Implementace replikace údajů o doménových jménech z relační databáze do DNS serveru

Implementation of Information Replication from a Relations Database to DNS Server

Ivo Valerián

Bakalářská práce
2009

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ivo VALERIÁN**
Studijní program:	**B 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Implementace replikace údajů o doménových jménech v rámci domény bata.cz z relační databáze do DNS serveru BIND9**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na dané téma.**
2. **Popište vlastnosti DNS serveru BIND9.**
3. **Popište možnosti nástroje nsupdate.**
4. **Popište databázový systém PostgreSQL.**
5. **Implementujte program pro replikaci dat z PostgreSQL do DNS BIND9.**

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Mistrovství v počítačových sítích, Stephen J. Bigelow, Computer Press 2004, ISBN: 80-251-0178-9.**
2. **Administrace a diagnostika sítí, James M. Kretchmar, Libor Dostálek, Computer Press 2005, ISBN: 80-251-0345-5.**
3. **Umění programování v Unixu, Eric S. Raymond, Computer Press 2004, ISBN: 80-251-0225-4.**
4. **Apache Server 2 – Kompletní příručka administrátora, Mohammed J. Kabir, Computer Press 2004, ISBN: 80-251-0319-6.**
5. **PostgreSQL – Praktický průvodce, Bruce Momjian, Computer Press 2003, ISBN: 80-722-6954-2.**
6. **Naučte se Perl za 21 dní, Laura Lemay, Computer Press 2002, ISBN: 80-7226-616-0.**
7. **The Perl Directory, URL: >http://www.perl.org<.**
8. **PostgreSQL, URL: >http://www.postgresql.org<.**
9. **Jemný úvod do jazyka PL/pgSQL PostgreSQL, Pavel Stěhule, URL: >http://postgresql.ok.cz/doc/plpgsql.html<.**
10. **DNS, BIND, DHCP, LDAP and Directory Services, URL: >http://www.bind9.net<.**

Vedoucí bakalářské práce: **Ing. Miroslav Červenka**
Ústav aplikované informatiky

Datum zadání bakalářské práce: **20. února 2009**

Termín odevzdání bakalářské práce: **1. června 2009**

Ve Zlíně dne 13. února 2009

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato práce je součástí projektu firmy BAŤA, akciová společnost. Zabývá se aktualizováním dat v DNS serveru BIND v distribuci Debian z relační databáze PostgreSQL pomocí nástroje nsupdate a implementovaném v jazyce Perl, popisuje i princip a konfiguraci jednotlivých částí obsažených v tomto projektu.

Klíčová slova:

DNS, nsupdate, PostgreSQL, BIND ,Perl, DBI

ABSTRACT

This work is a part of an IT project within the BAŤA company. It deals with the problem of data synchronisation between a BIND DNS server running on a Debian based machine and a relational database based on PostgreSQL with assistance of the nsupdate tool. This work describes principles and configuration of individual parts included in this project. All code created and described here is implemented in the Perl scripting language.

Keywords:

DNS, nsupdate, PostgreSQL, BIND, Perl, DBI

Za spolupráci a vstřícnost bych chtěl poděkovat Ing. Norbertu Volfovi, dále svému vedoucímu práce Ing. Miroslavu Červenkovi.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

V dnešní době stále se rozšiřujících informačních systémů jsou stále častější různá softwarová vylepšení a zjednodušení už existujících nástrojů a služeb, proto i tato práce se tímto bude zabývat. Stále se rozrůstající společnosti by v dnešní době nemohly bez informačních technologií vůbec existovat. Jednou z nich je i firma Baťa, která má zastoupení na celém světě a jejich síť informačních systémů je obrovská a proto se musí stále zdokonalovat a hledat možnosti jak správu těchto technologií zjednodušit nebo alespoň zrychlit. Jednou z těchto vymožeností je provoz vlastních DNS serverů, které se starají o doménu. Pro správu DNS serverů je vynaloženo velké úsilí a mnoho prostředků. Další vymožeností je i dynamická aktualizace DNS serveru, která je předmětem této práce, proto zde bude popsáno mnoho věcí týkající se právě DNS serverů. Zároveň jsou důležité databázové systémy, které obsahují mnoho důležitých dat, které lze použít k různým zjednodušením, v tomto případě jsou data pro aktualizaci DNS serveru použitá právě z databázového systému PostgreSQL. Tím zjednodušením je právě způsob propojení těchto dvou systémů, aby mohly mezi sebou nějakým způsobem spolupracovat a tím i zdokonalovat celý komplex informačního systému.

