen v DNS

Kořen stromu DNS se nazývá `the root`. Uzly stromu jsou pojmenovány textovým řetězcem (bez teček) délky max. 63 znaků. Někdy se uzlům říká domény, ale toto označení je nepřesné, jak si za chvilku ukážeme. Názvy uzlů jsou pouze součástí doménových jmen. Uzly, které mají stejného bezprostředního předchůdce ve stromu, musí mít různý název, aby zaručili rozlišení uzlů (např. fit.vutbr.cz., feec.vutbr.cz., fme.vutbr.cz.). Kořen strom má speciální jméno – řetězec nulové délky. Cesta od listu ke kořenu slouží k uložení (a také vyhledávání) doménových adres.

Doména je podstrom v grafu doménových adres. *Doménové jméno* (domain name) je cesta mezi uzlem, který tvoří vrchol domény, a kořenem stromu DNS. Příkladem může být doménové jméno (obvykle se říká pouze doména) `vutbr.cz.`, kterou tvoří všechny uzly v prostoru doménových adres s relativním kořenem v uzlu `vutbr`, který patří do podstromu `cz.` Doména s vrcholem v uzlu ve vzdáleností jedna od kořene grafu se nazývá doména první úrovně (někdy též TLD, Top Level Domain, doména ve vzdálenosti dva od kořene stromu DNS je doménou (či subdoménou) druhé úrovně a tak dále. Listy stromu označují konkrétní síťová zařízení patřící do dané domény, například servery `pcmatousek`, `www`, `pc1`, v doméně `fit.vutbr.cz`.

Plné (absolutní) doménové jméno (`FQDN`, `Fully Qualified Domain Name`) je posloupnost jmen uzlů na cestě až ke kořeni stromu DNS. Tato posloupnost jmen se odděluje tečkami, například „`www.fit.vutbr.cz.`“. Všimněte si, že absolutní adresa vždy končí tečkou. Ve skutečnosti je tečka pouze oddělovač a adresa končí jménem kořene DNS stromu, což je řetězec nulové délky. Doménová jména bez tečky se nazývají relativní a jsou interpretována k relativní doméně, ve které se nachází. Když například na fakultním serveru zadáme `ping www`, kde `www` je relativní doménové jméno školního webového serveru, bude se adresa interpretovat obvykle v doméně `fit.vutbr.cz.`[1] Ukončující tečku při práci na Internetu obvykle nepíšeme, neboť ji při překladu doplňuje

1 Záleží na konfigurace resolveru.

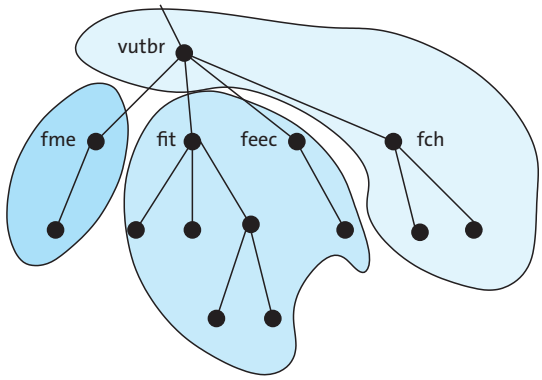
aplikace. Při práci s DNS, jako je například konfigurace serveru DNS či klienta DNS, je ukončující tečka nezbytná.

Databáze DNS obsahuje jednotlivá doménová jména uspořádaná do podstromů (domén). Správa domén (přidávání koncových uzlů, rušení uzlů, vytváření subdomén), je decentralizovaná a deleguje se na další organizace. Například doménová jména počítačů `pc1.fit.vutbr.cz`, `pc2.fit.vutbr.cz`, `pc3.fit.vutbr.cz`, `www.fit.vutbr.cz`, `ns.fit.vutbr.cz` atd., patří do domény `fit.vutbr.cz.`, kterou spravuje Fakulta informačních technologií VUT v Brně na svém lokálním serveru DNS `kazi.fit.vutbr.cz`.

Uspořádání prostoru doménových adres

Prozatím jsme se bavili o uspořádání prostoru doménových jmen, který má strukturu hierarchického stromu. Tento strom však není uložen na jednom místě v jedné databázi. DNS je systém hierarchický a decentralizovaný. Jednotlivé části podstromu celého prostoru doménových adres jsou fyzicky uloženy na lokálních serverech DNS, které dohromady tvoří systém DNS. Data o objektech v prostoru DNS (obecně zdrojích, resources) netvoří pouze doménová jména. Záznamy obsahují také informace o primárních a sekundárních serverech DNS, správcích domén, poštovních serverech a podobně. Veškeré tyto informace jsou zapisovány v textovém formátu, který definují RFC 1034 [27] a RFC 1035 [28]. Ukládají se do takzvaných *záznamů DNS* (RR, resource records). Přehled typů záznamů DNS a podrobnější popis jejich formátu je uveden v kapitole 3.3.

Fyzické části prostoru DNS, které jsou pod jednou správou, se nazývají *zóny*. Zóna není totožná s doménou. Například poskytovatel síťového připojení může spravovat více domén (a subdomén) tvořící jednu zónu. Nebo naopak velká doména (např. `edu`) je rozdělena na více částí a umístěna na více serverech DNS. Například doména `vutbr.cz` může být rozdělena do více zón, které pokrývají různé části doménového prostoru a které spravují různé subjekty, viz obrázek 3.2.



Obrázek 3.2: Příklad uspořádání DNS prostoru do zón

Zatímco doména je část prostoru adres, který má společný suffix (např. doména vutbr.cz), zóna je tvořena částmi prostoru uloženého na konkrétním serveru. Zónu mohou tvořit například subdomény fit.vutbr.cz a feec.vutbr.cz pod jednou správou. Zóna může obsahovat celou doménu nebo jen část domény. Existují také dva speciální typy zón:

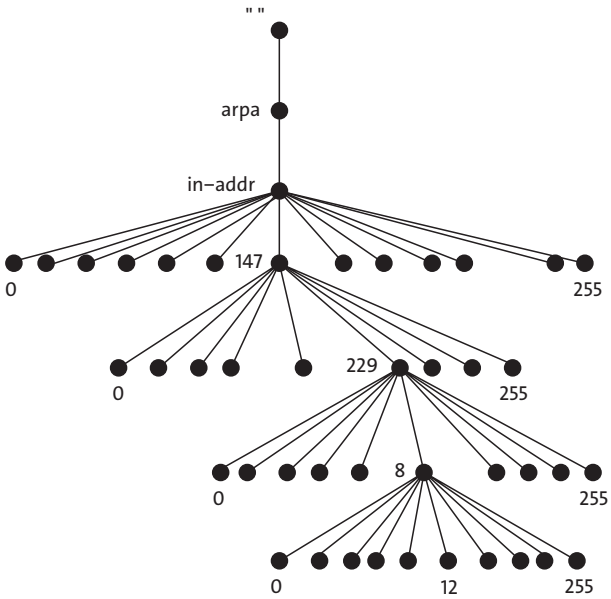
- Zóna `stub` obsahuje pouze informace o tom, které servery subdoménu obsluhují. Sama neobsahuje žádná data.
- Zóna `hint` obsahuje seznam kořenových serverů DNS.

Reverzní mapování adres

Jednou z důležitých funkcí systému DNS je *reverzní (zpětné) mapování* IP adres na doménová jména. Prozatím jsme se bavili o tzv. *přímém mapování*, které je základem DNS a kdy se překládá doménové jméno na příslušnou IP adresu. Reverzní adresování slouží ke zpětnému dohledání doménových adres pro IP adresy, které jsou uloženy například v logovacích souborech. Používá se například při autorizaci počítače, kdy kontrolujeme, zda IP adresa stanice má záznam v systému DNS. Pokud doménové jméno chybí, může to systém vyhodnotit jako použití podvržené IP adresy a komunikaci odmítnout. Toho využívají například poštovní servery v boji proti spamu.

Reverzní mapování DNS adres není tak jednoduché, jak by se na první pohled zdálo. Záznamy v datovém prostoru DNS jsou indexovány podle doménových adres. Pokud známe doménovou adresu, můžeme lehko vyhledat IP adresu. Vyhledání doménové adresy pro známou IP adresu by však znamenalo úplné prohledání celého stromu, kdy v každém uzlu porovnávám hodnotu uzlu se hledanou IP adresou (tzv. asociativní vyhledávání).

Tento způsob je pro praktické použití nerealizovatelný. Proto existuje v datovém prostoru DNS jedna speciální doména, jejíž uzly jsou pojmenovány čísly reprezentujícími IP adresu ve čtyřbytovém dekadickém formátu odděleným tečkami (například 147.229.8.12). Doména, ve které jsou tyto IP adresy uloženy, se nazývá `in-addr.arpa.`, viz obr. 3.3.



Obrázek 3.3: Reverzní mapování v DNS

Jak je z obrázku zřejmé, doména `in-addr.arpa.` obsahuje 256 subdomén první úrovně, které odpovídají prvnímu bytu IP adresy. Každá z těchto subdomén obsahuje dalších 256 subdomén druhého řádu, pak třetího řádu až do hloubky čtyři. IP adresy ve stromu jsou uspořádány od nejvyššího bytu k nejnižšímu (tj. jak se čtou, zleva doprava), což je opačný postup než u doménového jména. Doménová jména se ukládají od nejvyšší domény, která se píše až na konec.

Například doménová adresa `kazi.fit.vutbr.cz.` začíná nejméně obecnou položkou `kazi`, která je ve stromu DNS uložena nejdále od kořene. Pro IP adresu je postup opačný, záznam DNS pro IP adresu `147.229.8.12` je `12.8.229.147.in-addr.arpa.`, což je obrácený zápis než vlastní IP adresa.

Toto uspořádání umožňuje lépe delegovat správu subdomény na vlastníka odpovídajícího prostoru IP adres. Například VUT v Brně je odpovědné za doménu `229.147.in-addr.arpa.`, neboť má přidělený prostor adres B `147.229.0.0/16`. Tuto reverzní doménu může ovšem dále rozdělit na podsítě a jednotlivé části podstromu přidělit do správy jejich uživatelů, například fakult.

Registrace a správa domén

Databáze DNS je spravovaná hierarchicky a distribuovaná na velké množství serverů. Koordinaci systému DNS zajišťuje organizace ICANN (Internet Corporation for Assigned Names and Numbers), `www.icann.org`. Ta je zodpovědná za správu, přidělování a uložení doménových jmen. Každý záznam o doménové adrese musí být jedinečný, aby uživatelé mohli najít odpovídající IP adresy. Organizace ICANN deleguje části prostoru doménových jmen na tzv. akreditované registrátory doménových jmen (ICANN-Accredited Registrar) [15]. ICANN spravuje domény nejvyšší úrovně. Domény nižších úrovní jsou delegovány na další subjekty. Doménou první úrovně mohou být buď národní domény (ccTLD, country code TLD) , např. .uk, .cz, nebo generické domény (gTLD, generic top-level domains), viz tabulka 3.1. Seznam registrátorů generických domén první úrovně (TLD, Top-level domains) je na `http://www.internic.net/regist.html`.

Doména	Použití	Doména	Použití
com	komerční organizace	aero	letecký průmysl
edu	vzdělávací organizace v USA	biz	obchodní organizace
gov	vládní agentury v USA	coop	družstva
mil	doména US. Military	museum	muzea
net	organizace pro správu sítí	pro	profesionálové (doktoři, právníci)
int	mezinárodní organizace	info	informační místo
org	nevýdělečné organizace	name	jednotlivci

Tabulka 3.1: Generické domény první úrovně

Některé z výše uvedených domén, například `.edu`,`.gov`.,`.int`,`.mil`, mají předem definované použití a mohou je využívat pouze určité subjekty. Domény `aero`, `biz`, `coop`, `museum`, `pro`, `info`, `name` byly přidány v roce 2000, aby se zvýšil počet domén nejvyšší úrovně a také se snížil tlak na doménu `.com`. Od té doby se objevují další domény. Jednou z kategorií jsou tzv. sponzorované domény, které spravuje určitý subjekt pro speciální použití, například `.asia`, `.cat`, `.jobs`, `.mail`, `.mobi`, `.post`, `.tel`, `.travel`, viz http://www.icann.org/tlds/. Standard RFC 2606 [10] zavedl také v roce 1999 rezervované zóny pro speciální účely, viz tabulka 3.2.

Doména	Použití
.test	testovací účely
.example	vytváření dokumentace a příkladů
.invalid	navozování chybových stavů
.localhost	lokální (softwarové) komunikační rozhraní
local	intranet

Tabulka 3.2: Rezervované domény podle RFC 2606

Speciální doménou nejvyšší úrovně TLD je i `.arpa`, která se používá pro účely správy síťové infrastruktury. Doménu `.arpa` spravuje organizace IANA. Použití domény je definováno standardem RFC 3172 [12] a dalšími standardy, které jsou spolu s využitím domény `arpa` uvedeny v tabulce 3.3.

Doména	Použití	Odkaz
e164.arpa	mapování telefonních čísel E.164 na URI	RFC 6116 [4]
in-addr.arpa	mapování IPv4 adres na doménové adresy	RFC 1035 [28]
ip6.arpa	mapování IPv6 adres na doménové adresy	RFC 3596 [35]
iris.arpa	lokalizace služeb Internet Registry Information Services	RFC 4698 [11]
uri.arpa	překlad URI podle systému Dynamic Delegation Discovery System	RFC 3405 [22]
urn.arpa	překlad URN podle systému Dynamic Delegation Discovery System	RFC 3405 [22]

Tabulka 3.3: Speciální domény `.arpa`

Národní domény nejvyšší úrovně (ccTLD) registruje národní správce domény. Název domény (.cz, .de, .uk) vychází ze standardu ISO-3166 [1]. Správci domén jsou uvedeni v databázi organizace IANA. Např. pro Českou republiku je to organizace CZ-NIC – viz záznam v databázi IANA. V současné době CZ-NIC udržuje centrální databázi, změny do ní však vkládají oprávnění registrátoři – viz http://www.nic.cz/page.php?sid=9.

3.2.2 Server DNS (nameserver)

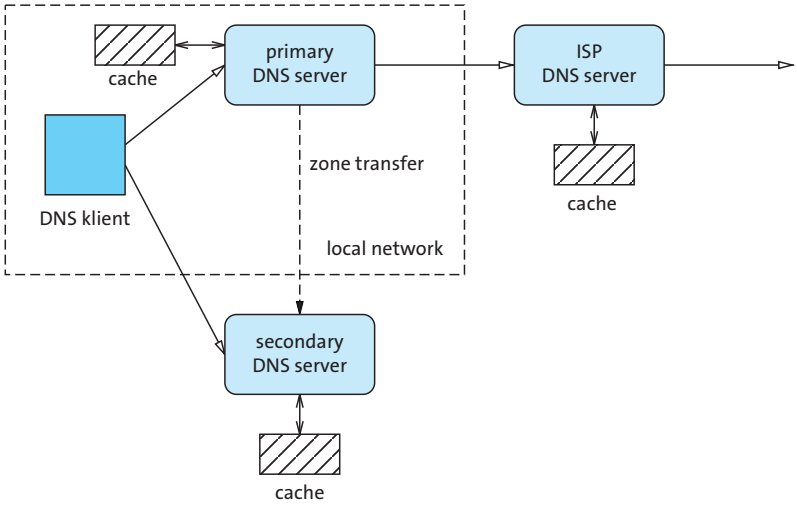
Další částí systému DNS jsou servery DNS (nameservers). Server DNS je aplikace, která uchovává data z prostoru doménových jmen. Tento prostor je rozdělen do zón a umístěn na jednotlivé servery DNS. Základním úkolem serveru DNS je odpovídat na dotazy směřující na databázi DNS. Server DNS uchovává data ve formě množiny záznamů DNS (resource records). Záznamy jsou uloženy v lokálním souboru nebo si je server načte z jiného serveru DNS pomocí přenosu zón. Informace, které server DNS spravuje a za které je zodpovědný, se nazývají se autoritativní. Specifikace DNS [27] definuje dva základní typy serverů – primární a sekundární.

- **Primární server DNS** (master, primary nameserver)
  Primární server obsahuje úplné záznamy o doménách, které spravuje. Tyto záznamy jsou uloženy lokálně v souboru. Server poskytuje autoritativní (tj. vždy přesné) odpovědi pro tyto domény. Pro každou doménu musí existovat právě jeden primární nameserver.

- **Sekundární server DNS** (slave, secondary nameserver)
  Sekundární server získává data od primárního serveru. Soubor, který obsahuje databázi konkrétní domény (či subdomény), se nazývá *zónový soubor*. Proces přenosu zónových souborů z primárního serveru na sekundární se nazývá *přenos zón* (zone transfer). Přenos zón je podrobněji popsán v kapitole týkající se komunikace DNS. Sekundární server musí zajistit pravidelný přenos zónových dat a aktuálnost dat. Sekundární server je také autoritativní server pro danou doménu.

- **Záložní server DNS** (caching-only nameserver)
  Záložní server pracuje jako proxy server. Přijímá dotazy od klientů a přeposílá je dalším serverům DNS. Když záložní server dostane odpověď na svůj dotaz, uchová si ji a použije ji v budoucnosti. Záložní servery poskytují neautoritativní odpovědi, tj. odpovědi, které mohou být neúplné a neaktuální. Zrychlují však proces rezoluce doménového jména.

Zóna DNS je souvislá část jmenného prostoru DNS, pro kterou je daný server DNS autoritativní. Server může obsahovat více zón. Každá zóna obsahuje záznamy DNS pro danou část jmenného prostoru.



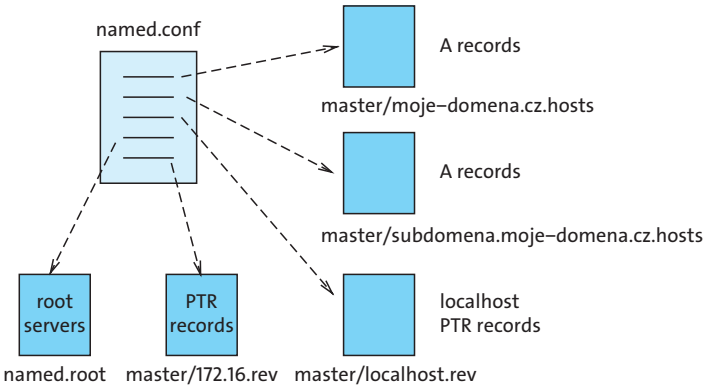
Obrázek 3.4: Typy DNS serverů

Platnost záznamů DNS na sekundárním a záložním serveru je časově omezena. Hodnota expirace je uvedena u každého záznamu. Pokud dojde k expiraci záznamu, musí server daný záznam smazat ze své databáze, případně si ho znovu načíst z primárního serveru. Pokud by neprovedl aktualizaci, může dojít k situaci, kdy dva různé servery DNS odpoví na stejný dotaz různým způsobem, což vede ke ztrátě konzistence dat.

Zejména při změně v záznamech DNS (změna IP adresy, přidání nového záznamu apod.) je potřeba počítat s tím, že rozšíření změn trvá v řádu hodin, což je doba, kdy dojde k expiraci původních záznamů a načtení nových. Na druhou stranu by nastavení krátké doby expirace vedlo k častým dotazům na obnovení záznamu a přetížení autoritativních serverů. Protože změny se šíří v síti DNS pomalu, je nutné v případě, kdy potřebujeme zjistit aktuální hodnotu nějakého záznamu v DNS, kontaktovat přímo primární nebo sekundární server DNS.