I. TEORETICKÁ ČÁST

1 DNS

V dnešním světe je k Internetu připojeno obrovské množství počítačů. Jak si o všech udržet přehled, když náleží do tolika různých zemí, sítí a administračních skupin? Vše záleží na dvou důležitých kusech infrastruktury. Jedním je systém DNS (DNS je zkratka z anglického The Domain Name System) neboli systém doménových jmen, jež spravuje identitu počítačů a druhým je systém směrovacího internetového systému, který sleduje vzájemné propojení počítačů.

Ačkoliv systém DNS začal postupem času sloužit k několika různým účelům, jeho hlavním smyslem je mapování mezi IP adresami a jmény počítačů. Uživatelské programy a samotní uživatelé nejraději používají jména, kdežto síťový software nižší úrovně rozumí číslům, tudíž IP adresám. Systém DNS proto poskytuje řešení, kterým všechny uspokojí. Systém DNS také začal hrát významnou roli v přístupu k webovým serverům a ve směrování pošty.

Systém DNS je v podstatě distribuovaná databáze. Což znamená, že jedna organizace uchovává informace o svých počítačích a druhá organizace uchovává informace zase jen o svých počítačích, ale obě organizace nějakým způsobem spolupracují při sdílení dat, když se třeba jedna organizace potřebuje podívat na nějaká data z druhé organizace.

1.1 Historie DNS

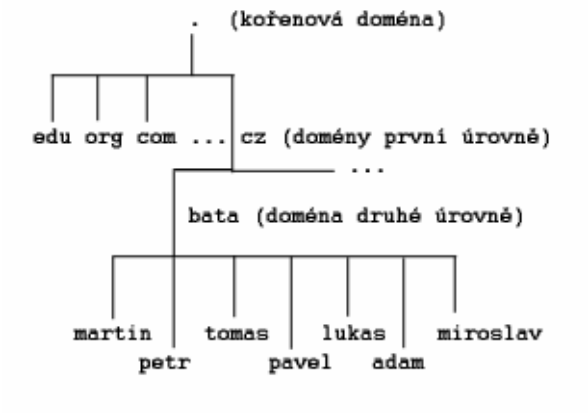
Dříve než se objevily DNS servery se mapování mezi IP adresami a jmény počítačů uchovávalo v jediném textovém souboru, řízeném centrálně a šířené všem počítačům na Arpanetu. Postup pro pojmenování počítače zahrnoval zjištění, zda-li někdo na celém světě již neobsadil jméno, které jste chtěli, protože jména počítačů nebyla hierarchická. Soubor obsahující mapování mezi jmény počítačů a adresami byl neustále zastaralý a jeho aktualizace zabíraly značnou část kapacity Arpanetu.

Brzy se ukázalo, že i když byla statická tabulka počítačů vhodným řešením pro malou síť, tak pro stále se zvětšující Arpanet byla nevhodná. Problémy statické tabulky proto vyřešil systém DNS pomocí dvou zásadních principů, tím prvním jsou hierarchická jména počítačů a tím druhým distribuovaná odpovědnost. Systém DNS specifikoval Paul Mockapetris v dokumentech RFC 882 a 883 (rok 1983) a aktualizovány byly v dokumentech RFC 1034 a 1035 (rok 1987).

Dokumenty RFC 1034 a 1035 jsou sice stále považovány za základní specifikaci systému DNS, ale různé části datových záznamů a protokolu byly za posledních deset let nahrazeny a doplněny pomocí více než třiceti jiných RFC dokumentů. V současné době neshrnuje tyto věci dohromady žádný RFC dokument ani norma. V minulých dobách byl systém DNS definován vlastně jako „to, co implementuje program BIND“. Tato definice zastarala i tím, že vznikly i jiné DNS servery, než právě BIND.

1.2 Hierarchie domén

Za jednotku hierarchie jsou považovány domény. Do každé takové domény spadá skupina počítačů, které nějakým způsobem spolu souvisí a doména může obsahovat další „poddomény“. Těm se říká doména prvního, druhého, třetího řádu atd. (například doména .cz může obsahovat doménu druhé úrovně bata.cz apod.). Na samotném vrcholu hierarchie domén je takzvaná kořenová doména, která je označovaná znakem “.”. Celé jméno systému je složeno jednak z vlastního jména počítače (třeba „miroslav“) a jména domény (třeba „bata.cz“), všechno dohromady se to nazývá plně kvalifikované doménové jméno, je to podle anglického výrazu FQDN (Fully Qualified Domain Name) a jako příklad FQDN můžeme uvést např. „miroslav.bata.cz“.



Obr. 1.Hierarchie domén

1.3 Princip DNS

Pokud chceme systém DNS používat, tak potřebujeme nějakou službu, která bude mít na starosti převod jmen na číselné adresy a naopak. Služby DNS jsou poskytovány rozsáhlou hierarchií tzv. nameserverů (jmenných serverů). Na portu 53 tyto nameservery provozují službu DNS, tento port je pro tuto službu vyhrazen (seznam s vyhrazenými čísly portů

spolu se seznamem síťových služeb je definován v souboru /etc/services). Funguje to tak, že každá doména má určený určitý počet nameserverů, které jsou autoritativní pro tuto doménu – to znamená, že v daný okamžik podávají zaručeně platné a aktuální informace o strojích příslušejících k jejich doméně jen tyto autoritativní servery. Jestli chceme zjistit IP adresu nějakého počítače, tak se dotážeme lokálního nameserveru, který naši žádost převezme a zeptá se příslušného autoritativního serveru. Systém DNS vytváří hierarchii domén a chceme-li například zjistit, jakou IP adresu má počítač miroslav.bata.cz, tak se musíme ptát postupně od shora. To znamená, že první se dotážeme kořenového nameserveru, který nás odkáže na nějaký z nameserverů domény .cz (a současně se jmény nám poskytne IP adresy těchto nameserverů). Zeptáme se tedy nameserveru pro doménu .cz a ten nás zase odkáže na nameserever domény bata.cz (například se jedná o stroj hlavni.bata.cz a opět dostaneme i jeho IP adresu). Poté už se dotážeme přímo hlavního serveru domény bata.cz, stroje hlavni.bata.cz, který doménu bata.cz spravuje a obdržíme od něj konečně IP adresu počítače miroslav.bata.cz. Abychom jsme se nemuseli pokaždé zdlouhavě ptát na autoritativní nameservery jednotlivých domén (je to značně neefektivní a pomalé), mohou si nameservery udržovat dotazy v paměti (cache), které již byly zpracovány a ušetřit tak často opakované dotazy.

V rámci každé domény často kvůli robustnosti existuje více než jen jeden autoritativní nameserver. V takovém případě je jeden z nich primární server a ostatní fungují jako sekundární servery, které se synchronizují vůči primárnímu nameserveru. Prakticky se taky často setkáváme s tzv. caching-only nameservery. Tyto caching-only servery nejsou autoritativní pro žádnou z domén a jsou pouze k urychlení služby DNS v rámci lokální sítě.

1.4 Typy DNS záznamů a jejich životnost

Autoritativní servery jednotlivých domén mají na starosti všechny informace o dané doméně a tyto informace jsou na nameserveru uloženy v zónových souborech. V Unixových systémech se používá pro provoz nameserveru software BIND (jež obsahuje démon named).

Typy:

SOA (*Start Of Authority*) – v tomto záznamu jsou obsaženy administrativní informace o dané doméně.

A – tento záznam svazuje IP adresu s tzv. kanonickým jménem počítače a ke každé IP adrese může být přiřazeno pouze jedno kanonické jméno. Dané IP adrese můžeme samozřejmě přidělit i další jména, ale už jen pouze jako jmenné aliasy.

CNAME (*Canonical NAME*) – pomocí těchto záznamů definujeme jmenné aliasy. Například počítač s kanonickým jménem miroslav.bata.cz bude dostupný i pod aliasem mirek.bata.cz.

PTR – tyto záznamy slouží při reverzním mapování IP adres na jména (máme-li tedy IP adresu, tak se můžeme dotázat na jméno počítače).

NS (*Name Server*) – tento záznam definuje autoritativní nameservery pro danou doménu.

MX (*Mail eXchanger*) – tyto záznamy definují poštovní servery pro danou doménu.