Konfigurace serveru DNS

V této části si ukážeme příklad konfigurace jednoduchého serveru DNS. Pro demonstraci konfigurace použijeme hojně rozšířenou distribuci serveru DNS BIND (Berkeley Internet Name Domain) a aplikaci named. Základní konfigurace serveru DNS je tvořena pěti základními soubory, viz obr. 3.5:



Obrázek 3.5: Příklad konfigurace serveru DNS BIND

named.conf je základní konfigurační soubor serveru DNS. Obsahuje informace o doménách a zónových souborech, které server obsahuje, na jakém rozhraní odpovídá na dotazy apod. Zároveň obsahuje informaci o primárním či sekundárním serveru. Obsahuje též příkazy acl, include, key, logging, options, server či zone pro filtrování dotazů, podepisování, logování apod.

```
options {
        directory "/etc/namedb";
        listen-on { 127.0.0.1; 172.16.12.1;};
        forwarders { 147.229.8.12; };
};
zone "." {
        type hint;
        file "named.root";        // odkaz na kořenové nameservery
};
zone "0.0.127.IN-ADDR.ARPA" {    // primární server pro loopback
        type master;     // (reverzní záznam)
        file "localhost.rev";
};
zone "moje-domena.cz" {          // primární server pro doménu
        type master;
        file "moje-domena.cz";
};
zone "16.172.in-addr.arpa" {     // primární server, reverzní záznamy
        type master;             // pro adresy od 172.16.0.0
        file "172.16";
};
zone "moje-domena.cz" {          // sekundární server pro doménu
```

```
        type slave;
        file "moje-domena.cz";
masters {172.16.12.1;};
}; zone "16.172.in-addr.arpa" {  // sekundární server, reverzní záznamy
        type slave;              // pro adresy od 172.16.0.0
        file "172.16";
masters {172.16.12.1;};
};
```

named.root obsahuje seznam kořenových serverů DNS. Tento soubor obsahuje IP adresy kořenových serverů DNS, které používá v případě, že neumí odpovědět na zadaný dotaz.

```
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:BA3E::2:30
;
; formerly NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     192.228.79.201
...
```

named.local, localhost.rev slouží pro překlad adresy localhost (loopback).

```
$TTL  3600
@             IN   SOA   ns.moje-domena.cz. root.pc1.moje-domena.cz.(
                                 20050311 ; Serial
                                 3600     ; Refresh
                                 900      ; Retry
                                 3600000  ; Expire
                                 3600 )   ; Minimum
              IN   NS    ns.moje-domena.cz.
1             IN   PTR   localhost.moje-domena.cz.
```

Přímé zónové soubory obsahují záznamy typu SOA, NS, A, MX, CNAME a další.

```
@ IN SOA ns.moje-domena.cz hostmaster.moje-domena.cz (
                                 20050809 ; Serial
                                 28800    ; Refresh 8 hours
                                 7200     ; Retry 2 hours
                                 604800   ; Expire 7 days
                                 86400 )  ; Maximum TTL 1 day
              IN NS ns.moje-domena.cz.
              IN NS ns.muj-provider.cz.
ns            IN A 192.168.10.1
...
```

Reverzní zónové soubory obsahují záznamy typu SOA, NS či PTR.

```
@ IN SOA ns.moje-domena.cz hostmaster.moje-domena.cz (
                                20050809 ; Serial
                                28800    ; Refresh 8 hours
                                7200     ; Retry 2 hours
                                604800   ; Expire 7 days
                                86400 )  ; Maximum TTL 1 day
             IN NS ns.moje-domena.cz.
1            IN PTR ns.moje-domena.cz.
...
```

Při vytváření zóny je potřeba vytvořit minimálně dva záznamy: záznam SOA pro specifikaci správy domény a záznam NS, který ukazuje na autoritativní server dané zóny. Do zóny lze přidávat i další záznamy typu A, CNAME, MX, SRV, PTR a podobně.

Správnou konfiguraci serveru DNS lze ověřit následujícími kroky:

1. Test spuštění DNS serveru: příkaz „`/etc/rc.d/named status`“, informace v souboru `/var /log/messages` či výpis procesů příkazem „`ps ax | grep named`“.
2. Kontrola konfiguračního souboru: „`named-checkconf <filename>`“, kontrola zónového souboru: „`named-checkzone zone <filename>`“.
3. Testování funkčnosti programy `dig`, `host`, `nslookup`, či `rndc` (remote name server control).

Ne všechny informace o doménových jménech se musí nutně načítat ze systému DNS. Z historických důvodů existuje na unixových strojích soubor `/etc/hosts`, kde může být uveden statický překlad doménových jmen na IP adresy. Funkce knihovny BSD sockets určené pro překlad jmen `gethostbyname()` a `gethostbyaddr()` používají oba zdroje. Pořadí zdrojů, kam směřuje dotaz na překlad, specifikuje soubor `/etc/host.conf`, viz následující ukázka konfigurace.

```
order hosts,bind
multi on
```

Tato konfigurace určuje pořadí vyhledávání v systému DNS. Říká, že se nejprve hledají informace v souboru `/etc/hosts` a teprve potom, když nedostaneme požadovanou informaci, se zavolá program `bind`, viz `man host.conf`.

3.2.3 Resolver

Resolver je klientský program, který se dotazuje na data uložená v systému DNS. Uživatelské programy, které potřebují informace z DNS, přistupují k těmto datům pomocí resolveru. Základním úkolem resolveru je

- posílat dotazy na servery DNS,
- interpretovat odpovědi od serveru (přijaté záznamy, chybová hlášení),
- předat informace uživatelskému programu, který o data žádal.

Resolver musí být schopen přistupovat alespoň k jednomu serveru DNS. Od serveru získá přímo hledanou odpověď nebo mu server vrátí odkaz na další server, kde je hledaná informace uložena. Resolver tak může přeposílat dotazy dalším serverů.

Resolver je obvykle implementován jako systémová rutina, která je součástí operačního systému (viz `man resolver`) a ke které přímo přistupují uživatelské programy, například funkce knihovny BSD socket `gethostbyname()`, `gethostbyaddr()`nebo programy `nslookup` či `dig`.

Konfigurace resolveru

Konfigurace resolveru obsahuje informace o primárním a sekundárním doménovém serveru, aktuální doméně, způsobu vyhledávání apod. U unixových systémů je konfigurace uložena v souboru `/etc/resolv.conf`. Soubor obsahuje několik částí:

nameserver obsahuje IP adresu serveru DNS.
Jedná se o IP adresu serveru DNS, který odpovídá na dotazy. Položka může obsahovat více hodnot. Resolver volá servery podle uvedeného pořadí. Pokud neexistuje žádný záznam, dotaz se posílá lokálnímu počítači.

domain definuje lokální doménu.
Položka obsahuje hodnotu implicitní domény. Resolver přidá toto jméno ke každému názvu počítače, který není zadán jako plně kvalifikované. Například k položce `www` přidá jméno a `fit.vutbr.cz.`, které je uvedeno v sekci `domain`. Tuto položku také využívá funkce `gethostname()`.

search definuje doménu pro vyhledávání.
Pokud název počítače neobsahuje plně kvalifikované jméno ukončené tečkou, doplní se hodnota v položce `search` nebo záznam `domain`. Obvykle obsahuje pouze jednu položka – název lokální domény. Pokud obsahuje více položek, provádí se vyhledávání ve více doménách (např. vutbr.cz a fit.vutbr.cz).

sortlist určuje pořadí sítí.
Tato hodnota obsahuje preferované pořadí sítí. Pokud odpověď na DNS obsahuje více IP adres hledaného záznamu, záznamy se přerovnají podle zde uvedeného pořadí. Aplikuje se na hodnoty vrácené funkcí `gethostbyname()`.

options obsahuje další volby.
Toto pole obsahuje další volby, které lze nastavit pro činnost resolveru. Například `timeout`, počet pokusů pro připojení na DNS server `attempts` a další.

Příklad nastavení resolveru:

```
search netlab.fit.vutbr.cz fit.vutbr.cz
nameserver 127.0.0.1
```

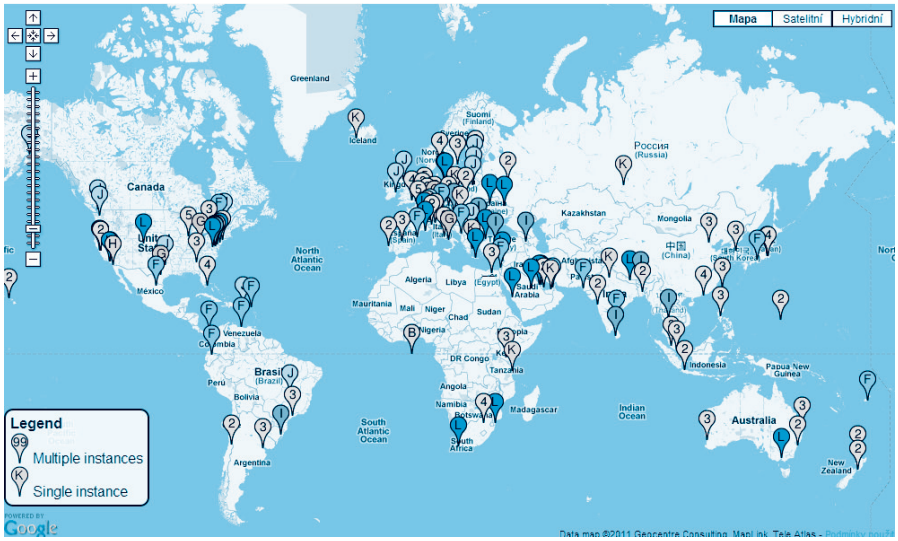
Rezoluce dotazů DNS

Rezoluce (resolution, rozlišení) je proces hledání odpovědi v systému DNS. Protože prostor doménových adres je strukturován jako kořenový strom, stačí každému serveru

DNS pouze jediná informace, jak vyhledat libovolný uzel ve stromu – adresa kořenového serveru DNS. Server DNS se zeptá kořenového serveru DNS na nejvyšší doménu a pak postupuje po stromové struktuře od kořene až k uzlu, který obsahuje hledanou informaci.

Kořenový server DNS (root nameserver) je autoritativní server DNS pro všechny domény nejvyšší úrovně TLD. Ke každé existující doméně TLD zná adresu autoritativního serveru. Při dotazu na jakékoliv doménové jméno odpoví kořenový DNS server tak, že vrátí adresu serveru DNS, který informaci obsahuje. Servery DNS pro domény první úrovně obsahují informace o příslušných subdoménách druhé úrovně atd. Činnost kořenového serveru DNS popisuje dokument RFC 2870 [33].

Kořenový server DNS je nezastupitelný pro správný běh celého systému DNS. Z tohoto důvodu existuje třináct kořenových serverů s adresami `[a–m].root-servers.net.`, které jsou rozmístěné po celém světě u různých operátorů, viz obrázek 3.6. Obvykle se nejedná o jediný počítač. Zátěž každého jednoho kořenového serveru je rozložena na více strojů. Přesto každý z nich odpovídá na tisíce dotazů během každé sekundy.



Obrázek 3.6: Umístění kořenových serverů DNS, převzato z www.root-servers.org.

Otázkou je, zda stačí třináct kořenových serverů v případě výpadku. Během posledních let bylo zaznamenáno několik větších útoků na infrastrukturu DNS. Jedním z nich byl útok DDoS (Distributed Denial of Service) dne 21. 10. 2002, který byl zaměřen na zahlcení všech třinácti kořenových serverů DNS pakety ICMP, TCP SYN, fragmenty TCP a UDP. Pakety přicházely na jednotlivé servery rychlostí cca 100 Mb/s. Útok trval cca 75 minut, během něhož byly některé kořenové servery nedostupné. Neobjevily se však žádné dopady na uživatelské sítě, pouze zpoždění v odpovědích od DNS. Jak uvádí oficiální zpráva o útoku[2], útok prokázal dostatečnou robustnost systému DNS i v případě soustředěného útoku.

Druhým větším útokem na infrastrukturu DNS byl útok DDoS ze dne 6. 2. 2007, který trval dvě a půl hodiny, a poté se opakoval po dalších pět hodin. Při útoku bylo šest ze tři-

2 P. Vixie, et al: Events of 21-Oct-2002, http://c.root-servers.org/october21.txt.

nácti kořenových serverů zasaženo, nicméně ostatní kořenové servery pracovaly efektivně, takže útok neměl téměř žádný dopad na uživatele Internetu[3].

Jak vidíme, jsou kořenové servery DNS nezbytné pro činnost systému DNS. Pokud budeme chtít znát například autoritativní servery pro doménu `cz.`, stačí poslat dotaz na jakýkoliv kořenový DNS server. Vrátí seznam serverů DNS, které jsou autoritativní pro doménu `cz.`

```
/home/matousp> nslookup -type=NS cz. a.root-servers.net
    Server:     a.root-servers.net
    Address:    198.41.0.4#53

    Non-authoritative answer:
    *** Can't find cz.: No answer

cz  nameserver = a.ns.nic.cz.
cz  nameserver = b.ns.nic.cz.
cz  nameserver = c.ns.nic.cz.
cz  nameserver = d.ns.nic.cz.
cz  nameserver = f.ns.nic.cz.
a.ns.nic.cz     has AAAA address 2001:678:f::1
a.ns.nic.cz     internet address = 194.0.12.1
b.ns.nic.cz     has AAAA address 2001:678:10::1
b.ns.nic.cz     internet address = 194.0.13.1
c.ns.nic.cz     has AAAA address 2001:678:11::1
c.ns.nic.cz     internet address = 194.0.14.1
d.ns.nic.cz     has AAAA address 2001:678:1::1
d.ns.nic.cz     internet address = 193.29.206.1
f.ns.nic.cz     has AAAA address 2001:628:453:420::48
f.ns.nic.cz     internet address = 193.171.255.48
```

V ukázce vidíme, že kořenový server `a.root-servers.net` sám nezná doménu .cz, ale odkáže nás odkázat na autoritativní servery pro tuto doménu. Jak vidíme, autoritativní servery DNS pro doménu cz jsou na pěti serverech `[a-f].ns.nic.cz`.

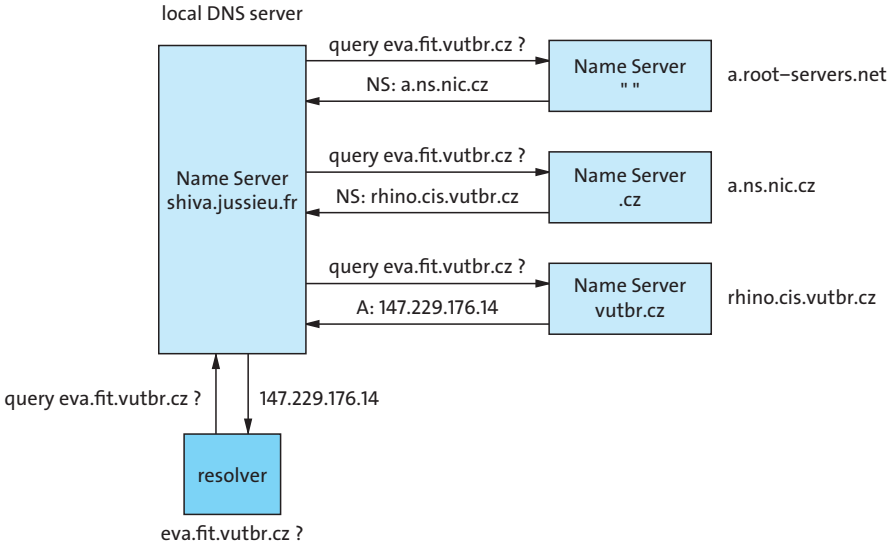
Jak vypadá v praxi zasílání dotazu na systém DNS (rezoluce) si ukážeme na následujícím příkazu, viz též obrázek 3.7. Uživatelský program, který běží na počítači například někde ve Francii, potřebuje přeložit doménovou adresu `eva.fit.vutbr.cz` na IP adresu. Požadavek předá programu resolveru.

Proces vyhodnocování dále probíhá takto:

1. Pokud má resolver vlastní cache paměť, zkusí vyhledat dotaz v ní. Pokud informaci nenajde, přepošle dotaz nejbližšímu (lokálnímu) DNS serveru. V našem případě je to `shiva.jussieu.fr`.
2. DNS server hledá odpověď ve svých záznamech. Pokud ji nenajde (např. jde o dotaz na jinou doménu), musí najít autoritativní server pro danou doménu. Pošle tedy dotaz kořenovému serveru, kde zjišťuje odpověď. Pokud kořenový server sám nezná odpověď, přepošle odkaz na server, který je pro doménu (v našem případě `cz.`) autoritativní.

3 viz zpráva ICANN, http://www.icann.org/en/announcements/factsheet-dns-attack-08mar07.pdf

3. Kořenový server tedy odpoví záznamem NS pro doménu `cz.`
4. Lokální server DNS postupně kontaktuje autoritativní server domény `.cz.` a požádá o překlad doménového jména.
5. Protože ani server `a.ns.nic.cz` nezná doménu `fit.vutbr.cz`, pošle odkaz na server obsahující doménu `vutbr.cz.`
6. Lokální server DNS tedy kontaktuje autoritativní server DNS pro doménu `vutbr.cz.`
7. Server `rhino.cis.vutbr.cz` odpověď zná a vrátí záznam A obsahující IP adresu hledaného počítače `eva.fit.vutbr.cz`.
8. Lokální server DNS přepošle odpověď resolveru. Kopii si uloží do lokální paměti cache a ponechá ji tam, dokud nevyprší platnost záznamu.
9. Resolver předá odpověď uživatelskému programu.



Obrázek 3.7: Příklad rezoluce rekurzivního DNS dotazu

Ve výše uvedeném příkladu vidíme, že lokální server DNS se opakovaně dotazuje různých serverů DNS, dokud nedostane požadovanou odpověď. Ostatní servery pouze najdou nejlepší možnou odpověď (obvykle záznam NS), který předají tázajícímu serveru. Sami se už nesnaží kontaktovat tyto servery DNS a najít autoritativní odpověď. To už je věcí resolveru, aby se opakovaně dotazoval. Pokud by to neudělal resolver, je to na uživateli, aby opakovaně posílal dotazy na jednotlivé servery ve stromu DNS.

Dotaz, který zaslal resolver lokálnímu DNS serveru, byl *rekurzivní*. Servery DNS rozlišují dva typu dotazů – rekurzivní a iterativní.

- **rekurzivní dotaz**
  Rekurzivní dotaz je podobný rekurzi, jak ji známě např. z programování. Resolver zašle dotaz na určitý údaj ve stromu DNS konkrétnímu serveru DNS. Server DNS musí odpovědět na dotaz buď požadovanými daty nebo chybovou hláškou, když například nezná odpověď. Pokud server není autoritativní pro hledaná data, musí se zeptat dalších serverů a najít autoritativní odpověď. Může poslat rekurzivní dotaz na některý z autoritativních serverů a čekat na odpověď. Nebo může poslat iterativní dotaz a zís-

kat odkaz na jiný server, který zná odpověď. Zda server podporuje rekurzivní či iterativní dotazování záleží na jeho konfiguraci.
Server DNS, který obdržel rekurzivní dotaz, přeposílá vždy stejný dotaz na další servery. Nikdy se neptá například na záznamy NS pro hledanou doménu. Většina programů (například `nslookup`, `dig`) zasílá rekurzivní dotazy.

- **iterativní dotaz**
  Iterativní dotazy šetří práci na straně serveru DNS. Při tomto dotazování vrátí server resolveru nejlepší odpověď, kterou může dát. Více se nedotazuje. Dotazovaný server DNS se podívá do své lokální databáze. Pokud nenajde odpověď, vrátí adresy serverů, které jsou nejblíže hledané adrese.

V příkladu, který jsem výše uvedli, poslal resolver rekurzivní dotaz na lokální server DNS. Tento server pomocí iterativních dotazů vyhledával požadované informace. Odpověď pak předal resolveru.

Ukládání záznamů DNS v paměti cache
Abychom redukovali počet dotazů na systém DNS, používá většina serverů DNS lokální vyrovnávací paměť cache, kde si průběžně ukládá odpovědi od serverů pro budoucí vyhledávání. Některé verze serverů DNS umožňují implementovat také *negativní cache*. V této paměti si server ukládá negativní odpovědi od autoritativních serverů, tj. informace o tom, že hledaný záznam v dané doméně neexistuje. Negativní cache také zrychluje proces rezoluce dotazů DNS. Odpovědi získané z cache paměti nemusí být přesné a označují se jako *neautoritativní*, viz následující příklad:

```
/home/matousp> nslookup www.stanford.edu
Server:         147.229.9.43
Address:        147.229.9.43#53

Non-authoritative answer:
www.stanford.edu        canonical name = www-lb.stanford.edu.
Name: www-lb.stanford.edu
Address: 171.64.13.26
```

V tomto příkladu jsme se dotázali lokálního serveru DNS s adresou 147.229.9.43 (uvedeného v síťové konfiguraci počítače) na IP adresu serveru www.stanford.edu. Odpověď, kterou jsme dostali, je neautoritativní, tzn. předává nám ji server, který nespravuje záznamy v zóně `stanford.edu`. Pokud bychom chtěli získat autoritativní odpověď, musíme se dotázat primárního či sekundárního serveru pro tuto zónu. Pak odpověď vypadá takto:

```
/home/matousp> nslookup www.stanford.edu Argus.stanford.edu
Server:         Argus.stanford.edu
Address:        171.64.7.115#53

www.stanford.edu        canonical name = www-lb.stanford.edu.
Name: www-lb.stanford.edu
Address: 171.64.13.26
```

Ukládání lokálních kopií záznamů DNS na neautoritativním serveru DNS přináší kromě zrychlení vyhledávání i jedno nebezpečí – informace nemusí být aktuální. U pozitivních odpovědí se mohl záznam změnit či být zrušen, u negativního ukládání odpovědí se mohlo stát, že chybějící záznam byl přidán. Z tohoto důvodu má každý záznam DNS uvedenu *maximální dobu platnosti záznamu* (TTL, time to live), kterou nastavuje administrátor zóny pro daný zónový soubor.

Hodnota TTL určuje maximální dobu (v sekundách), po kterou je možné data uchovat v lokální paměti cache. To ovšem neznamená, že se data nemohou změnit. Nicméně po uplynutí doby TTL je musí server DNS ze své paměti cache odstranit a načíst si aktuální data z autoritativního serveru. Podobně to platí o zálohování negativních dotazů. I pro negativní dotazy existuje hodnota TTL.

Nastavení hodnoty TTL výrazně ovlivňuje poměr mezi konzistencí databáze na jedné straně a zvýšenou zátěží komunikace na straně druhé. Malé TTL zaručí lepší konzistenci dat, na druhou stranu způsobí větší počet dotazů na vlastní autoritativní servery DNS. Větší TTL může způsobit nekonzistenci dat. RFC 1033 [21] doporučuje jako minimální hodnotu TTL jeden den (86 400 sekund), což je vhodné pro servery, kde nedochází k častým změnám. Při použití DNS k rozložení zátěže (load balancing) je vhodné mít mnohem menší TTL. Pro stabilní servery se doporučuje hodnota tři až pět dní. Tyto hodnoty se nastaví v záznamech typu SOA, o kterých si povíme v další kapitole.