TTL (Time To Live) parametr je také důležitý, definuje životnost jednotlivýh záznamů. Nebylo by možné bez toho např. uchovávat DNS záznamy v cache nameservrech – nameservery musí po uplynutí doby TTL záznamu z cache paměti vymazat, v případě potřeby si mohou záznam vyžádat znovu v aktuální podobě.

1.5 Nástroje pro práci s DNS

V unixových a tedy i v linuxových distribucích nevyjímaje jsou obsaženy nástroje (klienty) pro práci se systémem DNS. Jsou to obvykle nástroje pro příkazovou řádku. Často se můžeme setkat s nástrojem nslookup (od jeho používání se začíná pomalu upouštět protože je to už starší nástroj, nicméně ho ve většině systému najdeme) a dále s nástroji host a dig. Všechny tyto nástroje jsou součástí balíčku bind-utils.

1.5.1 Nslookup

Nástroj nslookup používáme buď jednorázově nebo v interaktivním režimu. Pokud potřebujeme zjistit adresu stroje www.bata.cz, tak zadáme do příkazové řádky jednoduše:

```
nslookup www.bata.cz
```

1.5.2 Host

Nástroj host je jednoduchý, ale efektivní nástroj, jehož použití je přímočaré. Pro zjištění IP adresy servery např. www.bata.cz , musím do příkazové řádky zadat:

```
host www.bata.cz
```

1.5.3 Dig

Nástroj dig je velmi flexibilní, lze ho spouštět i v dávkovém režimu (celé úlohy a příkazy jsou předem seřazeny do určité posloupnosti, v níž se provádějí bez zásahu uživatele). Potřebujeme-li zjistit IP adresu serveru např. www.bata.cz ,tak zadáme do příkazové řádky:

```
dig www.bata.cz
```

2 BIND

Systém BIND (Berkely Internet Name Domain) vytvořila organizace ISC a je to balíček svobodného softwaru, implementující protokol DNS a poskytující jmenné služby v operačních systémech Unix, Linux, Windows a MacOS.

Programu BIND existují tři verze: BIND 4, BIND 8 a BIND 9. BIND 4 vznikl koncem osmdesátých let minulého století. BIND 8 byl představen v roce 1997 a BIND 9 spatřil světlo světa v polovině roku 2000. Verze BINDu 5, 6 ani 7 neexistují. BIND 8 byl představen se systémem 4.4 BSD, pro nějž byla všechna čísla verzí zvýšena na 8. Také program sendmail přeskočil ve stejném okamžiku několik čísel a přešel rovnou na verzi 8.

V programu BIND 8 bylo obsaženo velké množství technických novinek, např. zvýšení výkonu, zlepšení zabezpečení a robustnosti. Program BIND 9 posunul hranice ještě o něco výš, protože začal podporovat i více procesorů, začal fungovat bezpečně s vlákny a měl skutečné zabezpečení založené na veřejných klíčích, dále podporoval protokol IPv6, inkrementální doménové přenosy a množství dalších užitečných funkcí. V programu BIND 9 byla použita nová datová struktura nazvaná červeno-černý strom ukládající zónová data v paměti. Program BIND 9 je také kompletním přepracováním s reimplementací. Odděluje ty části zdrojového kódu, které jsou pro daný operační systém specifické, což usnadnilo převod programu BIND i na neunixové systémy. Vnitřnosti programu BIND se výrazně změnily, kdežto postup konfigurace zůstal identický.

2.1 Součásti programu BIND

Systém BIND je složen ze tří součástí. První z nich je démon nazvaný named. Ten odpovídá na dotazy. Druhou součástí jsou knihovní rutiny, které mají na starosti hostitelské dotazy kontaktováním serverů distribuované databáze. Třetí a poslední součást je příkazové rozhraní pro systém DNS a jsou to nástroje nslookup, host a dig.

2.1.1 Jmenný server BIND – named

Úkolem programu named je dotazování se na hostitelská jména a IP adresy. Jestliže program named neví odpověď na dotaz, tak se zeptá jiných serverů a jejich odpovědi uchovává ve vyrovnávací paměti (je to paměť typu „cache“). Program named má také na starosti „zónové přenosy“ za účelem kopírování dat mezi servery dané domény. „Zóna“ je

vlastně doména bez subdomén. Nameservery zpracovávají „zóny“, i když se často říká „doména“ , tak se myslí „zóna“.

3 DYNAMICKÉ AKTUALIZACE ZÓNOVÝCH SOUBORŮ

Systém DNS je postaven na předpokladu, že mapování jmen na adresy se nemění příliš často a je relativně stabilní. Některé organizace používající protokol DHCP k dynamickému přidělování adres počítačům během jejich spouštění a připojování k síti tento předpoklad neustále porušují. Proto existují dvě klasická řešení, jak tomu předejít. Prvním řešením je přidáním do databáze DNS obecné položky. Druhým řešením je pak neustálé upravování DNS souborů, které je příliš časově náročné. Proto ani jedno z těchto řešení není pro mnoho organizací uspokojivé.

Ukázka prvního řešení, kdy se DNS server nakonfiguruje následovně:

```
dhcp-hostitel1.domena    IN   A    192.168.0.1
dhcp-hostitel2.domena    IN   A    192.168.0.2
...
```

Toto řešení je sice jednoduché, ale znamená trvalé spojení konkrétních IP adres s hostitelskými jmény a počítače tak mění svá hostitelská jména vždy když obdrží novou IP adresu. V tomto prostředí jsou logování a bezpečnostní opatření založená na hostitelských jménech velmi obtížné.

U novějších verzí programu BIND funkce dynamické aktualizace poskytuje alternativní řešení. Doméně DHCP umožňuje upozorňovat program BIND na přidělené adresy a aktualizovat tak obsah databáze DNS za chodu. K dispozici je i rozhraní pro příkazový řádkový procesor sloužící k ručnímu provádění dynamických aktualizací.

Dynamické aktualizace jsou schopny přidávat, odstraňovat nebo upravovat zdrojové záznamy v databázi DNS. Pozice, na níž jsou dynamické aktualizace regulovány, je zóna. Povolení dynamických aktualizací celé databázi DNS vypadá trochu nebezpečně, a proto některé organizace vytvářejí subdomény (např. dhcp.bata) a dynamické aktualizace povolují jen v této subdoméně.

Jakmile se zóna dynamicky aktualizuje, tak ji nemůžeme upravit ručně, dokud nezastavíme program BIND příkazem `ndc stop` (u programu BIND 9 musíme použít příkaz `rndc stop`), aby se na disk zapsala aktuální kopie databáze. Poté můžeme již upravovat zónový soubor ručně. Samozřejmě se zničí původní formátování zónového souboru (soubor bude vypadat jako soubory spravované démonem named pro podřízené servery).

Během dynamických aktualizací se udržuje logovací soubor „jméno-zóny.jnl“ pro případ že by server havaroval. Tento soubor je nutné po zastavení démona named smazat a před tím, než začneme upravovat ručně zónový soubor. V nejlepším případě je ale nejrozumnější se do dynamických zón DNS nemíchat, protože mohou vznikat nekonzistentní údaje.

3.1 Nsupdate

Nástroj nsupdate, který je součástí programu BIND 9, poskytuje dynamickým aktualizacím rozhraní pro příkazový řádek. Operuje v dávkovém režimu a přijímá příkazy ze souboru nebo z klávesnice. Konec vstupu se dává najevo dvěma prázdnými řádky. Příkazový jazyk obsahuje jednoduchý příkaz „if“ pro vyjádření konstrukcí jako je např. „jestliže toto hostitelské jméno v systému DNS neexistuje, tak přidej počítač“. Jako predikáty pro akci nsupdate můžeme požadovat, aby jméno buď neexistovalo nebo existovalo, nebo vyžadovat neexistenci nebo existenci zdrojového záznamu.

Pro příklad uvedu jednoduchý skript nsupdate, který provede přidání počítače:

```
%nsupdate
>update add novyhostitel.bata  86400  A  192.168.1.100
>send
```

Dynamické aktualizování je hrůzostrašné, protože potenciálně může poskytovat neřízený přístup k zápisu do důležitých systémových údajů. Proto pro řízení přístupu nebudeme používat IP adresy, protože je lze velice snadno zfalšovat, ale budeme používat autentizaci TSIG s klíčem ve formě sdíleného tajemství, protože je dostupná a snadno se nastavuje. Nyní si uvedeme jeden příklad jak použít TSIG:

```
%nsupdate –y název-klíče:tajný-klíč
```

Dynamické aktualizace zóny se povolují větou `allow-update` nebo `update-policy` v souboru named.conf v adresáři /etc/bind. Proměnná `allow-update` poskytuje povolení k aktualizaci jakýchkoliv záznamů v souladu se zdrojovou IP adresou nebo autentizaci založenou na klíčích. `Update-policy` je rozšíření programu BIND 9, které umožňuje jemné řízení aktualizací podle typu záznamu nebo jména počítače. Je potřeba autentizace založená na klíčích.