3.3 Záznamy DNS (Resource Records)

Pro ukládání informací v datovém prostoru DNS slouží záznamy DNS (resource records, RR). Záznamy jsou uloženy v textové podobě v zónových souborech na serverech DNS. Nejběžnější jsou záznamy typu A, které mapují doménové jméno na IP adresu (tzv. přímé mapování) a záznamy typu PTR pro opačné (reverzní) mapování. V této kapitole si povíme také o záznamech SOA, NS, MX a CNAME. Standardy RFC 1034 [27], RFC 1035 [28], RFC 1183 [7], RFC 3401 [23], RFC 3597 [3] popisují jednotlivé typy záznamů DNS. Některé typy záznamů se běžně používají (AAAA,SIG, NAPTR), jiné jsou spíše experimentální a nerozšířili se (RP, RT, MB). Tabulka 3.4 uvádí přehled standardů pro základní typy záznamů DNS.

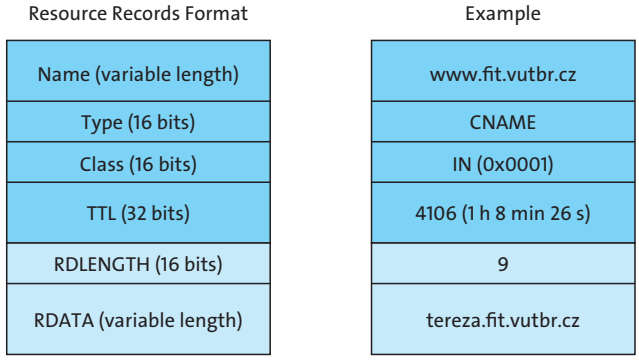
Typ záznamu	Název	Standard
SOA	Start of Authority	RFC 1034
NS	Name Server	RFC 1034
A	Address	RFC 1034
MX	Mail Exchanger	RFC 1034
CNAME	Canonical Name	RFC 1034
PTR	Domain Name Pointer	RFC 1034
NAPTR	Naming Authority Pointer	RFC 2915, 3403, 3761
TXT	Text	RFC 1034
SRV	Service Record	RFC 2782
LOC	Location	RFC 1876
AAAA	IPv6 Address	RFC 3596
DNSKEY	DNS Key Record	RFC 4034
RRSIG	DNSSEC Signature	RFC 4034

NSEC	Next-Secure Record	RFC 4034
NSEC3	NSEC record version	RFC 5155
DS	Delegation Signer	RFC 4034

Tabulka 3.4: Základní typy záznamů DNS a příslušné standardy

3.3.1 Formát záznamů

Všechny typy záznamů DNS mají stejný obecný formát definovaný standardem RFC 1035 [28]. Formát záznamů obsahuje položky `NAME`, `TYPE`, `CLASS`, `TTL`, `RDLENGTH` a `RDATA`. Každý záznam DNS obsahuje všechny uvedené položky. Položka `RDATA` se liší podle typy záznamu, kde pro daný typ (např. A či MX) obsahuje odpovídající informace. Struktura záznamu je i s příkladem na obrázku 3.8.



Obrázek 3.8: Formát DNS záznamu

Položka `Name` obsahuje jméno uzlu ve stromu DNS, kde je daný záznam uložen. Položka `Type` určuje typ záznamu (např. CNAME), `Class` definuje třídu záznamu, `TTL` maximální dobu platnosti záznamu. Pokud je hodnota TTL nastavena na 0, což je běžné například pro záznamy SOA, záznam nesmí být uložen v paměti cache. Pole `RDLENGTH` a `RDATA` obsahují hodnotu záznamu, na kterou se obvykle ptáme.

Na obrázku 3.8 vidíme příklad záznamu typu CNAME, který obsahuje kanonické jméno (CNAME) pro doménovou adresu `www.fit.vutbr.cz`. Server DNS vrátí na dotaz typu CNAME pro uzel `www.fit.vutbr.cz` hodnotu `tereza.fit.vutbr.cz`. Záznam patří do třídy IN (Internet), jeho platnost je 1 hodina, 8 minut a 26 sekund, délka dat 9 bytů a obsahuje hodnotu `tereza.fit.vutbr.cz`, což je kanonické jméno pro hledanou doménovou adresu.

Všímavého čtenáře možná napadne, že řetězec `tereza.fit.vutbr.cz` je delší než 9 bytů. Je to tak. DNS používá pro přenos doménových jmen speciální kompresi paketů DNS, kde opakující se části doménových adres nahrazuje odkazy na první výskyt v záznamu. Například zde vidíme opakující se řetězec `fit.vutbr.cz`, který je možné komprimovat. Více o kompresi paketů je v části 3.4.4.

Kromě třídy IN (Internet), definuje DNS ještě další tři historické třídy – CS (CSNET), CH (CHAOSnet) a HS (Hesiod). CHAOSnet byl protokol pro sítě LAN vyvinutý v polovině 70. let na MIT, který se využíval na pracovních stanicích v oblasti umělé inteligence.

CHAOSnet měl vlastní adresování s délkou 16 bitů zapisované v osmičkové soustavě, například 315.124. Dnes se už nepoužívá.

Podobně třída Hesiod sloužila v 80.–90. letech pro ukládání textových informací o uživatelích, skupinách uživatelů, tiskárnách apod. Například pro uživatele s uživatelským jménem `jim` a skupinou `candlewood` (jim.candlewood.hesiod) vrátila UID a GID, například 501.142. Dnes se pro ukládání informací nepoužívá DNS, ale spíše adresářové služby typu LDAP.

Různé typy záznamů DNS se liší v počtu, významu a délce polí, které obsahují v části `RDATA`. Výše uvedený příklad obsahuje pouze jedno pole s kanonickým jménem. Záznamy typu A obsahují jedno pole s 32-bitovou IP adresou, záznamy typu MX dvě pole s 16-bitovou hodnotou preference a doménovým jménem poštovního serveru. Přesný popis polí u jednotlivých typů záznamů je uveden v dříve zmíněných standardech.

Formát záznamů DNS v zónovém souboru

Výše uvedený formát záznamů DNS je určen pro binární kódování a používá se převážně při komunikaci mezi servery a klienty DNS. Záznamy DNS se v zónových souborech na serverech obvykle zapisují v textové podobě. Obsah odpovídá binární podobě záznamů DNS, formát je odlišný. Pro nás je textové vyjádření důležitější než binární podoba určená pro přenos, protože správce serveru DNS pracuje výhradně s textovými daty. Formát zónového souboru a uložení záznamů v nich je opět definováno standardem RFC 1035 [28]. Vypadá takto:

```
<blank                                  [<comment>]
$ORIGIN <domain-name>                   [<comment>]
$INCLUDE <file-name> [<domain-name>]    [<comment>]
<domain-name> <rr>                      [<comment>]
<blank> <rr>                            [<comment>]
```

Na začátku zónového souboru jsou definovány dvě řídící položky `\$ORIGIN` a `\$INCLUDE`, které nepatří mezi záznamy. Položka `\$ORIGIN` slouží ke změně původní adresy uzlu v DNS na jinou hodnotu. Například při nastavení `\$ORIGIN fit.vutbr.cz.` na začátku doménového souboru můžeme používat relativní doménová jména (např. `www`, `email`, `kazi`), která se v následujících záznamech zónového souboru interpretují jako absolutní adresy `www.fit.vutbr.cz`, `email.fit.vutbr.cz` či `kazi.fit.vutbr.cz`. Položka `\$INCLUDE` vkládá do zónového souboru další soubor např. s definicí dalších záznamů DNS.

Komentáře začínají znakem „;“ (středník). Znak „@“ (zavináč) označuje substituci doménového jména v uzlu (`origin`). Pokud záznam DNS `<rr>` začíná mezerou (`blank`), patří daný záznam k doméně, která byla uvedena u předchozího záznamu. Zónový soubor může obsahovat více záznamů DNS `<rr>`. Každý záznam začíná na novém řádku. Formát DNS záznamu `<rr>` je následující:

```
[<TTL>] [<class>] <type> <RDATA>
```

Nepovinné hodnoty polí `TTL` a `class` mohou být vynechány. Pokud chybí, použijí se hodnoty definované v předchozím záznamu, případně se např. pro TTL použije implicitní hodnota daná direktivou `\$TTL` na začátku zónového souboru. Příklad zónového souboru vypadá takto:

```
$TTL 3600
@ IN SOA isa.fit.vutbr.cz. root.isa.fit.vutbr.cz.(
              200711090      ; Serial
              3600   ; Refresh
              900    ; Retry
              3600000        ; Expire
              3600 ) ; Minimum
     IN NS    isa.fit.vutbr.cz.
     IN MX 0  eva.fit.vutbr.cz.
isa  IN A     10.10.10.1
h01  IN A     10.10.10.101
h02  IN A     10.10.10.102
```

Výše uvedený zónový soubor obsahuje záznamy pro doménu `netlab.fit.vutbr.cz.`, což je definováno v konfiguračním souboru serveru DNS. Zónový soubor obsahuje záznamy pouze této domény. Obsahuje jeden záznam typu SOA, jeden záznam typu NS, jeden záznam typu MX a tři záznamy typu A.

Na začátku zónového souboru je direktivou `\$TTL` definována maximální platnost záznamu, která je jednu hodinu (3600 sekund). Znak „@“ říká, že doménové jméno uzlu, ve kterém je uložen tento záznam, je shodný s názvem domény uvedeným v konfiguračním souboru serveru DNS `named.conf`. Pokud bychom chtěli vložit záznam jiné domény, potřebujeme napsat celé doménové jména. Místo znaku `@` bychom napsali doménu `netlab.fit.vutbr.cz.` Server DNS `named` totiž při restartu načítá konfigurační soubor `named.conf` a zónové soubory, které se z něho odkazují. Při načítání záznamů se zónových souborů provede substituci znaku „@“ za název uvedený v konfiguračním souboru, viz obrázek 3.5.

Dále výše uvedený zónový soubor obsahuje záznam typu SOA, který obsahuje primární soubor domény (tj. jméno primárního DNS serveru), emailová adresu správce (zde se nesmí vyskytovat substituční znak „@“, nahrazuje se znakem „.“) a položky s hodnotami. Pokud přesahují hodnoty záznamu jeden řádek, je potřeba uzavřít soubor hodnot do kulatých závorek. Po záznamu SOA následuje informace o doméně (záznam typu NS) a záznamy typu A s překlady doménových adresa na IP adresy. Podrobnější popis záznamů je v následující kapitole.

3.3.2 Popis nejběžnějších záznamů DNS

Záznam SOA – Start Of Authority

Záznam SOA obsahuje informace týkající se uložení autoritativních dat pro danou zónu. Většinou je to první záznam v zónovém souboru. Každá zóna má právě jeden záznam SOA.

```
$TTL 3600
@ IN SOA isa.fit.vutbr.cz. root.isa.fit.vutbr.cz.(
              2007110901     ; Serial
              3600     ; Refresh
              900      ; Retry
              3600000  ; Expire
              3600 )   ; Minimum
```

Záznam SOA obsahuje jméno primárního serveru DNS pro danou doménu (isa.fit.vutbr.cz). Dále obsahuje kontakt na správce domény – jeho emailovou adresu (root@isa.fit.vutbr.cz). Sériové číslo `serial` slouží k identifikaci změn záznamu. Doporučený formát je `YYYYMMDDnn`, kde `YYYYMMDD` označuje rok, měsíc a den, kdy proběhla změna, a hodnota `nn` označuje pořadí změny, například když během jednoho dne došlo k několika změnám.

Důležité je, že při každé změně záznamu se musí sériové číslo změnit. Sériové číslo záznamu SOA totiž používají sekundární servery DNS ke zjišťování změn. Pokud aktualizujeme zónový soubor pro danou doménu a nezměníme sériové číslo záznamu SOA, sekundární DNS server se při aktualizaci podívá na sériové číslo a když zjistí, že se nezměnilo, předpokládá, že zónový soubor je beze změny a už si nenačte nový obsah zónového souboru.

Hodnota `Refresh` udává čas (v sekundách), jak dlouho má sekundární server čekat, než bude zjišťovat změnu. `Retry` určuje, kdy se sekundární server opět pokusí o přenos zóny v případě neúspěšného připojení a aktualizace dat. Hodnota `Expire` definuje maximální dobu platnosti dat na sekundárním serveru. Pokud server není schopen po uplynutí této doby data aktualizovat, musí je smazat. Hodnota `Minimum` udává implicitní hodnotu TTL záznamů v zónovém souboru. Tyto hodnoty mohou být upřesněny u jednotlivých záznamů v zónovém souboru.

Doporučené hodnoty záznamů pro servery DNS nejvyšší úrovně (TLD) a pro ostatní servery DNS uvádí podle RFC 1537 [26], viz tabulka 3.5.

Položka	Hodnota pro servery TLD	Hodnota pro ostatní servery DNS
Refresh	24 hodin	8 hodin
Retry	2 hodiny	2 hodiny
Expire	30 dní	7 dní
Minimum TTL	4 dny	1 den

Tabulka 3.5: Doporučené hodnoty SOA podle RFC 1537

Záznam NS – Name Server

Záznam typu NS určuje autoritativní server (či servery) pro danou doménu. Pomocí těchto záznamů dochází k budování hierarchické struktury systému DNS. Záznam NS, který je umístěn v daném uzlu stromu DNS, obsahuje ukazatele na níže připojené následovníky uzlu, viz obr. 3.1.

```
fit.vutbr.cz. IN NS gate.feec.vutbr.cz.
IN NS guta.fit.vutbr.cz.
IN NS kazi.fit.vutbr.cz.
IN NS rhino.cis.vutbr.cz.
```

Tento příklad uvádí autoritativní servery, které obsahují platné záznamy DNS o doméně `fit.vutbr.cz`. Pokud bychom chtěli zjistit, který ze serverů je primární a které jsou sekundární, musíme se dotázat na záznam SOA a z něho tu informaci vyčíst.

Záznam A – IP Address

Záznam typu A obsahuje přímé mapování doménových adres na IP adresy. Je to jediný záznam, který obsahuje na pravé straně IP adresu. Pokud chceme získat IP adresu napří-

klad DNS serveru (záznam NS) nebo poštovního serveru (záznam MX), musíme nejdříve získat kanonické jméno obou serverů a teprve poté požádat o A záznam každého z nich.

```
eva.fit.vutbr.cz. IN A 147.229.176.14.
```

Záznam MX – Mail Exchanger
Záznam typu MX nás informuje o poštovním serveru, který pro danou doménu přijímá poštu. Emailová adresa v sobě obsahuje jméno adresáta i místo, kde je jeho poštovní schránka (at, znak „@“). Například emailová adresa `matousp@fit.vutbr.cz`, říká, že elektronická pošta pro uživatele `matousp` se doručuje do emailové schránky uložené v doméně `fit.vutbr.cz`. Mapování domény na adresy poštovního serveru obsahuje záznam MX.

```
fit.vutbr.cz. IN MX 10 kazi.fit.vutbr.cz.       # vyšší priorita
              IN MX 20 eva.fit.vutbr.cz
```

Oproti ostatním záznamům obsahuje záznam MX navíc prioritu, která určuje preferenci poštovního serveru. Pokud máme pro danou doménu uvedeno více poštovních serverů, použije se ten s nižší hodnotou. Nižší hodnota označuje vyšší prioritu. V případě, že je prioritní poštovní server nedostupný, použije se pro doručení záložní server podle dalšího záznamu MX. Vlastní hodnota (10 nebo 20) není v záznamech důležitá. Rozhoduje, která hodnota je vyšší a která nižší. Pokud pro danou doménu neexistuje záznam MX, nelze poštu řádně doručit.

Záznam CNAME – Canonical Name
V DNS můžeme pro jeden počítač vytvořit více doménových jmen. Pokud například na počítači `tereza.fit.vutbr.cz` poběží služba WWW, můžeme mu přiřadit snadno zapamatovatelný alias `www.fit.vutbr.cz`. Pokud tam poběží i služba LDAP, přidáme další název `ldap.fit.vutbr.cz`. Záznamy CNAME pro výše uvedený příklad mohou vypadat takto:

```
www     IN CNAME tereza.fit.vutbr.cz.
ldap    IN CNAME tereza.fit.vutbr.cz.

tereza  IN A 147.229.9.22
```

Při každém dotazu do DNS na počítač `ldap.fit.vutbr.cz` vrátí systém DNS kanonické jméno počítače (tj. tereza) a k němu i odpovídající IP adresu uloženou v záznamu typu A. Pomocí aliasů jsme schopni odlišit doménové jméno od IP adresy a v případě potřeby (například přetížení serveru) změnit alias tak, že se odkazuje na záložní server.

Záznam CNAME tedy přiřazuje k aliasu kanonické (oficiální) jméno počítače. Pokaždé, když DNS server při hledání najde záznam CNAME, nahradí doménové jméno kanonickým jménem a pokračuje v hledání.

Alias definovaný záznamem CNAME nesmí nikdy stát na pravé straně záznamu. Všechny ostatní záznamy se musí mapovat na oficiální (kanonické) jméno, nikoliv na alias.

Záznam PTR – Domain Name Pointer
Záznam typu PTR provádí zpětné (reverzní) mapování. To znamená, že převádí číselnou IP adresu na doménové jméno, viz obr. 3.3. Jak jsme uvedli v kapitole 3.2.1, reverzní adresy jsou uloženy ve speciální doméně `in-addr.arpa.` v systému DNS.

```
14.176.229.147.in-addr.arpa. IN PTR eva.fit.vutbr.cz.
```

Protože se jedná o jinou doménu, jsou reverzní záznamy uloženy v jiném zónovém souboru než přímé záznamy. Proto většina serverů DNS ukládá ve své konfiguraci jednak zónové soubory pro přímý překlad (například `cz.vutbr.fit`) a jednak zónové soubory pro zpětný překlad (například `147.229`). Jak vidíme, jsou zónové soubory pro přehlednost pojmenovány názvem domény, která je v zónovém souboru uložena. To znamená, že zónový soubor `cz.vutbr.fit` obsahuje záznamy z domény `fit.vutbr.cz`, případně subdomén dané domény. Zónový soubor `147.229` zase obsahuje reverzní záznamy, které obsahují IP adresy z prostoru 147.229.0.0/16. Zónový soubor pro reverzní záznamy musí kromě reverzních záznamů také obsahovat záznam SOA, případně záznamy NS. Vlastní záznamy PTR musí ukazovat pouze na jedno doménové jméno – kanonické jména. Reverzní záznamy nesmí obsahovat ukazatel na aliasy.

Pokud při vytváření záznamů v DNS zapomeneme pro danou doménovou adresu vytvořit také zpětný záznam, obvykle způsobí chybějící záznamy PTR problémy při autentizaci. Týká se to například služeb SMTP, FTP, rsh, ssh a dalších. Důvodem je, že tyto služby často kontrolují domény, ze kterých se uživatelé připojují. Pokud se počítač zkouší připojit z nějaké IP adresy, server nejprve přeloží IP adresu na doménové jméno dotazem na záznam PTR. Pak např. zkontroluje, zda daná doména se nachází v seznamu nebezpečných domén. Pokud chybí záznam PTR, překlad neproběhne a spojení je odmítnuto.

Záznam TXT – Text
Záznam TXT uchovává v DNS textová data týkající se dodatečných informací o doméně, serveru, správci apod. daného uzlu ve stromu DNS, viz následující příklad. Dnes se záznamy TXT spíše využívají pro ověření vlastnictví domény pro Gooble Apps či prevenci zneužití emailu (obsahují identifikaci emailových serverů, které mohou z dané domény odesílat emaily).

```
fi.muni.cz. 1773   IN TXT  "Masaryk University Brno"
fi.muni.cz. 1773   IN TXT  "Botanicka 68a, 602 00, Brno, Czech republic"
fi.muni.cz. 1773   IN TXT  "Faculty of Informatics"

vutbr.cz.  86066 IN TXT  "v=spf1 mx ip4:147.229.0.0/16 ip6:2001:67c:1220::/48 -all"
vutbr.cz.  86066 IN TXT  "google-site-verification:ejP3Eb5B3rd47i28yuXfa4kqXkck"
```

Záznam SRV – Service Record
Záznam SRV [2] slouží pro lokalizaci služeb a serverů. Může se také použít pro distribuce zátěže či zálohování služeb, podobně jako je tomu u záznamu MX. Zápis používá speciální notaci pro zápis služby a protokolu, nad kterým služba pracuje. Obecný formát záznamu je

```
_Service._Proto.Name TTL Class SRV Priority Weight Port Target
```

Název služby a protokolu se odděluje od názvu doménového jména podtržítkem, viz následující příklad:

```
_http._tcp.www.hello.edu. IN SRV 0 2 80   www.hello.edu.     # rozdělení zátěže
                          IN SRV 0 1 80   www2.hello.edu.
                          IN SRV 1 1 8000 backup.hello.edu.  # priorita 1 (záloha)
_gopher._tcp.hello.edu    IN SRV 0 0 0 .                     # služba neběží
```

Kromě názvu a adresy obsahuje záznam SRV také prioritu, váhu, port a cílový server, kde běží žádaná služba. Priorita se chová podobně jako u MX záznamu. Váha umožňuje administrátorovi rozložit zátěž na více strojů. Pokud má například první záznam váhu 1 a druhý váhu 2, rozdělí se zátěž v poměru 1:2, tj. druhý stroj dostane dvakrát více požadavků než ten první. Hodnota `Port` určuje číslo portu, na kterém daná služba běží. Pokud chceme oznámit, že daná služba neběží, jako cíl zadáme tečku.

V dnešní době se záznam SRV používá například u IP telefonie, kdy klient zjišťuje adresu SIP serveru a číslo portu, na kterém služba běží. Následujícím dotazem IP telefon v doméně vutbr.cz lokalizovat nejbližší SIP server a port této služby. Využije se k tomu dotaz na službu SIP nad protokolem UDP, viz ukázka:

```
/home/matousp> nslookup -type=SRV _sip._udp.vutbr.cz
Server: 147.229.9.43
Address:147.229.9.43#53

_sip._udp.vutbr.cz      service = 0 1 5060 vampire.net.vutbr.cz.
```

Záznam NAPTR – Naming Authority Pointer

Záznam typu NAPTR definovaný v RFC 3403 [24] byl navržen pro mapování řetězců na data. Záznam slouží jako podpora pro dynamicky konfigurovatelné systémy DDDS (Dynamic Delegation Discovery Systems). Při vyhledávání dat se nad řetězci iterativně používají transformační pravidla dokud není dosaženo ukončující podmínky (podobně jako přepisování nonterminálu v gramatikách).

Záznam NAPTR obsahuje regulární řetězec, který klientský program využije k přepsání řetězce v doménové adrese. Formát záznamu NAPTR obsahuje více hodnot, viz následující ukázka:

```
NAPTR Order Preference Flags Services Regexp Replacement
```

První hodnota `Order` určuje, v jakém pořadí se záznamy NAPTR vyhodnocují. Seznam záznamů NAPTR je řazený a zpracovává se od nejmenší hodnoty `Order` k nejvyšší. Záznamy se stejnou hodnotu patří ke stejnému pravidlu a výběr záleží na kombinaci hodnot `Preference` a `Services`.

Hodnota `Preference` určuje, v jakém pořadí budou zpracovány záznamy NAPTR se stejným pořadím `Order`. Záznam s nižší hodnotou má vyšší prioritu (podobně jako u záznamů MX). Příznaky `Flags` řídí proces přepisování a interpretaci záznamů. Jejich hodnota závisí na aplikaci. Hodnota `Services` popisuje parametry služby. Položka `Regexp` obsahuje regulární výraz, který se aplikuje na vstup. Hodnota `Replacement`

uvádí kanonické doménové jméno, které se doplní, pokud je součástí regulárního výrazu operace nahrazení. V praxi se parametr `Replacement` příliš nepoužívá.

Následující záznam demonstruje použití záznamu NAPTR pro převod telefonního čísla zapsaného v telekomunikačním standardu E.164 na URI protokolu SIP (Session Initiation Protocol, viz [16]). Tento převod se používá v IP telefonii, kdy uživatel místo adresy SIP vytočí na IP telefonu telefonní číslo. Pomocí záznamu NAPTR dojde k mapování telefonního čísla ve formátu E.164 na URI. Tento systém mapování se nazývá ENUM (E.164 Number Mapping) [4, 14]. Záznamy NAPTR podobně jako záznamy PTR využívají speciální větev `e164.arpa.` stromu DNS. Následující příklad ukazuje použití záznamu NAPTR pro překlad ENUM:

```
2.7.0.0.5.9.0.2.4.e164.arpa.     3201 IN NAPTR 100 50 "u"
                                 "E2U+sip" "!^\\+(.*)$!sip:\\1@cesnet.cz!" .
```

Tento záznam slouží k převodu telefonní čísla CESNET s předčíslím +420 950 072 podle standardu E.164 na adresu SIP. Tato telefonní čísla jsou registrována v doméně `2.7.0.0.5.9.0.2.4.e164.arpa`. Pro překlad všech telefonních čísel s daným předčíslím nám stačí jediný záznam v DNS. Je to díky tomu, že se využívá regulární výraz. Díky tomu nemusíme pro každou klapku přidávat záznam v daném reverzním podstromu.

Výše uvedený záznam NAPTR obsahuje příznak „u“, který říká, že pravidlo je koncové (tj. dále se nepřekládá) a jeho výsledkem je URI. Znak „!“ v regulárním výrazu odděluje část, která se použije pro porovnání od části, která je výsledkem substituce. Tento regulární výraz přeskočí v telefonním čísle ve formát E.164 znak „+“. Všechny ostatní znaky až do konce vstupního řetězce (vyjádřené zápisem „(.*)“) se vloží na místo ve výsledném řetězci odpovídající parametru „$\backslash 1$“. Parametr tohoto typu se nazývá zpětná reference a odkazuje se na první závorku substituovaného výrazu. S tímto zápisem se můžeme setkat např. v programovacím jazyku Perl. Význam použití regulárních výrazů je podrobně popsán například v RFC 2915 [25].

V našem případě se například číslo +420950072020 zapsané ve stromě DNS jako `0.2.0.2.7.0.0 .5.9.0.2.4.e164.arpa` převede na adresu `sip:420950072020@cesnet.cz`. Aplikaci regulárního výrazu neprovádí server DNS, ale klient SIP, který žádá o záznam NAPTR. To ukazuje i následující dotaz na záznam NAPTR:

```
nslookup -type=naptr 0.2.0.2.7.0.0.5.9.0.2.4.e164.arpa

0.2.0.2.7.0.0.5.9.0.2.4.e164.arpa naptr = 200 50 "u" "E2U+h323"
                        "!^\\+(.*)$!h323:\\1@gk1ext.cesnet.cz!" .
0.2.0.2.7.0.0.5.9.0.2.4.e164.arpa naptr = 100 50 "u" "E2U+sip"
                        "!^\\+(.*)$!sip:\\1@cesnet.cz!" .
```

Výsledkem dotazu je celý záznam NAPTR obsahující regulární výraz, který klient sám aplikuje na telefonní číslo. Za zmínku stojí i tečka na konci výrazu. Tato tečka označuje prázdné pole typu `Replacement`. Použití záznamu NAPTR pro IP telefonii se budeme věnovat v kapitole [7].

Záznam LOC – Location

Záznam typu LOC definovaný standardem RFC 1876 [35] umožňuje administrátorovi zapsat do DNS údaje o umístění počítače, sítě či dalších zdrojů. Popis lokace zahrnuje

informace o zeměpisné šířce (latitude), délce (longitude) a výšce (altitude). Zeměpisná šířka a délka mohou být vyjádřeny ve stupních (minutách a sekundách), výška se udává v metrech. Následující ukázka ukazuje použití záznamu LOC pro serveru `kazi.fit.vutbr.cz`:

```
kazi.fit.vutbr.cz. 14400 IN LOC 49 13 35.000 N 16 35 45.000 E 195.00m 1m 10000m 10m
```

Výše uvedený příklad ukazuje záznam LOC pro server `kazi.fit.vutbr.cz`, jehož umístění je zadaná jako souřadnice 49°13'35.000" severní šířky a 16°35'45.000" východní délky, 195 metrů nad hladinou moře. Dále záznam obsahuje nepovinné údaje o velikosti entity, tj. délka 1 m (což pro server asi stačí) a přesnosti polohy, tj. 10 km metrů horizontálně a 10 metrů vertikálně.

Otázkou je, k čemu tyto informace slouží. Na jednu stranu mohou udávat fyzickou cestu paketů, napojení na GPS apod. Na druhou stranu je tu bezpečnostní riziko a také fakt, že poloha serverů se mění. V současné době najdeme záznamy typu LOC v DNS spíše výjimečně.

AAAA – IPv6 Address

Posledním záznamem, o kterém se zde podrobněji zmíníme, je záznam typu AAAA. Tento záznam definovaný standardem RFC 3596 [35] mapuje doménové adresy na IP adresy verze 6 (IPv6). Zápis je vcelku jednoduchý, podobně jako u záznamu A. Následující ukázka obsahuje příklady dvou záznamů typu AAAA a také ukázku reverzního záznamu pro IPv6 adresu serveru `www.cesnet.cz`.

```
isa.fit.vutbr.cz.          14400 IN AAAA 2001:67c:1220:8b0::93e5:b012
www.cesnet.cz.             85443 IN AAAA 2001:718:1:101::4

4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.1.0.0.0.8.1.7.0.1.0.0.2.ip6.arpa.
                           86400 IN PTR www.cesnet.cz.
```

Složitější je vytváření reverzních záznamů pro adresy IPv6. Tyto adresy patří do speciální domény `ip6.arpa.`, která obsahuje reverzní záznamy adresového prostoru IPv6. Protože je každá subdoména stromu `ip6.arpa.` čtyřbitová, tvoří reverzní doménovou adresu IPv6 celkem 32 šestnáctkových číslic oddělených tečkou. Zároveň se zde nemůže použít notace zkracování adres IPv6 pomocí znaků „:“ a „::“. Pro to se pro generování reverzních adres IPv6 používají automatizované nástroje (např. `sipcalc`), které pro danou adresu IPv6 vygenerují její reverzní záznam.

Pro sítě s IPv6 a IPv4 může mít jedno síťové rozhraní adresu IPv4 i IPv6. V tom případě jsou pak v DNS oba záznamy – A i AAAA.

3.3.3 Přehled záznamů DNS

Jak jsme si ukázali v předchozích odstavcích, datový prostor doménových adres obsahuje nejen překlady doménových adres na IP adresu (záznamy typu A), ale také spoustu dalších informací týkající se domén, služeb, regulárních výrazů a podobě. Každý překlad DNS lze jednoduše popsat jako mapování (funkci), která převádí jeden typ informace na jiný. Pro správné využití systému DNS je nezbytné správně rozlišit jednotlivé typy

mapování. V tabulce \ref{preklad} uvádíme přehled základní typů typů DNS a jejich použití.

Záznam	Mapování	Příklad
A	doménové jméno → IP adresa	tereza.fit.vutbr.cz → 147.229.9.22
PTR	IP adresa → doménové jméno	22.9.229.147.in-addr.arpa. → tereza.fit.vutbr.cz.
NS	doména → doménový server	fit.vutbr.cz. → gate.feec.vutbr.cz.
MX	doména → poštovní server	fit.vutbr.cz. → kazi.fit.vutbr.cz.
SOA	doména → identifikace správce	fit.vutbr.cz. → boco.fee.vutbr.cz. michal.fit.vutbr.cz. 200710032 10800 3600 604800 86400
CNAME	doménové jméno → doménové jméno	www.fit.vutbr.cz. → tereza.fit.vutbr.cz.
AAAA	doménové jméno → IPv6 adresa	www.cesnet.cz. → 2001:718:1:101:204:23ff:fe52:221a
PTR	IPv6 adresa → doménové jméno	a.1.2.2.2.5.e.f.f.f.3.2.4.0.2.0.1.0.1.0.1.0.0.0.8.1.7.0.1.0.0.2 .ip6.arpa. → www.cesnet.cz.

Tabulka 3.6: Přehled základních typů překladu DNS

Z tabulky je například vidět, že pouze záznamy typu A a AAAA překládají doménovou adresu na IP adresu (verze 4 či verze 6). Pokud definujeme poštovní server pomocí záznamu MX, resp. doménový server pomocí záznamu NS, překládá se doména na doménovou adresu poštovního serveru, resp. DNS serveru, nikoliv na jejich IP adresu. Překlad na IP adresu zajistí až vyhledání záznamu typu A či AAAA.

3.3.4 Rozdělování zátěže pomocí rotace záznamů

Novější verze serverů DNS umožňují rozdělit zátěž na více serverů pomocí tzv. záznamů s posunutými adresami (shuffle address records). Znamená to, že na jeden dotaz odpovídá DNS server různými IP adresami, které pravidelně rotuje. Nejedná se o *vyvažování zátěže* (load balancing), které řeší například záznamy SRV. Zde se dotazy rozdělují deterministicky a rovnoměrně, nikoliv podle skutečného vytížení serverů. Mluvíme tedy o *rozdělování zátěže* (load distribution). U některých serverů DNS (např. BIND 9) je možné pomocí volby `rrset-order` v konfiguračním souboru nastavit způsob rotace záznamů, například fixní, cyklický či náhodný.

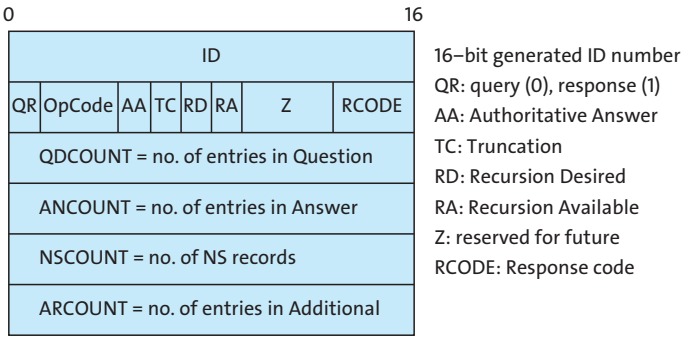
Pro implementaci rotace záznamů se využívá více záznamů typu A, které ukazují na stejné jméno počítače, ale mají různou IP adresu. Pokud je v zóně více záznamů typu A pro doménové jméno, server automaticky přiděluje různé IP adresy (rovnoměrně rotuje záznamy, tzv. round-robin load distribution), viz následující příklad.

```
foo.bar.baz. 60 IN A 192.168.1.1
foo.bar.baz. 60 IN A 192.168.1.2
foo.bar.baz. 60 IN A 192.168.1.3
```

Tento přístup je vhodný, pokud máme více ekvivalentních síťových zdrojů, například servery FTP či WWW se stejným obsahem (mirroring). U záznamů je vhodné nastavit menší hodnotu TTL, aby záložní servery, které nepodporují vícenásobné záznamy, tyto záznamy po krátké době vymazaly z paměti cache.

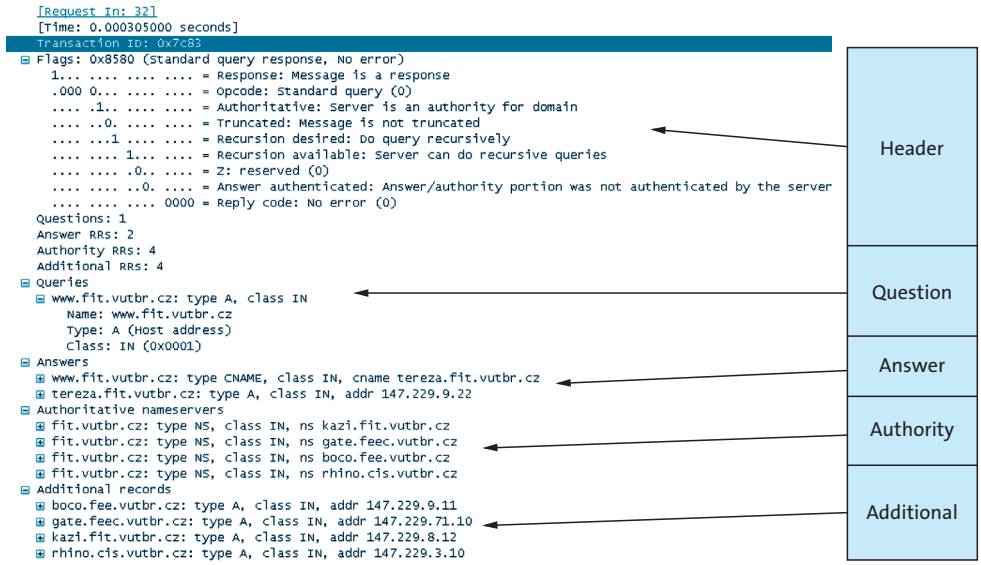
3.4 Přenos dat a komunikace v DNS

Protokol DNS a veškerá komunikace v systému DNS je popsána standardem RFC 1035 [28]. Protokol DNS používá pro posílání dotazů transportní protokol UDP s číslem portu 53. Velikost paketů UDP je standardem DNS omezena na 512 bytů. Delší zprávy je nutné rozdělit do více paketů pomocí bitu TC (TrunCation) v hlavičce protokolu DNS (obr. 3.9). V tomto případě může klient požádat o přenos pomocí TCP.



Obrázek 3.9: Hlavička paketu DNS podle RFC 1035

Při použití UDP jde o nespolehlivou komunikaci. Proto musí aplikace – v případě ztráty paketu – poslat dotaz na systém DNS opakovaně. Pakety UDP také mohou přijít v jiném pořadí, než klient žádal. Pro identifikaci odpovědi slouží 16-bitové číslo v hlavičce paketu UDP. Pro přenos zón se nepoužívá transportní protokol UDP ale TCP.



Obrázek 3.10: Formát paketu DNS

Paket DNS tvoří pět základních položek (obrázek č. 3.10): hlavičku (header), části obsahující dotaz (question), část s odpovědí (answer), záznam obsahující informace o záznamech NS (authority) a sekce obsahující doplňující informace (additional).

Hlavička je povinná položka paketu DNS. Tvoří ji pole pevné délky, které obsahuje informace o zbývajících částech paketu – počet záznamů v dalších sekcích paketu, viz obrázek 3.9, typ paketu (dotaz/odpověď), příznaky pro rekurzivní vyhledávání (RD, RA), rozdělení odpovědi do více paketů UDP (TC) či typ odpovědi (RCODE).

3.4.1 Aktualizace zónových dat

Vytvoření, přidávání, změnu či rušení dat v zónovém souboru můžeme provádět ručně editací textového souboru nebo pomocí automatizovaných nástrojů. Důležité je si uvědomit, že tyto změny lze provádět pouze na primárním serveru DNS. Pokud provedeme změnu například na sekundárním serveru, přepíše se tato změna při nejbližší synchronizaci. Aktualizace zónového souboru tvoří následující kroky:

1. Aktualizace sériového čísla v SOA záznamů příslušného zónového souboru.

```
@ IN SOA isa.fit.vutbr.cz. root.isa.fit.vutbr.cz.(
              2008073002   ; Serial
              3600         ; Refresh
              900          ; Retry
              3600000      ; Expire
              3600 )       ; Minimum
```

2. Přidání, změna či odstranění záznamů typu A, CNAME, MX, apod. v přímém zónovém souboru, např. `cz.vutbr.fit.netlab`.

```
pc12 IN A 10.1.1.12
www IN CNAME pc12
```

3. Restart primárního serveru DNS, při kterém si aplikace načte aktualizovaná data do paměti.

Pokud existuje reverzní zóna (např. `10.1.1`), musíme příslušné změny provést i v záznamu PTR a záznamu SOA této reverzní zóny. Například dojde k přidání reverzním záznamu:

```
12   IN PTR pc12.fit.vutbr.cz.
```

3.4.2 Dynamické změny v DNS

Prostředí Internetu lze považovat za dynamické prostředí, kde se neustále mění topologie, uživatelé se připojují a odpojují od sítě. Toto chování má vliv na změny v databázi DNS. Jde například o uživatele, kteří získávají IP adresy přes DHCP pomocí

vytáčeného spojení či kabelového modemu. Také tyto dynamicky přidělené IP adresy je potřeba mapovat na doménové adresy. Z pohledu systému DNS to vyžaduje přidání nového záznamu typu A. Protože není možné tyto změny provádět ručně, vznikl v roce 1997 standard RFC 2136 [31], který popisuje *dynamické aktualizace v DNS* a příkaz `UPDATE`.

Dynamické aktualizace umožňují přidávat, měnit a rušit záznamů DNS v rámci dané domény. Většina dynamických aktualizací je využívána servery DHCP, které přidělí automaticky počítači IP adresu a zároveň zaregistrují převod IP adresy na doménovou adresu v DNS. Využívá se přitom funkce resolveru `ns\_update()`. Existuje také program `nsupdate`, který spustí změny z příkazové řádky. Následující příklad přidá do databáze DNS nový záznam typu A v doméně `netlab.fit.vutbr.cz` uvedené v příkazu `prereq`.

```
% nsupdate
> prereq nxdomain netlab.fit.vutbr.cz
> update add pc3.netlab.fit.vutbr.cz. A 10.10.10.134
>
```

Jakmile server provede dynamickou aktualizaci, musí inkrementovat sériové číslo zónového souboru, které oznamuje změnu sekundárním serverům. Toto se také děje automaticky. Primární server však nemusí z hlediska efektivity měnit sériové číslo záznamu SOA při každé dynamické změně. Některé servery (např. BIND 8) provedou změnu sériového čísla až po stu aktualizací nebo po pěti minutách (podle toho, co nastane dříve). Souvisí to se zónovým přenosem, o kterém se budeme bavit v další kapitole. Po každé změně sériového čísla může primární server poslat zprávu DNS NOTIFY a vyvolat zónový přenos mezi primárním serverem a sekundárními servery, což s sebou přináší zvýšený přenos a zpracování aktualizací.