Jestliže chceme klientům povolit aktualizaci jejich záznamů PTR nebo A, ale chceme jim zakázat změnit jejich záznam NS, SOA nebo KEY, tak použijeme větu `update-policy`. Tuto větu můžeme použít i k tomu, když počítači povolíme jen aktualizovat jeho vlastní záznamy. Parametry umožňují vyjádřit jména explicitně, jako subdoménu, jako žolíka, nebo jako klíčové slovo self, kterým se nastavuje obecná politika pro přístup počítačů ke svým vlastním záznamům. Zdrojové záznamy se identifikují podle typu a třídy. Uvedu příklad:

```
update-policy {grant dhcp-key subdomain dhcp.bata.cz A};
```

Touto konfigurací dovolíme všem, kdo znají klíč dhcp-key, aby mohl aktualizovat adresní záznamy v subdoméně dhcp.bata.cz. Tento příkaz se musí vyskytovat v souboru named.conf v adresáři /etc/bind pod příkazem zone pro doménu dhcp.bata.cz. Také by tady musel být příkaz key pro definici dhcp-key.

3.2 Bezpečná komunikace mezi servery pomocí TSIG

Organizací IEFT bylo vyvinuto pro umožnění bezpečné komunikace mezi servery pomocí transakčních podpisů mechanismus TSIG (dokument RFC 2845). Řízení přístupu založené na transakčních podpisech je bezpečnější než řízení přístupu založené na zdrojových IP adresách. Transakční podpisy autentizují příjemce/odesilatele a ověřují, jestli údaje nebyly pozměněny. Používají se typicky pro běžné zónové přenosy mezi hlavním serverem a jeho podřízenými servery a pro zabezpečení dynamického aktualizování.

U transakčních podpisů je použita metoda symetrického šifrování. To znamená, že dešifrovací klíč je stejný jako šifrovací. Tento jeden klíč se nazývá sdílené tajemství. Pro každou dvojici serverů, které mají bezpečně komunikovat, by měl být použit jiný pár klíčů. Metoda TSIG je daleko méně výpočetně náročná než kryptografie s veřejným klíčem, ale je vhodná pro lokální sítě, na kterých je jen málo serverových páru, které spolu mají bezpečně komunikovat.

Podpisy TSIG podepisují dotazy DNS a i odpovědi na ty dotazy. Podpisy se používají jen mezi servery, nikoliv mezi servery a resolvery. Podpisy TSIG jsou kontrolovány v době, kdy se diagram přijímá a posléze se zahazují. Nestávají se součástí údajů DNS, ani se neukládají do paměti cache. Specifikace TSIG umožňuje více šifrovacích metod, ale program BIND implementuje pouze jednu z nich a tou je algoritmus HMAC-MD5.

4 POSTGRESQL

PostgreSQL je relační databázový systém s otevřeným zdrojovým kódem. Je již patnáct let vyvíjen a zakládá si na bezpečnosti a spolehlivosti. Je distribuován pod licencí BSD (Berkeley Software Distribution), která umožňuje volné spojování otevřeného kódu s uzavřeným. Je velmi často srovnáván s další velmi rozšířenou otevřenou databází MySQL.

PostgreSQL funguje pod většinou rozšířených systémů jako je Linux nebo Windows. Splňuje podmínky ACID, podporuje plně cizí klíče, operace JOIN, spouště, uložené procedury a pohledy. Jsou v něm obsaženy SQL92 a SQL99 datové typy.

Umožňuje běh uložených procedur napsaných v několika rozšířených programovacích jazycích – v Perlu, Pythonu,C nebo ve speciálním PL/pgSQL (jazyku vycházejícím z PL/SQL firmy Oracle).

Hlavní předností systému PostgreSQL je jeho rozšiřitelnost. Systém lze rozšířit o nové funkce, operátory, datové typy, agregační funkce, procedurální jazyky.