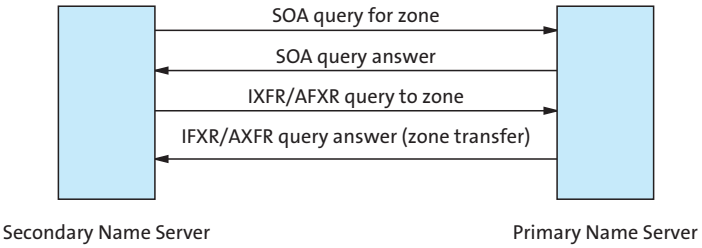
Důležitou částí dynamické aktualizace je také autentizace resolveru, který aktualizaci provádí. Zabezpečení dynamické aktualizace popisuje podrobněji RFC 3007 [5] a využívá mechanismy TSIG a DNSSEC (viz též kapitola 3.5).

3.4.3 Zónový přenos

Většina komunikace v systému DNS probíhá mezi resolvery a servery. Resolver pošle serveru DNS jednoduchý dotaz na získání informace ze jmenného prostoru DNS. Server informaci vyhledá a pošle požadovaná data resolveru. Pokud informaci nezná, může se dotázat dalších serverů.

Jiným způsobem komunikace v systému DNS je *zónový přenos*, kdy dochází k přenosu zónových souborů mezi primárním serverem (master) a sekundárním serverem (slave). Tradičně používají sekundární servery DNS schéma vyzývání (polling). Interval vyzývání odpovídá intervalu aktualizací (refresh interval) uvedený v záznamu SOA. Aktualizace nastává tehdy, když expiruje doba platnosti záznamů. Teprve v tomto okamžiku sekundární server zjišťuje případné změny v záznamech na primárním serveru DNS a tyto změny nahrává i do své databáze.

Tento přístup může způsobit dočasnou nekonzistenci dat. Bylo by vhodnější, kdyby primární server informoval podřízené sekundární servery o tom, že došlo ke změně v zónovém souboru. změnila. Standard RFC 1996 [29] přináší mechanismus zasílání upozornění o změnách formou dotazu DNS NOTIFY.



Obrázek 3.11: Schéma zónového přenosu

Když primární server zjistí podle sériového čísla, že se změnila zóna, pošle speciální oznámení všem sekundárním serverům pro tuto zónu. Jejich adresy jsou uloženy v záznamu NS pro danou zónu. Jakmile sekundární server obdrží zprávu NOTIFY, potvrdí ji. Primární server poté přestane zprávu vysílat. Sekundární server poté jedná, jako kdyby došlo k uplynutí času aktualizace (refresh time) a požádá o přenos celé zóny (příkaz AXFR).

Pokud je zóna příliš obsáhlá, není vhodné při aktualizaci jednoho záznamu přenášet všechny informace. K tomu se používá tzv. přírůstkový přenos zón (incremental zone transfer, IXFR). Při tomto přenosu řekne podřízený sekundární server primárnímu serveru, kterou verzi zóny má a požádá ho o zaslání změn. Žádost IXFR v sobě obsahuje záznam SOA. Když server obdrží žádost o přírůstkový přenos, projde si lokální záznamy změn ve své databázi, kde je popsáno, čím se liší verze sekundárního a primárního serveru. Veškeré nalezené změny přepošle primární server sekundárnímu serveru nazpět. Pokud není nalezen záznam o změnách, pošle primární server celou zónu, jako kdyby obdržel příkaz AXFR.

Přenos celého zónového souboru je citlivá otázka z hlediska bezpečnosti. Při jednoduchých dotazech od resolveru odpovídá server DNS pouze na konkrétní dotazy. Uživatel neví, jaké záznamy existují v dané doméně a jaké IP adresy jsou zaregistrovány v DNS. Při načítání zónového souboru získává server informace o celé doméně (či subdoméně). Z tohoto důvodu je v konfiguraci primárního serveru DNS obvykle nastavena IP adresa sekundárního serveru, který může požádat o zónový přenos, kolik serverů se může současně přihlásit a podobně. Pro přenos zón se také doporučuje vytvořit zabezpečené spojení mezi oběma servery.

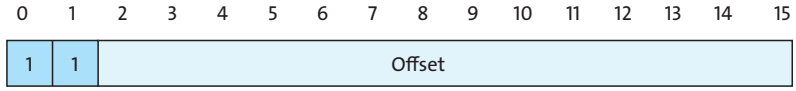
3.4.4 Komprese paketů DNS

Když se podíváme síťovým analyzátorem na komunikaci DNS, zjistíme, že přenášená data jsou daleko menší, než obsah, který vypíše například program `nslookup`. Je to proto, že při vytváření paketů DNS dochází ke komprimování textových dat. Tato činnosti je vhodná, protože některé řetězce se v paketu DNS opakují, například jména domény apod. Proto se místo výskytu jména používá ukazatel na první výskyt řetězce.

Každé doménové jméno se skládá z posloupnosti řetězců obsahující jména domén (například „eva“,„fit“,„vutbr“,„cz“). Jak vidíme, jména domén mají proměnlivou délku. Narozdíl například od jazyka C, není řetězec v paketu DNS ukončen znakem `0x00`, ale zapisuje se v paketech DNS jinak. Doménové jméno se ukládá jako posloupnost dvojic délka/hodnota. Každá dvojice obsahuje délku domény a název domény. Tato posloup-

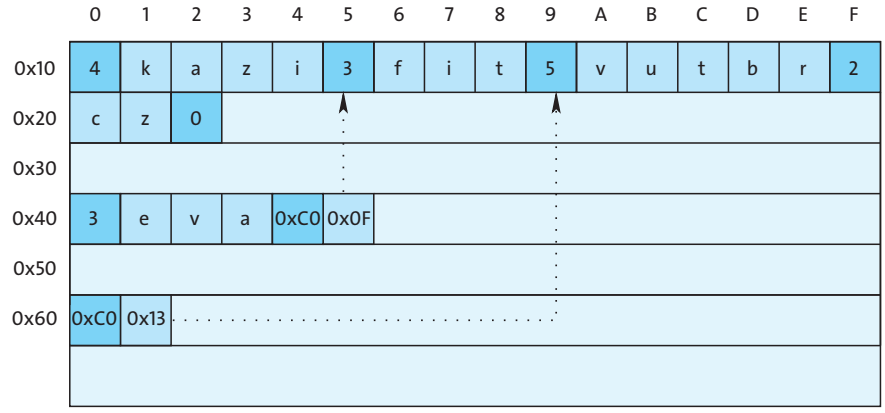
nost dvojic je ukončena nulovým oktetem. Například doménové jméno „eva.fit.vutbr.cz“ se v paketu DNS zapíše jako řetězec znaků „3 eva 3 fit 5 vutbr 2 cz 0“ (v zápisu ASCII)[4], kde první byte označuje délku řetězce, který následuje. Posloupnost je ukončena nulovým řetězcem, což je vlastně název kořenového uzlu ve stromu jmen DNS.

Toto je jeden typ uložení doménového jména. Protože se zejména suffixy doménových jmen opakují, využívá se při ukládání doménových jmen ukazatelů na předchozí výskyt jména (či jeho části). Zda se jedná o řetězec obsahující doménové jméno nebo o ukazatel, poznáme podle prvních dvou bitů délky, které mají speciální význam. Pokud jsou první dva bity tohoto bytu nulové, zbývajících šest bitů obsahuje délku řetězce, který následuje počínaje dalším oktetem. Pokud jsou první dva bity jedničkové (např. 0xC0), pak následující bity nevyjadřují délku paketu, ale ukazatel (pointer) na předchozí výskyt řetězce v paketu. Tento ukazatel je vlastně offset od začátku paketu DNS, kde se zmíněný řetězec vyskytuje. Offset tvoří jednak posledních šest bitů bytu s délkou, který začíná posloupností 11 (binárně), a také následujícím bytem. Dohromady tvoří ukazatel 14 bitů, viz obr. 3.12.



Obrázek 3.12: Formát ukazatele při kompresi paketu DNS

Použití ukazatelů vede k významné redukci velikosti paketu. Opakující se výskyty části doménového jména v paketu se nahradí ukazatelem na první výskyt. Pole `Offset` určuje vzdálenost od začátku paketu DNS, tj. od prvního oktetu DNS. Příklad komprese doménových jmen kazi.fit.vutbr.cz, eva.fit.vutbr.cz, vutbr.cz je naznačen na obrázku 3.13.



Obrázek 3.13: Příklad komprese doménových jmen v paketu DNS

Protože první dva bity délky se používají k rozlišení řetězce od ukazatele, pro délku se použije pouze následujících šest bitů. Proto může mít řetězec obsahující doménové jméno délku maximálně 2^6-1, tj. 63 bytů. Standard také uvádí, že maximální délka doménového jména (konkatenace všech domén) je 255 bytů.

4 V bytovém zápisu to bude posloupnost bytů „03 65 76 61 03 66 69 74 05 76 75 74 62 72 02 63 7a 00“

3.4.5 Programování komunikace v systému DNS

DNS!programování} Přestože systém DNS existuje desítky let a můžeme nalézt spoustu aplikací typu server i klient, občas je zapotřebí vytvořit vlastní program, který se dotazuje na informace v DNS. Pro programování lze použít schránky BSD, o kterých jsme mluvili v předchozí kapitole. Vytvoření zprávy DNS je z pohledu programátora složitější, neboť hlavička obsahuje spoustu parametrů a voleb, pro každý typ záznamu se používá trošku jiný formát sekcí dotaz a odpověď, pakety využívají komprimaci textový řetězců doménových jmen a další.

Proto se doporučuje využití knihoven `<resolv.h>` (man resolver) a `<arpa/nameser.h>`, které obsahují funkce pro posílání dotazů DNS a zpracování odpovědí. Každý program, který využívá tyto funkce, musí obsahovat následující hlavičkové soubory:

```
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/nameser.h>
#include <resolv.h>
```

V této části si představíme funkce pro vytváření a zasílání požadavků na DNS, procedury pro zpracování odpovědí od serveru a ukážeme si příklad aplikace – klienta DNS.

Funkce pro vytváření a zasílání požadavků

Následující funkce slouží k vytvoření požadavků na DNS a jejich odeslání na server. Funkce patří mezi systémové funkce operačního systému. Pro komunikaci není nutné explicitně vytvářet spojení UDP. Toto zajišťují samotné funkce.

`res_search()` – příprava dotazu na DNS

```
int  res_search(const char *dname, int class, int type, u_char  *answer, int len);}
```

Tato funkce pracuje na nejvyšší úrovni. Připravuje odeslání dotazu na DNS. Při volání ji používá např. funkce `gethostbyname()`. Funkce dostane doménové jméno `dname`, pomocí seznamu v resolveru tam doplní doménu a takto připravený dotaz předá funkci `res_query()` pro odeslání dotazu. Dotaz na doménové jméno je obsažen v prvním parametru `dname`, dále následují parametry dotazu a řetězec pro odpověď `answer` o velikosti `len`. Pokud není doménové jméno absolutní (FQDN), doplní se doména podle informací v proměnné `HOSTALIASES`.

`res_query()` – vytvoření dotazu na DNS

```
int res_query(const char *dname, int class, int type, u_char *answer,int len);}
```

Tato funkce vytváří dotaz na záznamy v DNS. Pomocí funkce `res_mkquery()` vytvoří požadavek v binární podobě, zašle jej funkcí `res_send()` serveru a vyhodnotí, zda v pořádku došel. Má podobné parametry jako `res_search()`. Liší se tím, že nedoplňuje doménové jméno.

`res_mkquery()` – vytváří zprávu do DNS

```
int res_mkquery(int op, const char *dname, int class, int type, const u_char
*data, int datalen, const u_char *newrr, u_char *buf, int buflen);
```

Funkce tvoří paket pro DNS. Vyplňuje políčka v hlavičce protokolu DNS, provádí kompresi doménových jmen, doplňuje část dotaz v paketu DNS. Vytvoří dotaz na záznam typu `op` v DNS. Vytvořený paket umístí do bufferu `buf`.

`res_send()` – zaslání dotazu na server

```
int res_send(const char *msg, int msglen, u_char *answer,int anslen);}
```

Funkce pošle naformátovanou DNS zprávu `msg` a počká na odpověď, kterou uloží do proměnné `answer`. Data posílá pomocí protokolu UDP, může použít i TCP.

`res_init()` – inicializace struktury `_res`

`int res_init(void);` Funkce `res_init()` načte konfiguraci ze souboru `resolv.conf` a inicializuje strukturu `_res`. Tato struktura je používaná dalšími funkcemi `res_xxx()`. Obsahuje adresy serverů DNS, maximální dobu pro čtení dat, implicitní domény a další nastavení získaná z lokální konfigurace `resolveru`.

Funkce pro zpracování odpovědí

Pro zpracování odpovědí od serveru se používají funkce z knihovny `<arpa/nameser.h>`:

`ns_initparse()` – inicializace a vytvoření datové struktury `ns_msg` pro zpracování odpovědi dalšími funkcemi `ns_xxx()`

`ns_msg_base()`, `ns_msg_end()`, `ns_msg_size()` – vrátí ukazatel na začátek a konec paketu včetně délky paketu.

`ns_msg_count()` – vrátí počet záznamů v jednotlivých sekcích paketu DNS, např. question, zone, answer, nameserver, update, additional a další.

`ns_rr_name()`,`ns_rr_type()`,`ns_rr_class()`,`ns_rr_rdata()`,`ns_rr_xxx()` – vrátí obsah příslušného pole ze zprávy DNS, např. pole `name`, `type`, `class`, `rdata` a dalších.

Příklad aplikace – zaslání dotazu na DNS a zpracování odpovědi

```
#include <netinet/in.h>
#include <arpa/nameser.h>
#include <arpa/inet.h>
#include <resolv.h> #include <netdb.h>

int resolve(const char *dname){
   in_addr_t addr4;
   register int i;
```

```
  int nRet, ipAddr[4] = {0, 0, 0, 0};
  char buf[ldname];
  if ((addr4 = inet_network(dname)) != -1){
    for (i = 0; addr4; ){
       ipAddr[i++] = addr4 & 0xFF;
       addr4 >>= 8;
    }
    sprintf(buf, "%u.%u.%u.%u.in-addr.arpa", ipAddr[i % 4],
       ipAddr[(i+1) % 4], ipAddr[(i+2) % 4], ipAddr[(i+3) % 4]);
    nRet = runDnsQuery(buf, ns_t_ptr, TRUE); // zjišťuje záznam PTR
  }
  else if ((nRet = runDnsQuery(dname, ns_t_a, FALSE))) // zjišťuje záznam typu A
    nRet = runDnsQuery(dname, ns_t_mx, FALSE); // zjišťuje záznam typu MX
  return nRet;
}
int runDnsQuery(const char *dname, int nType, int bNoDataError){
  u_char pResAnswer[lanswer],Uncompressed[ldname];
  int nResAnswerLen,i,j;
  ns_msg hMsg;
  ns_rr rr; const u_char *p;
  // zaslání dotazu a načtení odpovědi
  if (nResAnswerLen = (res_search(dname, ns_c_in, nType, pResAnswer, lanswer)) < 0)
    err();
  if (ns_initparse(pResAnswer, nResAnswerLen, &hMsg)) err(); // zpracování odpovědi
  for (i = 0; i < ns_msg_count(hMsg, ns_s_an); i++){ // zpracování záznamu
    if (ns_parserr(&hMsg, ns_s_an, i, &rr) < 0) err();
    switch (ns_rr_type(rr)){
       case ns_t_cname: // záznam typu CNAME
          if (nType == ns_t_mx) break;
          if (ns_name_uncompress(ns_msg_base(hMsg), ns_msg_end(hMsg),
             ns_rr_rdata(rr), Uncompressed, ldname) < 0) err();
          printf("%s is a nickname for %s\n", ns_rr_name(rr), szUncompressed);
       case ns_t_a: // záznam typu A
          printf("%s has address ", ns_rr_name(rr));
          p = ns_rr_rdata(rr);
       case ns_t_mx: // záznam typu MX
          p = ns_rr_rdata(rr);
       case ns_t_ptr: // záznam typu PTR
          if (ns_name_uncompress(ns_msg_base(hMsg), ...) < 0)
    }
  }
  return TRUE;
}
int main(int argc, char *argv[]){
  for (int i = 1; i < argc; i++)
    resolve(argv[i]);
  return 0;
}
```

3.4.6 Zjišťování informací v systému DNS

Většina aplikačních programů používá pro překlad doménových adres a dotazovaní se na záznamy v DNS služby resolveru, který je součástí operačního systému. Pro uživatele jsou tyto služby transparentní a většinou o nich ani neví.

Pokud chceme získat nějaké informace ze systému DNS, můžeme využít speciální programy, které z DNS servery komunikují. V následující části popíšeme tři základní programy pro komunikaci v DNS – `nslookup`, `host` a `dig`.

Program nslookup

Program `nslookup` slouží jako klient DNS, který posílá dotazy serveru DNS. Pracuje ve dvou režimech – interaktivním a řádkovém. Pokud spustíme program bez parametrů, přepneme se do interaktivního režimu. Většinou se používá `nslookup` v neinteraktivním řádkovém režimu. Po spuštění můžeme specifikovat, ke kterému serveru se přihlašujeme a jaké typy záznamů požadujeme. Pokud nezadáme typ záznamu, program vyhledává záznamy typu A a PTR.

V následujícím příkladu zjišťujeme mailový server pro doménu `stud.fit.vutbr.cz`. Ptáme se tedy serveru DNS kazi.fit.vutbr.cz na záznamy typu MX pro danou doménu. Server vrátí dva servery: primární poštovní server (eva.fit.vutbr.cz) a sekundární poštovní server (kazi.fit.vutbr.cz).

```
/home/matousp> nslookup -type=MX stud.fit.vutbr.cz kazi.fit.vutbr.cz
    Server:      kazi.fit.vutbr.cz
    Address:     147.229.8.12#53

    stud.fit.vutbr.cz     mail exchanger = 10 eva.fit.vutbr.cz.
    stud.fit.vutbr.cz     mail exchanger = 20 kazi.fit.vutbr.cz.
```

Narozdíl od resolveru komunikuje program `nslookup` pouze s jedním serverem, tj. neposílá rekurzivní dotazy. Pokud nenajde odpověď, musíme sami zkusit vyhledat doménový server, který obsahuje hledaná data. Program nepoužívá funkce knihovny `resolv.h`, ale vlastní funkce. Některé implementace programu podporují i přenos zón. Pokud program vrátí neautoritativní odpověď, je potřeba zjistit autoritativní server pro danou doménu a poslat dotaz na tento serveru. V následujícím serveru hledáme autoritativní odpověď na překlad doménového jména `www.cesnet.cz`.

```
1) /home/matousp> nslookup www.cesnet.cz // hledání odpovědi na lokálním serveru DNS
    Server:      147.229.9.43
    Address:     147.229.9.43#53

    Non-authoritative answer:
    Name: www.cesnet.cz
    Address: 195.113.144.230

2) /home/matousp> nslookup -type=NS cesnet.cz // hledání autoritativního DNS serveru
    Server:      147.229.9.43
    Address:     147.229.9.43#53
```

```
    Non-authoritative answer:
    cesnet.cz    nameserver = ns.cesnet.cz.
    cesnet.cz    nameserver = ns.ces.net.
    cesnet.cz    nameserver = decsys.vsb.cz.

    Authoritative answers can be found from:

3) nslookup www.cesnet.cz ns.cesnet.cz // hledání autoritativní odpovědi
    Server:      ns.cesnet.cz
    Address:     195.113.144.194#53

    Name:        www.cesnet.cz
    Address:     195.113.144.230
```

Program host

Dalším užitečným programem pro posílání dotazů na DNS je program `host`. Používá se pro překlad doménových adres na IP adresy a zpět. Při použití parametrů je možné vyhledávat i jiné typů záznamů či žádat o přenos zón. Následující příklad ukazuje možnosti použití programu `host`.

```
/home/matousp> host www.cesnet.cz
    www.cesnet.cz has address 195.113.144.230
    www.cesnet.cz has IPv6 address 2001:718:1:101::4
    www.cesnet.cz mail is handled by 100 rs.cesnet.cz.
    www.cesnet.cz mail is handled by 10 mail.cesnet.cz.

/home/matousp> host -t mx stud.fit.vutbr.cz

    stud.fit.vutbr.cz mail is handled by 20 kazi.fit.vutbr.cz.
    stud.fit.vutbr.cz mail is handled by 10 eva.fit.vutbr.cz.
```

Program dig

Program `dig` (Domain Information Groper) je nástroj pro zasílání dotazů na server DNS. Program zobrazuje odpovědi tak, jak je server vrací. Z tohoto důvodu jsou výpisy dlouhé. Pro základní dotazování jsou vhodnější nástroje `nslookup` nebo `host`.

Základní formát je „`dig @server name type`“, kde `server` je jméno dotazovaného serveru DNS, `name` je doménové jméno záznamu, který hledáme a `type` určuje typ záznamu, například ANY, A, MX a podobně.

`Dig` implicitně vyhledává záznamy typu A. Výpis programu `dig` je uspořádán podobně jako formát paketu DNS – obsahuje část `HEADER`, sekce `QUERY`, `ANSWER`, `AUTHORITY` a `ADDITIONAL`, viz následující ukázka. Program umí také přenášet zóny.

```
/home/matousp> dig www.cesnet.cz
; <<>> DiG 9.6.2-P1 <<>> www.cesnet.cz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61045
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 2
```

```
;; QUESTION SECTION:
;www.cesnet.cz.                 IN      A

;; ANSWER SECTION:
www.cesnet.cz.          63351   IN      A       195.113.144.230

;; AUTHORITY SECTION:
cesnet.cz.              85903   IN      NS      decsys.vsb.cz.
cesnet.cz.              85903   IN      NS      ns.cesnet.cz.
cesnet.cz.              85903   IN      NS      ns.ces.net.

;; ADDITIONAL SECTION:
ns.cesnet.cz.           85956   IN      A       195.113.144.194
ns.cesnet.cz.           85956   IN      AAAA    2001:718:1:1::2

;; Query time: 1 msec
;; SERVER: 147.229.9.43#53(147.229.9.43)
;; WHEN: Tue Oct 25 18:22:04 2011
;; MSG SIZE rcvd: 157
```

Podobně jako i další programy umožňuje `dig` nastavit způsob přenosu dat (UDP či TCP), možnost rekurzivního či nerekurzivního zpracování, cílový port a další. Pokud chceme vyhledávat v reverzním doménovém stromě, musíme zadat volbu `–x`.

3.5 Zabezpečení DNS

Služba DNS je veřejná. Využívá ji každý uživatel komunikující po Internetu. Odpovědi systému DNS jsou většinou přijímány jako důvěryhodné. DNS však komunikuje přes nezabezpečený datový kanál, kde je možné odpovědi odchytit, změnit či podvrhnout. Slabým místem jsou také paměti cache záložních serverů. Pokud „vnutíme“ serveru DNS informaci do paměti cache, tak ji dále neověřuje a na dotazy odpovídá uloženou hodnotou, aniž by se dotazoval autoritativního serveru. Útoky typu *cache poisoning* (otrávení paměti cache) vedly k vytvoření standardů na podepisování DNS zpráv – TSIG [30] a DNSSEC [32]. O typech útoků, bezpečnostních standardech a způsobu zabezpečení si povíme v této kapitole.

3.5.1 Bezpečnostní rizika v DNS

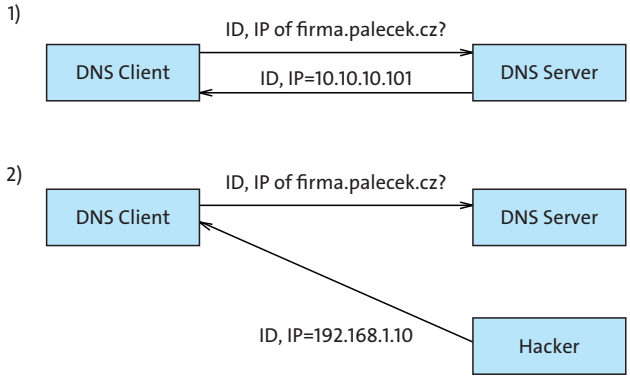
Dokument RFC 3833 [8] z roku 2004 se zabývá možnými způsoby ohrožení systému DNS a ukazuje, jak ochránit je třeba DNS chránit při zachování základního charakteru službu, tj. že jde o veřejnou službu, která je všeobecně dostupná kterémukoliv uživatele Internetu. Mezi dva důležité úkoly, které je potřeba z pohledu bezpečnosti DNS zajistit, patří:

- integrita dat (tj. data nejsou změněna během přenosu)
- autentizace zdroje dat (tj. odesilateli dat mohu důvěřovat)

Základní mechanismus, který se používá pro zajištění integrity dat a autentizace zdroje dat, je systém veřejných klíčů a podepisování záznamů DNSSEC. Dále je to technika TSIG pro podepisování záznamů soukromým klíčem. Mechanismy DNSSEC a TSIG budou podrobně popsány v další kapitole. Mezi základní třídy útoků na systém DNS patří:

Odposlech paketů. Při tomto útoku útočník sleduje komunikaci a v případě dotazu na DNS vrátí nesprávnou odpověď, případně v odpovědi pozmění nějaké informace. Tím se mu podaří například přesměrovat provoz na server s jinou IP adresou, posílání elektronické pošty apod. Řešením je zajištění integrity paketů DNS pomocí podepisování záznamů DNSSEC. Pro zónový přenos je vhodná například symetrická autentizace pomocí TSIG.

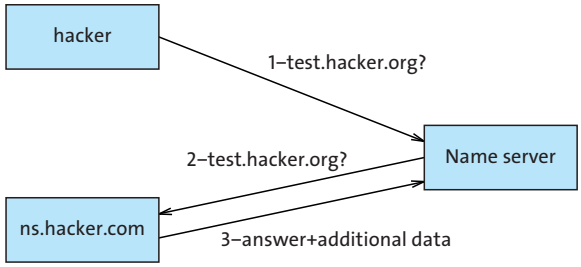
Hádání ID paketu a predikce odpovědi. Identifikační číslo paketu DNS je pouze 16-bitové. Číslo cílového port serveru je známé (53), číslo portu klienta je také 16-bitová hodnota. Existuje tedy pouze 2^{32} možných kombinací těchto hodnot, což je proveditelné pro útok hrubou silou. Sledováním síťového provozu můžeme tyto hodnoty předvídat nebo hádat. Při odchycení dotazu je resolver velmi náchylný k přijetí falešné odpovědi. Resolver, který zkontroluje podpis DNSSEC či TSIG, pozná neautorizovanou odpověď. Na obrázku 3.14 je znázorněn případ, kdy útočník podvrhne odpověď a vnutí klientovi nesprávnou IP adresu serveru. K útoku potřebuje znát ID původního dotazu.



Obrázek 3.14: Podvržení odpovědi útočníkem

Zřetězení jmen (otrávení paměti cache). Zřetězení jmen je podmnožinou útoků typu otrávení vyrovnávací paměti cache. Základem útoku je vložení nesprávné informace do paměti cache. To lze dosáhnout například změnou informací v poli RDATA odpovědi, zejména v záznamech CNAME, NS a DNAME. Klient vyšle dotaz, na nějž mu útočník vnutí odpověď, která je pozměněná. V části RDATA jsou přidány do sekce `Additional` jiná data např. nesprávné IP adresy autoritativních serverů DNS apod, viz obr. 3.15.

Takový útok se odehrál na InterNIC v červenci 1997, kdy adresa www.internic.net byla přesměrována na IP adresu www.alternic.net. V tomto případě útočník vložil do rozšiřující části odpovědi další doménové adresy s podrobnějšími informace. Z pohledu DNS se zdálo, že jde o rozšiřující informace, které souvisí se zaslaným dotazem. Většinu těchto útoků lze odvrátit kontrolou podpisů prostřednictvím DNSSEC, neboť resolver může ověřit, zda odesilatel zná tajný klíč, jehož odpovídající veřejný klíč je i s ověřením přístupný na veřejném místě v DNS.



Obrázek 3.15: Vložení nesprávných informací do paměti cache (otrávení)

Znemožnění služby (Denial of Service). Systém DNS je zranitelný útokem DoS, jak jsme si popisovali v kapitole 3.2.4. Mechanismy podepisování a autentizace DNSSEC a TSIG tyto útoky neřeší. Spíše je zhoršují, neboť podepisování je časově i výpočetně náročná operace. Řešení útoků DoS je na úrovni konfigurace DNS serveru (omezení počtu dotazů) či vlastní sítě (kontrola počtu navázaných spojení z jedné IP adresy).

Odmítnutí domény. Odmítnutí domény souvisí s otázkou, zda kontrolovat (ověřovat) neexistenci domény. Problém je, zda by měl být resolver schopný detekovat zrušení dat útočníkem v odpovědi DNS. Tato otázka se stále dost diskutuje. Je jasné, že je potřeba ověřit neexistenci neúplně zapsaných (wildcards) záznamů. DNSSEC obsahuje mechanismus, který umožňuje určit, které autoritativní názvy existují v zóně a který typy autoritativních záznamů existují pro dané názvy pomocí záznamů NSEC a NSEC3.

3.5.2 TSIG – podepisování transakcí

Mechanismus TSIG (Transaction Signatures) popisuje autentizaci transakcí DNS pomocí symetrické kryptografie. TSIG je popsán ve standardu RFC 2845 [30] z roku 2000, v roce 2003 byl popsán aktualizovaný přístup nazvaný GSS-TSIG, viz RFC 3645 [34]. Mechanismus využívá autentizace se sdíleným tajným klíčem a jednocestnou hašovací funkci HMAC-MD5 [13]. TSIG, narozdíl od DNSSEC, nezajišťuje autentizaci a integritu samotných dat, ale pouze odesilatele a příjemce, tedy datových přenosů (transakcí) mezi nimi.

Podepisování TSIG využívá nový typ záznamu TSIG (250), který obsahuje autorizační kód (hash) celého záznamu DNS (bez pole TSIG) spolu s názvem algoritmu, času podpisu a dalších údajů. Záznam TSIG nesmí být uložen v paměti cache a neobjeví se ani v zónových souborech. Je jedinečný pro každou odesílanou zprávu, proto ho není nutné ukládat.

TSIG zajišťuje autentizaci zdroje (serveru DNS) a integritu dat pomocí autentizačního kódu (kryptografického součtu) HMAC-MD5. Záznam TSIG se vkládá do sekce `Additional`. Jakmile se sestaví paket DNS, spočítá se kontrolní součet a vloží záznam TSIG. Po přijetí zprávy odesilatel zkontroluje typ hašovacího algoritmu, název klíče a platnost autentizačního kódu. Spočítá se autentizační kód záznamu a porovná s hodnotou ve zprávě. Pokud se shodují, je záznam přijat.

Klíče jsou definovány v konfiguračním souboru serveru DNS `named.conf`, viz následující ukázka pro zónu `palecek.cz`:

```
server ns1:
    key ns1-ns2.palecek.cz. {
       algorithm hmac-md5;
       secret ``RCMEsfegvM0VnuA==';
    };



    zone ``palecek.cz'' {
       type master;
       file ``db.palecek.cz'';
       allow-transfer { key ns1-ns2.palecke.cz.; };
    };
```

```
server ns2:
    key ns1-ns2.palecek.cz. {
       algorithm hmac-md5;
       secret ``RCMEsfegvM0VnuA==';
    };
    server 192.249.249.3 {
       keys {ns1-ns2.palecek.cz.;};
    }
    zone ``palecek.cz'' {
       type slave;
       masters {192.249.249.3;};
       file ``db2.palecek.cz';
    };
```

Tento příklad ukazuje konfiguraci primárního serveru `ns1` a sekundárního serveru `ns2`. Oba servery sdílejí stejný klíč a stejný zabezpečovací algoritmus HMAC-MD5. Primární server přijímá pouze podepsané žádosti na přenos zóny `allow-transfer`. Položka `server` u `ns2` vytváří asociaci mezi serverem a klíčem. Tato položka říká, že veškeré žádosti na server `192.249.249.3` se budou podepisovat klíčem `ns1-ns2.palecek.cz.`

Klíč lze vytvořit příkazem `dnssec-keygen`, který vygeneruje záznam DNS (soubor `.key`) a soukromý klíč `.private`, viz následující ukázka.

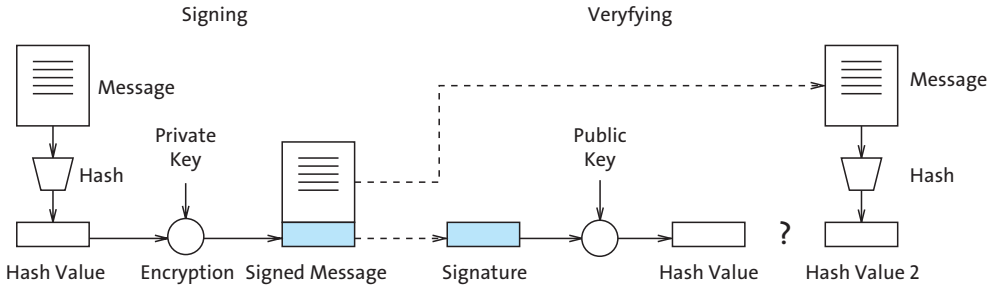
```
>dnssec-keygen -a hmac-md5 -b 256 -n HOST palecek.cz

Kpalecek.cz.+157+22718.key:
    palecek.cz. IN KEY 512 3 157 6Vl4IroYv42NJiHaJRybpQsXVvLDVEG4otheLKH1/io=

Kpalecek.cz.+157+22718.private:
    Private-key-format: v1.2
    Algorithm: 157 (HMAC_MD5)
    Key: 6Vl4IroYv42NJiHaJRybpQsXVvLDVEG4otheLKH1/io=
    Bits: AAA=
```

TSIG předpokládá použití klíče pro každou dvojici serverů DNS. Toto je nevhodné pro servery komunikující s velkým množstvím jiných serverů, neboť počet klíčů roste exponenciálně s počtem serverů. Problémem je také v distribuci klíčů, neboť se jedná o soukromý klíč, který je potřeba bezpečným způsobem předat druhé straně.

Mechanismu TSIG je možné použít i pro resolvery, kteří se dotazují jednoho serveru či několika málo serverů. Stejně tak je tento mechanismus vhodný pro záložní servery, které se odkazují na jeden konkrétní server. Například programy `dig` nebo `nsupdate` umí načíst klíč ze souborů `.key` a použít ho k zaslání podepsaných dynamických aktualizací. Použití pro resolvery na běžných pracovních stanicích je obtížné z hlediska distribuce klíčů. TSIG neumožňuje vytvářet bezpečné spojení se všemi servery DNS, ale pouze s těmi, s kterými se dohodl na klíčích. Pro autentizaci komunikace s veřejnými servery se používá DNSSEC.

3.5.3 DNSSEC – podepisování záznamů

Standard DNSSEC (DNS Security Extension) [32] z roku 2005 definuje rozšíření protokolu DNS pro zabezpečení přenosu dat v systému DNS pomocí asymetrické kryptografie s použitím veřejného a soukromého klíče. Narozdíl od symetrické kryptografie a mechanismu TSIG, existují u DNSSEC dva klíče – soukromý pro podepisování a veřejný pro ověření podpisu. Princip podepisování a ověřování podpisu je na obrázku 3.16.



Obrázek 3.16: Generování a ověření elektronického podpisu

DNSSEC používá nové typy záznamů v DNS – záznam DNSKEY pro uložení veřejného klíče, záznam RRSIG obsahující podpis konkrétního záznamu, dále záznamy NSEC pro sekvenční uspořádání záznamů v doméně a záznam DS pro ověření podpisu záznamu DNSKEY pomocí vyšší autority.

Všechny tyto záznamy se používají pro *podepisování zón* pomocí DNSSEC. Podepsaná zóna znamená, že zónový soubor obsahuje kromě všech záznamů zóny (typu A, MX, CNAME, PTR apod.) také elektronický podpis ke každému z těchto záznamů. Elektronický podpis – přesněji řečeno podepsaný kontrolní součet (hash) záznamu – se uloží do záznamu RRSIG. Použije se k ověření integrity záznamu a autentizaci vlastníka. Podpis se ověřuje oproti veřejnému klíči, který je přístupný v záznamu DNSKEY.

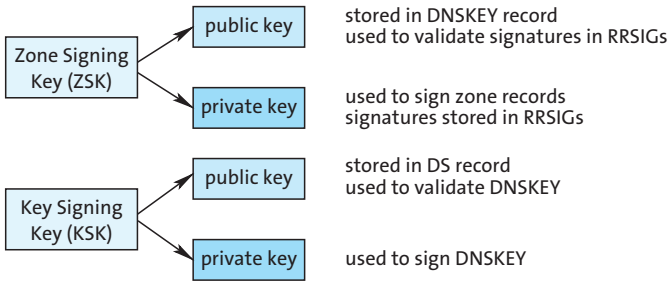
Pro vytvoření podpisu a následné ověření pomocí asymetrické kryptografie potřebujeme vygenerovat klíč. Protože se jedná o asymetrickou kryptografii, používá se dvojice klíčů – soukromý a veřejný klíč.

- Soukromý klíč slouží k podepisování. Klíč se uchovává na bezpečném místě.
- Veřejný klíč slouží k ověření podpisu. Je volně dostupný. Pro kontrolu pravosti veřejného klíče se používají například certifikáty X.509. U DNSSEC se pravost veřejného klíče ověřuje podepsáním vyšší autoritou pomocí KSK (viz dále).

Oba klíče jsou algoritmicky závislé, tzn. k danému soukromému klíči přísluší konkrétní veřejný klíč. Oba se vygenerují současně. Více se o principech asymetrické kryptografie můžete dočíst v [20].

Pomocí dvojice soukromý a veřejný klíč jsme schopni podepsat záznamy v DNS a také zkontrolovat, zda je daný podpis platný. Co však chybí, je potvrzení, že můžeme daným klíčům a podpisům důvěřovat. Co se stane, když si útočník sám vytvoří podvržené záznamy, které podepíše svým klíčem a nabídne nám k ověření svůj platný veřejný klíč? Pak ověříme pravost záznamů a jsme spokojeni. Přesto se jedná o nepravdivé (neautorizované) údaje.

Z tohoto důvodu nám nestačí jenom klíč pro podpis zóny ZSK (Zone Signing Key). Je potřeba ještě klíč pro ověření těchto klíčů, takzvaný KSK (Key Signing Key). Pro podepisování klíčů opět použijeme asymetrickou kryptografii. Máme tedy dva páry klíčů ZSK a KSK, které se použijí pro vybudování důvěry mezi servery DNS. Jejich vztah a uložení v DNS ukazuje obrázek 3.17.

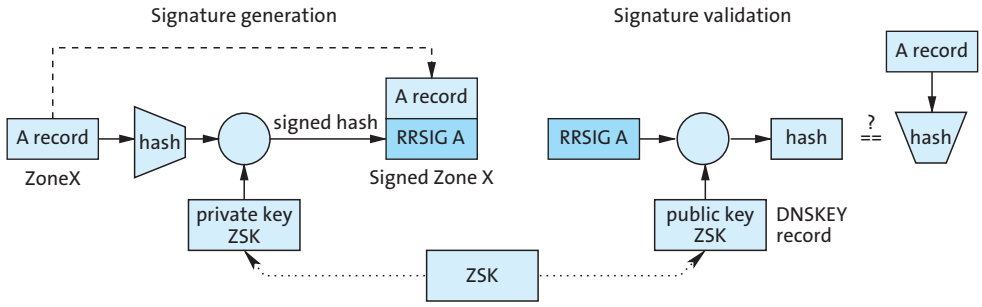


Obrázek 3.17: Klíč pro podpis zóny (ZSK) a klíč pro podepisování klíčů (KSK)

Tyto dva páry klíčů, ZSK a KSK, tvoří základ systému zabezpečení DNSSEC. Používají se k podepisování a validaci zón a k podpisování a validaci klíčů pro podpis zón. Klíče KSK vytváří (přesněji řečeno veřejný klíč KSK) vytváří tzv. *důvěryhodný vstupní bod SEP (Security Entry Point)*, viz [17]. Společně vytváří propojení KSK a ZSK tzv. *řetězec důvěry (chain of trust)*.

Řetězec důvěry

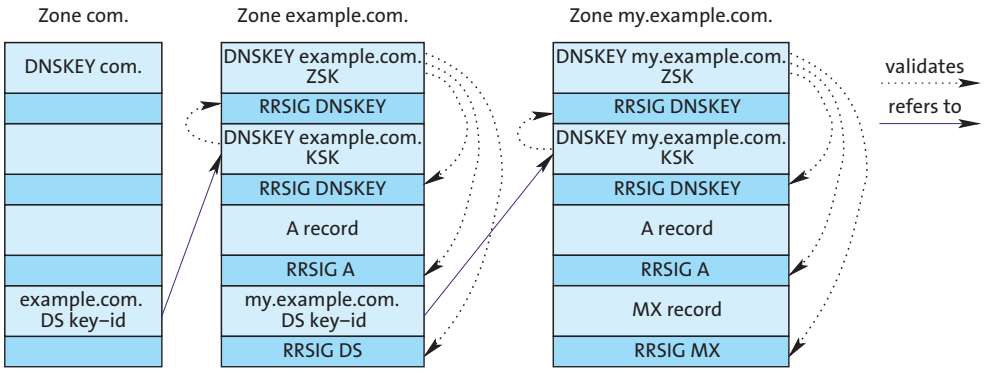
Podepsaná zóna obsahuje veřejný klíč DNS (uložený v záznamu DNSKEY), podpisy záznamů (uložené v záznamech RRSIG), odkazy na další záznamy (záznamy NSEC), případně záznam DS ověřující klíč zóny v DNSKEY. Pokud zóna neobsahuje tyto záznamy, jedná se o nepodepsanou zónu. Příklad podepsání a ověření záznamu v DNS je na obrázku 3.18.



Obrázek 3.18: Podepsání záznamu typu A a ověření podpisu

Jak jsme si již řekli, záznam typu DNSKEY obsahuje veřejný klíč pro podepisování záznamů v dané zóně. Tento klíč se používá k ověření podpisu záznamů uložených v RRSIG. Podpisy se připojují k záznamům typu A, CNAME, MX, NS a podobně v dané

zóně. Podepsaný je i záznam DS. DNSSEC určuje i umístění záznamů. Záznam DS by měl být umístěn v nadřazené zóně, viz obrázek 3.19.