Nevýhodou, zejména pro jednotlivce a menší firmy, je v porovnání hlavně s MySQL malá rozšířenost PostgreSQL systému na hostingových serverech a menší komunita, která je potřebná pro pomoc s případnými problémy.

4.1 Historie PostgreSQL

Systém Ingres (Interactive Graphics and Retrieval System) byl předchůdcem PostgreSQL, byl vyvinutý na univerzitě v Berkeley v letech 1977 – 1985. Pod vedením Prof. Michaela Stonebrakera byl vyvíjen jeho nástupce jako objektově-relační databázový server pod názvem Postgres. Později z toho firma Illustra vytvořila komerční produkt, který byl převzat později Informixem. Původní Postgres byl doplněn o podporu jazyka SQL a byl označován jako Postgres95. V roce 1996 byl sestaven tým lidí, kteří pokračovali na vývoji Postgres95 už nezávisle na univerzitě a jako open source. Systém byl postupně opravován a vylepšován a po odstranění nejhorších problému se mohlo začít na implementaci nových vlastností. Projekt byl přejmenován na PostgreSQL na konci roku 1996 a aktuální produkční verze je 8.3.

II. PRAKTICKÁ ČÁST

5 REPLIKACE DAT S POSTGRESQL DO DNS

Práce je zaměřena na replikaci dat z relační databáze PostgreSQL do DNS serveru BIND na Linuxové distribuci Debian. Data potřebná pro aktualizaci DNS serveru byla uložena ve dvou tabulkách databáze, první s názvem deviceplace a druhá networks. V tabulce deviceplace byly uloženy informace o pobočkách firmy Baťa v České Republice, Slovensku a Polsku (konkrétně šlo o doménové jména, které byly pro mě důležité pro replikaci do DNS). Tabulka network obsahovala informace o jednotlivých zařízeních, které byly na pobočkách (jednalo se o IP adresy zařízení a jejich názvy). Tabulky bylo potřeba sloučit pomocí příkazu JOIN s pomocí dat, která byla obsažena v obou tabulkách. Jednalo se o sloupce v nichž byly uloženy hodnoty identifikační údaje o sítích. „Select“ pro výběr důležitých dat vypadal následovně:

```
SELECT n.domainname,d.address,d.hostname FROM networks n JOIN deviceplaces d ON
n.idnetwork=d.idnetwork WHERE n.domainname IS NOT NULL AND d.address IS NOT NULL AND
d.hostname IS NOT NULL
```

Obr. 2. SQL dotaz výběru dat pro replikaci

Poté bylo potřeba implementovat data pomocí Perlu, aby byla použitelná pro DNS server BIND. Vytvořil jsem skript pro replikaci. Musely být použity moduly DBI a DBD. Modul DBD je driver pro připojení k databázi. Konkrétně jde o modul DBD::Pg, kterým lze nastavit připojení k databázi PostgreSQL. Nejprve je potřeba se připojit k databázi viz Obr.3 a na konci skriptu ukončit připojení viz Obr.4.

```
my $dbh = DBI->connect("dbi:Pg:dbname=$dbname",$user, $password,
                                { RaiseError => 1, AutoCommit => 0 });
```

Obr. 3. DBD driver – připojení

```
$dbh->disconnect;
```

Obr. 4. DBD driver – odpojení

Jakmile je databáze připojena, tak se s ní může začít pracovat. K tomu jsem použil modul DBI, který obsahuje potřebné funkce pro replikaci. Pomocí příkazu prepare a exekute jsem vybral data viz Obr.5.

```
my $sth = $dbh->prepare("SELECT n.domainname,d.address,d.hostname FROM networks n JOIN
deviceplaces d ON n.idnetwork=d.idnetwork WHERE n.domainname IS NOT NULL AND d.address IS
NOT NULL AND d.hostname IS NOT NULL");
   $sth->execute();
        while ( @row = $sth->fetchrow_array ) {
                print "Domena\t$row[0]\n";
                print "adresa\t$row[1]\n";
                print "hostname\t$row[2]\n";
```