Obrázek 3.19: Řetězec důvěry pomocí záznamů DS a DNSKEY

Záznamy DNSKEY a DS vytváří posloupnost (řetězec) podepsaných záznamů navzájem potvrzující pravost podpisů, tzv. *řetězec důvěry* (chain of trust). Pravost záznamu DS ověřuje podpis v záznamu RRSIG, který se kontroluje pomocí veřejného klíče uloženého v záznamu DNSKEY dané zóny. Záznam DS obsahuje kontrolní součet (hash) jiného klíče DNSKEY. Tento klíč je tedy autentizován informací v DS. Záznam DS se obvykle nachází v nadřazené zóně, která se nazývá *bodem delegace* (delegation point). Dvojici souvisejících záznamů DNSKEY a DS nazýváme *důvěryhodný pevný bod* (trust anchor).

Může se stát, že k dané zóně A neexistuje bod delegace, to jest neexistuje záznam DS v nadřazené zóně, který by obsahoval kontrolní součet záznamu DNSKEY v zóně A. Taková zóna je podepsaná, netvoří však řetězec důvěry. Nazývá se *ostrůvkem bezpečnosti* (island of security) a podepsané záznamy této zóny bychom měli používat opatrně.

Z tohoto popisu vyplývá, že je nutné, aby existoval nějaký počáteční bod řetězce důvěry. Protože je řetězec důvěry budován nad stromem DNS, je vhodné, aby jeho počátečním bodem byla kořenová zóna. V červenci 2010 došlo tedy k podepsání kořenové zóny, čím se vytvořil základní důvěryhodný bod pro vytváření řetězce důvěry (tzv. root trust anchor). Problematiku podepisování zón, delegace a rotace klíčů řeší RFC 4641 [18]. V následující části se podrobněji podíváme na jednotlivé záznamy DNS, které se používají v rozšíření DNSSEC.

3.5.5 Záznamy DNS pro DNSSEC

Rozšíření DNSSEC se vyvíjelo od roku 1999. Původní standard RFC 2535 [9] používal pro podepisování dat v DNSSEC záznamy typu KEY, SIG, NXT, které se dnes už nepoužívají. Místo nich byly roku 2005 standardizovány záznamy DNSKEY, RRSIG a NSEC. V roce 2008 došlo k aktualizaci a místo záznamů NSEC se doporučuje používat NSEC3, viz RFC 5155 [19].

Přehled nejdůležitějších záznamů DNS pro rozšíření DNSSEC je uveden v tabulce 3.7. Jejich význam si více popíšeme v následující části.

Záznam	Význam	Standard
DNSKEY (DNS Key Record)	veřejný klíč pro ověření podpisu	RFC 4034
RRSIG (Resource Record Signature)	podpis pro daný záznam	RFC 4034
DS (Delegation Signer)	potvrzení pravosti klíče v DNSKEY	RFC 4034
NSEC (Next-Secure Record)	odkaz na další záznam v doméně	RFC 4034
NSEC3 (NSEC version 3)	viz NSEC bez procházení zóny	RFC 5155
NSEC3PARAM (NSEC3 parameters)	parametry pro NSEC4	RFC 5155

Tabulka 3.7: Záznamy DNS použité při zabezpečení DNSSEC

Záznam DNSKEY – DNS Key Record} DNSSEC používá pro podepisování a autentizaci záznamů DNS kryptografii s veřejným klíčem. Veřejný klíč se ukládá do záznamu DNSKEY. Jednotlivé záznamy v zóně se podepisují privátním klíčem ZSK, pro ověření se používá odpovídající veřejný klíč v záznamu DNSKEY. Součástí záznamu je kromě klíče také typ klíče, algoritmus pro ověřování (např. RSA/MD5, Diffie-Hellman, DSA/SHA-1 či RSA/SHA-1) a další, viz následující ukázka.

```
> dig +multi fit.vutbr.cz dnskey
fit.vutbr.cz.    7943    IN    DNSKEY 257 3 5 (
                               AwEAAcgoOp3tkJBYisSHYRdrkrtKT0+tk2mtm3uypUIY feFAgyQSZtw-
                               zj3wGKkBRWpdsShxXPOxUjSruYTPeX1j0 D4HjIx/s7YIRqWYdCw5CUB-
                               3JXLk5ygLcW+MI/+MBR+RW tKHWBi73/+ZrZMBR6na3V5yTlbn4VsM
                               p0h53f8TMzU= ) ; key id = 59533
fit.vutbr.cz.    7943    IN    DNSKEY 256 3 5 ( AwEAAcyccLng09xZedVqpf/0DkSqLwngQXS1e-
                               9FeOdV6 D4hcrljjzHMl sD5l7UCdcXdnyJnS6UZwHlV/l2U4v7j ) ;
                               key id = 35421
```

Výše uvedené záznamy obsahují veřejné klíče KSK a ZSK pro doménu `fit.vutbr.cz`. Jedná se o veřejný klíče KSK (hodnota 257), který se použije pro podpis ZSK a veřejný klíč ZSK (hodnota 256). Oba klíče používající k autentizaci algoritmus RSA-SHA-1 (hodnota 5). Záznamy také obsahují identifikátor klíče (key id), který se používá v podpisu (v záznamu RRSIG) jako odkaz na klíč, podle kterého je možné zkontrolovat podpis. Délka klíče v ukázce je zkrácena.

Oba klíče ZSK a KSK se umisťují do zóny, pro kterou jsou vygenerovány. Klíč ZSK (ID 35421) se použije pro podepsání všech záznamů v zóně včetně záznamu DNSKEY obsahující ZSK. Je jediná výjimka – záznam DNSKEY obsahující ZSK se podepisuje KSK. Otisk klíče KSK se pak umístí v záznamu DS nadřazené domény k vytvoření řetězce důvěry. Podpisy klíčů v zónovém souboru `fit.vutbr.cz` vypadají následovně:

```
fit.vutbr.cz.    3750    IN    RRSIG DNSKEY 5 3 14400 20111201081631 (
                               20111101081631 35421 fit.vutbr.cz. rYwrvsFfERrEkbT4WddUW-
                               ZMKBUiHJ35h29AU4hAywfDZ Uaa+Z4/iIpY/86bNobbZppT3mcd0exw-
                               MunDEnac= )
fit.vutbr.cz.    3750    IN    RRSIG DNSKEY 5 3 14400 20111201081631 (
                               20111101081631 59533 fit.vutbr.cz. BKAYwfPKJsZC+BKKGgd0Ip-
                               4VG5rAkmWApY1p2FItvsTh 3SsUovKET159WUvllr4hpD1REH61OlgO-
                               G6leAea//68d AOfKsyeR1WWbCT3ylcfGFsMV5RFdDt15rA== )
```

Z výpisu je vidět, že první podpis je proveden klíčem s ID 35421, tj. ZSK. Jedná se tedy o podpis záznamu DNSKEY obsahující KSK. Druhý podpis je proveden KSK (id 59533), což znamená, že je to podpis záznamu DNSKEY obsahující ZSK provedeného KSK (typ 257). Je také vidět, že podpis je delší než podpis předchozího záznamu. Je to proto, že klíč KSK se generuje s větší délkou a obměňuje se méně častěji. Protože má větší délku, naroste i délka podpisu. Více o formátu a použití záznamu RRSIG pro podpis je v další části.

Záznam RRSIG – Resource Record Signature

Záznamy RRSIG obsahují elektronický podpis jednotlivých záznamů v podepsané zóně. Přesnější je říci, že podpis RRSIG se nevztahuje k jednotlivým záznamům, ale k množině záznamů, které mají stejné doménové jméno (vlastníka), typ, třídu a TTL. Z těchto údajů a obsahu záznamů se vypočítá podpis. Formálně může podpis zapsat následujícím způsobem, kde znak | označuje konkatenaci [32]:

```
signature = sign (RRSIG_RDAT | RR(1) | RR(2) ...)
RR(i) = owner | type | class | TTL | RDATA length | RDATA
```

Záznam RRSIG obsahuje také dobu platnosti podpisu, použitý algoritmus, jméno podepisující autority a příznak, který se odkazuje na veřejný klíč pro ověření podpisu.

```
> dig +dnssec +multi fit.vutbr.cz mx
fit.vutbr.cz.     13905    IN       MX 20 eva.fit.vutbr.cz.
fit.vutbr.cz.     13905    IN       MX 10 kazi.fit.vutbr.cz.
fit.vutbr.cz.     13905    IN       RRSIG MX 5 3 14400 20111201081631 (
                                    20111101081631 35421 fit.vutbr.cz. s+6p3B1IzpYpf/
                                    7PVN3Cd8aouew/OnIucu52mdEIBv+X oRK27lHnpR5JY+UaPkw-
                                    plpo7gwKddg/FvyjghoM= )
```

Výše uvedený výpis obsahuje záznamy MX pro doménu `fit.vutbr.cz` a elektronický podpis těchto dvou záznamů. Podpis obsahuje dobu platnosti podpisu (TTL podpisu musí odpovídat TTL příslušného záznamu), dále typ algoritmu RSA-SHA-1 (hodnota 5), počet řetězců v názvu záznamu (hodnota 3, tj. fit.vutbr.cz), dobu platnosti záznamu MX a dobu expirace (hodnota 20111201081631 znamená 1.12.2011 v 8:16.31). Hodnota 35421 je odkaz (key tag) na veřejný klíč ZSK v záznamu DNSKEY, který se použije pro ověření podpisu. Název `fit.vutbr.cz` je doménové jméno podepisující entity, to jest vlastníka záznamu DNSKEY. Zbytek záznamu RRSIG tvoří vlastní podpis uložený v kódování Base64.

Záznam NSEC – Next-Secure Record

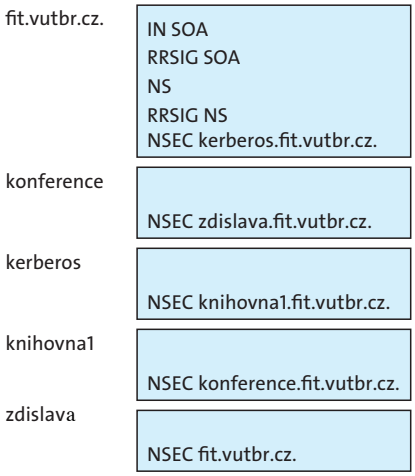
Záznam NSEC obsahuje dvě důležité informace: další doménové jméno záznamu v doméně a typy záznamů DNS, které se vztahují k tomuto doménovému jménu.

Záznamy NSEC se používají k autoritativnímu potvrzení neexistence záznamu v doméně. Tím, že záznam NSEC ukazuje na další doménové jméno, nám vlastně říká, že mezi současným doménovým jménem a dalším jménem, není žádná další položka. Bez záznamů NSEC by nešla autoritativně ověřit neexistence záznamů, protože neexistující záznam nemůže mít podpis. Pak by mohl útočník na dotaz na neexistující doménové jméno poslat podvrženou odpověď obsahující požadovaný záznam.

Použití záznamů NSEC vyžaduje seřazení jednotlivých záznamů DNS podle doménových jmen. Standard RFC 4034 [32] popisuje řazení kanonických doménových jmen. Kanonické řazení doménových jmen porovnává doménová jména postupně podle domén zprava doleva. Domény na stejné úrovni se porovnají jako dva řetězce podle ASCII hodnot a bez ohledu na velká či malá písmena. Pokud se nejpravější doménová jména shodují, porovnávají se domény na druhé úrovni a tak dále. Kanonické řazení demonstruje následující příklad:

```
example
a.example
yljkjljk.a.example
Z.a.example
zABC.a.EXAMPLE
z.example
\001.z.example
*.z.example
\200.z.example
```

Řazení je logické, tj. nesouvisí s aktuálním umístěním záznamů v zónovém souboru. Poslední záznam seřazení podle řazení kanonických doménových jmen se odkazuje na první záznam seřazeného seznamu doménových jmen. Příklad řazení je na obrázku 3.20.



Obrázek 3.20: Příklad řazení záznamu v podepsaném zónovém souboru `fit.vutbr.cz`

Tento obrázek ukazuje řazení záznamů DNS podle doménových jmen v následujícím pořadí: `fit.vutbr.cz`, `kerberos`, `knihovna1`, `konference` a `zdislava`. Poslední záznam ukazuje opět na první záznam, tj. doménové jméno `fit.vutbr.cz`. Na tomto příkladu můžete také vidět, že postupným doptáváním se na záznamy NSEC jsme schopni vyčíst všechny záznamy dané domény, tj. zjistit, jaká doménová jména se používají pro zařízení v dané doméně. Jedná se o závažnou otázku bezpečnosti sítě. Proto byl roku 2008 navržen alternativní způsob pro autentizovaný způsob potvrzení nee-

xistence doménového jména zvaný NSEC verze 3 [19]. O tom, jak funguje, si povíme později.

Záznam NSEC se umisťuje zpravidla nakonec záznamů k doménovému jménu. Za ním následuje ještě podpis v záznamu RRSIG. Následující příklad ukazuje použití záznamu NSEC.

```
fit.vutbr.cz.   86398   IN NSEC kerberos.fit.vutbr.cz. NS SOA MX TXT RRSIG NSEC DNSKEY
fit.vutbr.cz.   86398   IN RRSIG NSEC 5 3 86400 20111202094352 ( 20111102094352 35421
                        fit.vutbr.cz. nmUzpEnG4vIZN0uVRo5ja/cSEZG0Jec8fMPLAAiU2c2X
                        1PmpGYjLB6AFNCTEYM9+ZnEAACWY7xsYEFWxYoc= )
```

Jedná se o záznam NSEC u doménového jména `fit.vutbr.cz.`, který ukazuje na další záznam (NEXT) v zóně následujícím za jménem `fit.vutbr.cz.` podle kanonického řazení. Zároveň obsahuje seznam záznamů všech záznamů u jména `fit.vutbr.cz.`, což jsou záznam NS, SOA, MX, TXT, RRSIG, NSEC a DNSKEY. K záznamu NSEC je opět připojený klíč s podpisem záznamu.

Záznam DS – Delegation Signer

Záznam DS se odkazuje na záznam DNSKEY a používá se k autentizaci klíče v tomto záznamu. Záznam DS obsahuje odkaz na klíč DNSKEY, typ algoritmu (např. SHA-1) a otisk klíče (digest) v DNSKEY. Otisk klíče stačí pro identifikaci a autentizaci veřejného klíče uloženého v záznamu DNSKEY. Pomocí záznamu DS může resolver ověřit klíč v záznamu DNSKEY, na který se záznam DS odkazuje.

Záznamy DS a DNSKEY mají stejného vlastníka (tj. obsahují stejné doménové jméno), ale jsou uloženy na různých místech. Záznam DS se uchovává v rodičovské zóně. Například záznam DS pro doménu `fit.vutbr.cz.` je uložen v zónovém souboru `vutbr.cz.` Naopak, odpovídající klíč DNSKEY je uložen v zóně potomka, tj. `fit.vutbr.cz.`, viz obrázek 3.19. Příklad záznamů DNSKEY a DS je v následující ukázce.

```
>dig +dnssec +multi fit.vutbr.cz DNSKEY
fit.vutbr.cz.   12346   IN DNSKEY 257 3 5 ( AwEAAcgoOp3tkJBYisSHYRdrkrtKT0+tk2mtm3uy-
                        pUIY feFAgyQSZtwzj3wGKkBRWpdsShxXPOxUjSruYTPeX1j0 WBi73/+ZrZM-
                        BR6na3V5yTlbn4VsMp0h53f8TMzU= ) ; key id = 59533
>dig +dnssec +multi fit.vutbr.cz DS
fit.vutbr.cz.   84249   IN DS 59533 5 1 ( 7814F57DE4BB7FD48CF3DD952549D6CFF162B21F )
fit.vutbr.cz.   83951   IN RRSIG DS 5 3 86400 20111120120731 ( 20111021120731 39756
                        vutbr.cz. ZO18YyedzSvAwmUiw1FI275zupL+BwC5RgIvCDUZ21+C
                        NqxDAzbdONrMZverbGP6PCsMR691U7mCqkj+YMVtq1JM Dn1a3XQ-
                        PmWqrZ4KILlJ+LX291aA6BD/f8H3x+IY= )
```

Zde vidíme KSK pro doménu `fit.vutbr.cz`, který je uložený v záznamu DNSKEY v zónovém souboru `fit.vutbr.cz`. Tento klíč s ID 59533 se použije pro podepsání zónového klíče ZSK. Odkaz na tento klíč je umístěn v záznamu DS, který je umístěn v rodičovské zóně `vutbr.cz`. Zároveň je podepsán klíčem ZSK rodičovské zóny `vutbr.cz` s ID 39756.

Záznam NSEC3 – Next-Secure Record version 3

Nedílnou součástí zabezpečení DNSSEC jsou záznamy NSEC, které slouží pro potvrzení neexistence záznamu. Jak jsme si ukázali výše, nevýhodou použití záznamů NSEC je

to, že umožňují načtení celé zóny pomocí průchodu seznamu NSEC. Postupným vyčítáním záznamů NSEC (zone enumeration) jsme schopni získat všechna doménová jména dané domény včetně typů záznamů, které jsou pro dané jméno registrované v DNS. Přestože spousta serverů DNS filtruje přenos zón pouze na důvěryhodné servery DNS, pomocí procházením řazeného seznamu kanonických jmen v doméně jsme schopni tyto informace získat bez jakéhokoliv omezení.

Toto je hlavní motivace k návrhu nového typu záznamu NSEC3 [19]. Tento záznam zajišťuje stejnou funkci jako NSEC, tj. potvrzení neexistence záznamu. Proto potřebuje podobně jako NSEC vytvořit řazenou posloupnost všech registrovaných doménových jmen, aby mohly detekovat neexistující doménové jméno.

NSEC3 však nepoužívá pro řazení kanonická jména, ale číselnou hodnotu hašovací funkce, kterou aplikuje na původní jméno. Výsledné číselné hodnoty numericky řadí do seznamu (tzv. *hash order*). Toto pořadí je stejné, jako u řazení kanonických jmen dle RFC 4034 [32]. Příklady výstupů hašovací funkce obsahuje následující ukázka:

```
H(example)         = 0p9mhaveqvm6t7vbl5lop2u3t2rp3tom
H(a.example)       = 35mthgpgcu1qg68fab165klnsnk3dpvl
H(ai.example)      = gjeqe526plbf1g8mklp59enfd789njgi
H(ns1.example)     = 2t7b4g4vsa5smi47k61mv5bv1a22bojr
H(ns2.example)     = q04jkcevqvmu85r014c7dkba38o0ji5r
H(w.example)       = k8udemvp1j2f7eg6jebps17vp3n8i58h
H(*.w.example)     = r53bq7cc2uvmubfu5ocmm6pers9tk9en
H(x.w.example)     = b4um86eghhds6nea196smvmlo4ors995
H(y.w.example)     = ji6neoaepv8b5o6k4ev33abha8ht9fgc
H(x.y.w.example)   = 2vptu5timamqttgl4luu9kg21e0aor3s
H(xx.example)      = t644ebqk9bibcna874givr6joj62mlhv
H(2t7b4g4vsa5smi47k61mv5bv1a22bojr.example) = kohar7mbb8dc2ce8a9qvl8hon4k53uhi
```

Záznam NSEC3 obsahuje informaci o tom, jaká hašovací funkce je použita (hash algorithm), jaká je inicializační sekvence (tzv. *salt*), kolik iterací hašovací funkce se provede nad původním doménovým jménem (iterations), ukazatel na další hašované jméno a bitová mapa, která obsahuje seznam typů DNS, které jsou vytvořeny pro dané doménové jméno. Příklad podepsané zóny s použitím záznamů NSEC3 je v následující ukázce:

```
example. 3600 IN SOA    ns1.example. bugs.x.w.example. 1 3600 300 ( 3600000 3600 )
                 RRSIG  SOA 7 1 3600 20150420235959 20051021000000
                        ( 40430 example.
                        Hu25UIyNPmvPIVBrldN+9Mlp9Zql39qaUd8i VI2LmKusbZsT0Q== )
                 NS     ns1.example.
                 NS     ns2.example.
                 RRSIG  NS 7 1 3600 20150420235959 20051021000000 ( 40430 example.
                        PVOgtMK1HHeSTau+HwDWC8Ts+6C8qtqd4pQJ CnMXjtz6SyObxA== )
                 DNSKEY 257 3 7 AwEAAcUlFV1vhmqx6NSOUOq2R/dsR7Xm3upJ ( AbsUdblMFin-
                        8CVF3n4s= )
                 RRSIG  DNSKEY 7 1 3600 20150420235959 ( 20051021000000 12708 exam-
                        ple. AuU4juU9RaxescSmStrQks3Gh9FblGBlVU31 MGQZf3bH+QsCtg== )
                 NSEC3PARAM 1 0 12 aabbccdd
```

```
                RRSIG   NSEC3PARAM 7 1 3600 20150420235959 ( 20051021000000 40430
                        example. rN05XSA3Pq0U3+4VvGWYWdUMflOdxqnXHwJ TLQsjlkynh-
                        G6Cg== )
0p9mhaveqvm6t7vbl5lop2u3t2rp3tom.example. NSEC3 1 1 12 aabbccdd (
                        2t7b4g4vsa5smi47k61mv5bv1a22bojr MX DNSKEY NS SOA NSEC3PARAM
                        RRSIG )
                RRSIG   NSEC3 7 2 3600 20150420235959 20051021000000 ( 40430 exam-
                        ple. IBHYH6blRxK9rC0bMJPwQ4mLIuw85H2EY762 BOCXJZMnpuwhpA== )
2t7b4g4vsa5smi47k61mv5bv1a22bojr.example. A 192.0.2.127
                RRSIG   A 7 2 3600 20150420235959 20051021000000 ( 40430 example.
                        K+iDP4eY8I0kSiKaCjg3tC1SQkeloMeub2GW k8p6xHMPZumXlw== )
                NSEC3   1 1 12 aabbccdd ( 2vptu5timamqttgl4luu9kg21e0aor3s A RRSIG )
                RRSIG   NSEC3 7 2 3600 20150420235959 20051021000000 ( 40430 exam-
                        ple. 4TFoNxZuP03gAXEI634YwOc4YBNITrj413iq NI6mRk/r1dOSUw== )
```

Záznam NSEC3PARAM obsahuje parametry NSEC3 (hašovací algoritmus, příznaky, iterace, inicializační sekvenci), kterou jsou potřebné pro autoritativní servery, aby mohli vypočítat hašovací funkci pro doménové jméno.

Nevýhodou návrhu NSEC3 je, že není zpětně kompatibilní s dřívějšími standardy DNSSEC. Pokud například resolver nerozumí NSEC3, nebude schopen validovat odpovědi. Kompatibilita například vyžaduje, aby se pro generování podpisů DNSKEY používali algoritmy DSA-NSEC3-SHA1 nebo RSASHA1-NSEC3-SHA1. V současné době není mnoho zabezpečených domén, které by používaly záznamy NSEC3.

3.5.6 Zabezpečení záznamů DNS pomocí DNSSEC

Jak jsme si již řekli, DNSSEC narozdíl od TSIG podepisuje samotné záznamy soukromým klíčem DNS serveru. Podpis uložený v záznamech RRSIG a veřejný klíč v záznamu DNSKEY jsou součástí zónových souborů. Pravost veřejného klíče pro podpis zóny potvrzuje nadřazená autorita pomocí záznamu DS (Delegation Signer). Pomocí odkazů na nadřazené autority se buduje tzv. řetěz důvěryhodnosti (trust, authentication chain), který funguje na podobném principu jako certifikáty a jejich potvrzování. Použití DNSSEC zahrnuje následující kroky:

1. Vygenerování veřejného a soukromého klíče ZSK pro podpisy záznamů

```
>> dnssec-keygen -a RSASHA1 -b 512 -n ZONE palecek.cz.

Kpalecek.cz.+005+35205.private:
Private-key-format: v1.2     // privátní klíč
Algorithm: 5 (RSASHA1)
Modulus: upmhLafZL+ZrdYb2qGp6Avq2g+PCFzEI6xpV0ZeqBaM8J356fjUSJlKVToxPCMT8c2zcJL
PublicExponent: AQAB
PrivateExponent: K7mmQCmNxiCrW53lnK9eqeYP8LXlHEQcP+Mq6vMsHygSB3smvv42o45EeGR+fK
Prime1: 3UySK8tda81GgtE0zy6nyznEfVzLqjQPHOLzhxtIXljEUeigJ2QoOHMSa2bzvs6R3by/7Q/
Prime2: 19wqWwmBtPPE9We/c/tBMAuOmjN1EKom4K+UDDWM2bIdxcQugVsOF0ux5XaFsFfJu3uj3mc
Exponent1: dAwWTs7rI3/W7PzGI6tEpGabql3CE3QPRMeiiwP8lDUaZhWF/oIgMVHuo9GuGi37Aftb
```

```
Exponent2: gHVEO4MPznjXOFeyZroaVBY8meWPgAhSNJooV4+yr4bBEwUpy889Nn3Uc0KyUnwb0FfG
Coefficient: SFAYEx+hbU12K4wldwAcWJbZFIZW9StbJ+XEyjyWluXgi9GTPn/5IS117FwjjOYsUZ

Kpalecek.cz.+005+35205.key:          // veřejný klíč
palecek.cz. IN DNSKEY 256 3 5 AwEAAbqZoS2n2S/ma3WG9qhqegL6toPjwhcxCOsaVdGXqgWj
41 EiZSlU6MTwjE/HNs3CS7UGKL/jOypLzP7wen7MAbMQHQW8L4xdw7SGQN XorEPuN/zWvNU8+v9VI
```

2. Vygenerování veřejného a soukromého klíče KSK pro podepsání ZSK

```
>> dnssec-keygen -a RSASHA1 -b 2048 -n ZONE -f KSK palecek.cz.

Kpalecek.cz.+005+12094.private:      // privátní klíč
Private-key-format: v1.2
Algorithm: 5 (RSASHA1)
Modulus: +vmUVz8p7YhxLov95n6mxodRqKwaPr2XHGQt/FM9Uns4b+iQ0UfD2bU2OFezJeUT0D8dNx
         ugQBQc3WLaJhDUsWJBVwRa46/QATzOVeGxGZu1WhmuTiQ5QdWrgarOh4dlOTinr6d2Lrlk
PublicExponent: AQAB
PrivateExponent: Y3TSLzztrZ7bU1nQtS3Ng3EOkCwbfm99WZftjUejTnNcKyTbdsPcvgKMyQkgCk
..

Kpalecek.cz.+005+12094.key:          // veřejný klíč
palecek.cz. IN DNSKEY 257 3 5 AwEAAfr5lFc/Ke2IcS6L/eZ+psaHUaisGj69lxxkLfxTPVJ7O
            NFHw9m1NjhXsyXlE9A/HTcSba7oEAUHN1i2iYQ1LFiQVcEWuOv0AE8zlXhsdaRmbtVo
```

3. Vložení veřejného klíče KSK a ZSK do zónového souboru (pro ověřování podpisů).

```
$TTL 3600
@       IN    SOA     palecek.cz. root.palecek.cz. (
                              20050322  ; Serial
                              3600      ; Refresh
                              900       ; Retry
                              3600000   ; Expire
                              3600 )    ; Minimum
        IN    NS      palecek.cz.
        IN    MX  0   mail.palecek.cz.
pc1     IN    A       10.10.10.102
palecek.cz.  IN DNSKEY 257 3 5 AwEAAfr5lFc/Ke2IcS6L/eZ+psaHUaisGj69lxxkLfxTPVJ
             NFHw9m1NjhXsyXlE9A/HTcSba7oEAUHN1i2iYQ1LFiQVcEWuOv0AE8zlXhsdaRmbt
palecek.cz.  IN DNSKEY 256 3 5 AwEAAbqZoS2n2S/ma3WG9qhqegL6toPjwhcxCOsaVdGXqg 41
             EiZSlU6MTwjE/HNs3CS7UGKL/jOypLzP7wen7MAbMQHQW8L4xdw7SGQN XorEPuN/
             zWvNU8+v9
```

4. Podepsání zóny soukromým klíčem.

```
>> dnssec-signzone -o palecek.cz
db.palecek.cz db.palecek.cz.signed:  // zónový soubor pro doménu palecek.cz
   palecek.cz.      3600  IN SOA   palecek.cz. root.palecek.cz. (
                                   20050322    ; serial
```

```
                                3600        ; refresh (1 hour)
                                900         ; retry (15 minutes)
                                3600000     ; expire (5 weeks 6 days 16 hours)
                                3600        ; minimum (1 hour)
                                )
                    3600  RRSIG   SOA 5 2 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. So0a/txX-
                                w3zB5FRDE+LG21xKpbfu/pNcdRhC vYB2XjIKWKAha7Wy-
                                4qV0hOCORwQ= )
                    3600  NS      palecek.cz.
                    3600  RRSIG   NS 5 2 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. hN5HeB7zUgu-
                                1OqexZ/ATFll1HEOKiXcRrmBS Jy01TOpCSVJoUbZBxDUJ-
                                8vy9lFM= )
                    3600  MX      0 mail.palecek.cz.
                    3600  RRSIG   MX 5 2 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. yG8UAE5ltKS-
                                Gc+X8Y71zhtHESIovOrC0UM+R ns811M2yMewuN3H0V8j-
                                zwRg3mq4= )
                    3600  NSEC    firma.palecek.cz. NS SOA MX RRSIG NSEC DNSKEY
                    3600  RRSIG   NSEC 5 2 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. o14yMIow+t-
                                cALntZmlO50A3wv9fHkZWuyS3I gNH2Ws4gWP97jzi-
                                xkmR41rtmxhQ= )
                    3600  DNSKEY  256 3 5 ( AwEAAbqZoS2n2S/ma3WG9qhqegL6toPjwhcx
                                COsaVdGXqgWjPCd+en41 ) ; key id = 35205
                    3600  IN      DNSKEY 257 3 5 (
                                AwEAAfr5lFc/Ke2IcS6L/eZ+psaHUaisAhoJ lxxkLfxTP-
                                VJ7OG/okNFHw9mlNjhXsyXlE9A/ HTcSba7oEAUHN1i2iY-
                                Q1LFiQVcEWuOv0AE8z ) ; key id = 12094
                    3600  RRSIG   DNSKEY 5 2 3600 20111208135222 (
                                20111108135222 12094 palecek.cz. sdd0lmjckou-
                                62MQRbhY7MhjpJsLU6irzqp+q t0v71KwfLg28FfRUe2LO-
                                KGMhlE6u/SZF4C4= )
                    3600  RRSIG   DNSKEY 5 2 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. GfFFaG5o51wR-
                                3LXR8CLubS+3pFQFGDi6DIc5 Ej9su0lzkdrqX+aDwuA-
                                65XySiejfEUPMgq/n
pc1.palecek.cz.     3600  IN A   10.10.10.102
                    3600  RRSIG   A 5 3 3600 20111208135222 (
                                20111108135222 35205 palecek.cz. HiZ+9q/yxikb-
                                KzuUPFBKiOzHwLcts34yDO/f UAMxTKFnpmuovWWnDq5gA-
                                l1O2yY= )
                    3600  NSEC    pc2.palecek.cz. A RRSIG NSEC
...
```

Ve výše uvedeném výpisu můžeme vidět, že pro každý záznam existuje podpis (záznam RRSIG). Dále jsou v zónovém souboru záznam s klíčem (DNSKEY) a jeho pod-

pis. Záznam NSEC definují řazení, tj. určují, jaký další záznam následuje po předchozím. Tyto záznam slouží ke kontrole negativních odpovědí.

5. Umístění záznamu DS pro ověření podpisu zónového klíče do nadřazené domény.

```
dsset-palecek.cz:
palecek.cz.  IN DS 12094 5 1 51121AA0B28D46698518A2A843C55614940BD812
```

6. Nastavení zónového souboru v konfiguraci DNS serveru.

```
zone ``palecek.cz'' {
   type master;
   file ``db.palecek.cz.signed'';
};
```

Při každé změně zónového souboru musí být změněný souboru znovu podepsán. Budování řetězu důvěry a hierarchické podepisování klíčů pomocí záznamů DS je popsáno ve standardu RFC 4034 [32]. Další důležitou věcí, kterou musí administrátor zajistit, je rotace klíčů po expiraci jejich platnosti a znovu podepisování záznamů. Na to už dnes existují automatizované prostředky, které tuto výměnu klíčů zajišťují.

Mechanismus DNSSEC tvoří komplexní hierarchický systém pro zabezpečení informací v DNS proti podvržení a upravení. Pomocí principů veřejné kryptografie je možné vybudovat důvěryhodný řetěz DNS serverů. Podepisování DNSSEC záznamů se začíná ve světě více prosazovat a postupně se podepisují jednotlivé zóny a budují se řetězce důvěry pro ověření podepsaných zón. Díky podepsání kořenového serveru DNS lze jednotlivé podepsané domény propojit do jednoho stromu a ověřit pomocí důvěryhodné autority.

Mezi nevýhody DNSSEC patří zejména nárůst objemu dat přenášených přes DNS (podepsané záznamy jsou několikanásobně delší), závislost na nadřazené doméně (a jejím klíči pro ověřování). Při expiraci ZSK se může stát, že doména bude nedostupná.

Shrnutí

Překlad doménových adres DNS je důležitou a nezbytnou součástí pro fungování komunikace v Internetu. Systém DNS obsahuje databázi doménových jmen a dalších informací, které jsou uloženy v záznamech DNS. Službu DNS tvoří soustava serverů DNS, které tuto databázi spravují, a dále resolvery, které k databázi přistupují. Doménových prostor DNS je hierarchicky uspořádán a jeho správa je rozdělena na nižší subjekty. Koordinaci DNS a správu nejvyšších domén řídí organizace ICANN, která spravuje národní domény a generické domény první úrovně. Domény nižší úrovně (subdomény) spravují vlastníci domén. Prostor doménových adres vytváří kořenový strom, kde jednotlivé uzly jsou domény nižší úrovně a cesta ve stromu tvoří doménovou adresu. Pokud cesta zahrnuje i kořen, jedná se o absolutní doménové jméno (fully qualified domain name). Fyzicky jsou části doménového prostoru rozděleny do zón a uloženy na doménových serverech.

Základním úkolem systému DNS je překlad (mapování) doménových adres na IP adresy. Pro zpětný (reverzní) překlad adres IPv4 existuje speciální doména `.in-addr.arpa.`, která obsahuje prostor číselných IP adres rozdělených hierarchicky do stromu podle jednotlivých bytů adresy. Konkrétní informace v systému DNS jsou uloženy v datových strukturách, které se nazývají záznamy DNS (resource records). Existuje velké množství záznamů, které uchovávají různé typy dat. Mezi nejpoužívanější záznamy patří záznamy typu A, které mapují doménovou adresu na IP adresu, záznamy typu PTR pro reverzní překlad, záznamy NS pro mapování domény na autoritativní doménový server. Záznamy typu MX obsahují adresu poštovního serveru pro danou doménu, záznamy SOA uchovávají informace o platnosti záznamů a správě domény. Pro potřeby IP telefonie se využívá například záznamů NAPTR a SRV. Zabezpečení DNS využívá záznamu DNSKEY, RRSIG, NSEC či DS. IP adresy protokolu IPv6 jsou uloženy v záznamech AAAA, reverzní mapování v PTR.

Komunikaci v DNS zajišťuje aplikační protokol DNS, který zpravidla běží nad transportním protokolem UDP. Pro delší přenosy (nad 512 bytů), což jsou zejména zónové přenosy, se využívá transportní protokol TCP, port 53. PDU protokolu DNS obsahuje hlavičku s informacemi o dotazu/odpovědi, sekci dotazu, odpovědi a dodatečné (upřesňující) informace.

Většina komunikace probíhá mezi resolvery a servery DNS. Jedná se o jednoduché dotazy na vyhledávání. Dalším typem komunikace je synchronizaci mezi primárním a sekundárním serverem DNS, kterému se říká zónový přenos. K zónovému přenosu dochází buď po uplynutí platnosti záznamů v DNS na sekundárních či záložních serverech, nebo zasláním informace DNS NOTIFY. Sekundární server žádá buď aktualizaci celé zóny nebo pouze dat, která se změnila od poslední aktualizace. Resolver je většinou součástí operačního systému. Obsahuje funkce pro vytváření, posílání a dekódování požadavků na systém DNS. Funkce resolveru lze využít k napsání vlastních programů pro vyhledávání informací v DNS. Pro vyhledávání informací v DNS lze využít programy `nslookup`, `host` a `dig`.

Informace v DNS jsou nezbytné pro správnou činnost komunikace po Internetu. V případě změny, poškození či nedostupnosti může dojít k omezení a výpadkům spojení. Existují případy vnucení nesprávných údajů do paměti `cache` sekundárních a záložních serverů. Z tohoto důvodu se snaží DNS zajistit integritu a autenti-

zaci dat. Privátnost (šifrování) není potřeba zajišťovat, neboť systém DNS je ze své podstaty navržený jako veřejný systém.

Existují dva způsoby zajištění bezpečnosti přenosu dat v DNS – systém TSIG a DNSSEC. TSIG využívá podepisování transakcí (nikoliv záznamů) mezi komunikujícími stranami pomocí tajného klíče. Toto omezuje jeho činnost pouze na servery pod jednou administrativní správou. Mechanismus DNSSEC naopak vytváří strukturu vzájemně důvěryhodných serverů pomocí veřejných klíčů ověřovaných vyšší autoritou. Jednotlivé záznamy v DNS jsou podepsány soukromým klíčem. Protože jsou podpisy i veřejné klíče pro ověření uloženy ve speciálních záznamech, dochází k enormnímu nárůstu objemu dat. Také zpracování podpisů a ověřování je výpočetně náročné. Přesto dochází k postupnému nasazování rozšíření DNSSEC.

Použitá literatura a standardy

[1] *ISO 3166 Maintenance Agency – ISO's focal point for country codes.* International standard, ISO, 1974.

[2] A. Guldbrandsen, P.Vixie, and L.Esibov. *A DNS RR for specifying the location of services (DNS SRV).* RFC 2782, February 2000.

[3] A. Gustafsson. *Handling of Unknown DNS Resource Record (RR) Types.* RFC 3597, September 2003.

[4] S. Bradner, L. Conroy, and K. Fujiwara. *The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM).* RFC 6116, March 2011.

[5] B. Wellington. *Secure Domain Name System (DNS) Dynamic Update.* RFC 3007, November 2000.

[6] C. Davis, P. Vixie, T. Goodwin, and I. Dickinson. *A Means for Expressing Location Information in the Domain Name System.* RFC 1876, January 1996.

[7] C. Everhart, L. Mamakos, R. Ullmann, and P. Mockapetris. *New DNS RR Definitions.* RFC 1183, October 1990.

[8] D. Atkins and R.Austein. *Threat Analysis of the Domain Name System (DNS).* RFC 3833, August 2004.

[9] D. Eastlake. *Domain Name System Security Extensions.* RFC 2535, March 1999.

[10] D. Eastlake and A. Panitz. *Reserved Top Level DNS Names.* RFC 2606, June 1999.

[11] E. Gunduz, A. Newton, and S. Kerr. *IRIS: An Address Registry (areg) Type for the Internet Registry Information Service.* RFC 4698, October 2006.

[12] G. Huston. *Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa").* RFC 3172, September 2001.

[13] H. Krawczyk, M. Bellare, and R. Canneti. *HMAC: Keyed-Hashing for Message Authentication.* RFC 2104, February 1997.

[14] B. Hoeneisen, A. Mayrhofer, and J. Livingood. *IANA Registration of Enumservices: Guide, Template, and IANA Considerations.* RFC 6117, March 2011.

[15] J. Postel. *Domain Name System Structure and Delegation.* RFC 1591, March 1994.

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. *SIP: Session Initiation Protocol.* RFC 3261, June 2002.

[17] sO. Kolkman, J. Schlyter, and E. Lewis. *Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag.* RFC 3757, April 2004.

[18] O. Kolman and R. Gieben. *DNSSEC Operational Practices.* RFC 4641, September 2006.

[19] B. Laurie, G. Sisson, R. Arends, and D. Blacka. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence.* RFC 5155, March 2008.

[20] Wanbo Mao. Modern Cryptography. Theory and Practice. Prentice Hall, 2004.

[21] M. Lottor. *DOMAIN ADMINISTRATORS OPERATIONS GUIDE*. RFC 1033, November 1987.

[22] M. Mealling. *Dynamic Delegation Discovery System (DDDS) Part Five: URI.ARPA Assignment Procedures*. RFC 3405, October 2002.

[23] M. Mealling. *Dynamic Delegation Discovery System (DDDS), Part One: The Comprehensive DDDS*. RFC 3401, October 2002.

[24] M. Mealling. *Dynamic Delegation Discovery System (DDDS), Part Three: The Domain Name System (DNS) Database*. RFC 3403, October 2002.

[25] M. Mealling and R. Daniel. *The Naming Authority Pointer (NAPTR) DNS Resource Record*. RFC 2915, September 2000.

[26] P. Beertema. *Common DNS Data File Configuration Errors*. RFC 1537, October 1993.

[27] P. Mockapetris. *Domain Names — Concepts and Facilities*. RFC 1034, November 1987.

[28] P. Mockapetris. *Domain Names — Implementation and Specification*. RFC 1035, November 1987.

[29] P. Vixie. *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)*. RFC 1996, August 1996.

[30] P. Vixie, O. Gudmundsson, D. Eastlake, and B. Wellington. *Secret Key Transaction Authentication for DNS (TSIG)*. RFC 2845, May 2000.

[31] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE)*. RFC 2136, April 1997.

[32] R. Arends, R. Austein, M. Larson, D. Massay, and S. Rose. *Resource Records for the DNS Security Extensions*. RFC 4034, May 2005.

[33] R. Bush, D. Karrenberg, M. Kosters, and R. Plzak. *Root Name Server Operational Requirements*. RFC 2870, June 2000.

[34] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead, and R. Hall. *Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)*. RFC 3645, October 2003.

[35] S. Thomson, C. Huitema, V. Ksinant, and M. Soussi. *DNS Extensions to Support IP Version 6*. RFC 3596, October 2003.