Obr. 5. DBI – výběr dat

Jakmile jsem měl data z databáze přístupné, tak jsem mohl začít s vkládáním dat do DNS serveru. K tomu jsem využil modulu Net::DNS::Update, který obsahuje funkci pro vkládání záznamů do DNS serveru viz Obr.6.

```
            # Create the update packet.
            my $update = Net::DNS::Update->new("$row[0]");

            # Add A record for the name.
            $update->push("update" => rr_add("$row[2].$row[0]. 86400 A $row[1]"));

            $update->sign_tsig($key_name, $key);

            # Send the update to the zone's primary master.
            my $res = Net::DNS::Resolver->new;
            $res->nameservers("primary-master.$row[0]");

            my $reply = $res->send($update);

            # Did it work?
            if ($reply) {
                if ($reply->header->rcode eq 'NOERROR') {
                    print "Update succeeded\n";
                } else {
                    print "Update failed: ", $reply->header->rcode, "\n";
                }
            } else {
                    print "Update failed: ", $res->errorstring, "\n";
            }
  }
```

Obr. 6. Vkládání dat do DNS

Dále by jsem chtěl popsat část kódu, který se zabývá zabezpečením přístupu do DNS serveru. Je použit rndc-key, který je definován na začátku skriptu. Na Obr.7 je kód, který se stará o bezpečnost přístupu k DNS serveru.

```
$update->sign_tsig($key_name, $key);
```

Obr. 7. Zabezpečení přístupu do DNS

Nakonec celý skript pro replikaci dat z PostgreSQL do DNS serveru je na Obr.8

```
#!/usr/bin/perl -w

use DBI;
use strict;
use warnings;
use utf8;
use Net::DNS::Update;
use Net::DNS;

my $dbname    ="dns";
my $user      ="";
my $password  ="";
my @row       ="";
my $hostname  ="";
my $domainname="";
my $address   ="";
my $key_name  = "rndc-key";
my $key       = "plkxpqI8Pw7KcPW+wFmgtQ==";
#Net::DNS version
#print Net::DNS->version, "\n";

#man DBD::Pg
#$dbh = DBI->connect($data_source, $username, $password)
my $dbh = DBI->connect("dbi:Pg:dbname=$dbname",$user, $password,
                        { RaiseError => 1, AutoCommit => 0 });

my $sth = $dbh->prepare("SELECT n.domainname,d.address,d.hostname FROM networks n JOIN deviceplaces d ON n.idnetwork=d.idnetwork WHERE n.domainname IS NOT
NULL AND d.address IS NOT NULL AND d.hostname IS NOT NULL");
   $sth->execute();
        while ( @row = $sth->fetchrow_array ) {
                print "Domena\t$row[0]\n";
                print "Adresa\t$row[1]\n";
                print "Hostname\t$row[2]\n";

                # Create the update packet.
                my $update = Net::DNS::Update->new("$row[0]");

                # Add A record for the name.
                $update->push("update" => rr_add("$row[2].$row[0]. 86400 A $row[1]"));

                $update->sign_tsig($key_name, $key);

                # Send the update to the zone's primary master.
                my $res = Net::DNS::Resolver->new;
                $res->nameservers("primary-master.$row[0]");

                my $reply = $res->send($update);

                # Did it work?
                if ($reply) {
                    if ($reply->header->rcode eq 'NOERROR') {
                        print "Update succeeded\n";
                    } else {
                        print "Update failed: ", $reply->header->rcode, "\n";
                    }
                } else {
                        print "Update failed: ", $res->errorstring, "\n";
                }
        }

$dbh->disconnect;
#__END__
```

Obr. 8. Skript pro replikaci dat

6 KONFIGURACE DNS SERVERU BIND

DNS server BIND se konfiguruje pomocí souborů, které se nacházejí v adresáři /etc/ a /etc/bind/. Jedná se o soubory resolv.conf (resolver) a named.conf (jmenný server)

6.1 Konfigurace resolveru

Nejprve se pustím do nastavení resolveru, který se nachází v adresáři /etc/. Resolver implementuje funkce pro převod doménových jmen na IP adresy a naopak, funkce pro dotazování jmenného serveru apod. Resolver k správnému fungování potřebuje znát řadu konfiguračních údajů, které čte z konfiguračních souborů umístěných v adresáři /etc (jde o soubory hosts, nsswitch.conf a host.conf).

Soubor hosts obsahuje statický seznam doménových jmen a IP adres. Soubory nsswitch.conf a host.conf říkají jaký se použije způsob převodu jmen na IP adresy (jestli statický záznam nebo služba DNS, popřípadě obojí a jestliže se použije obojí, tak se ještě určuje pořadí).

Soubor resolv.conf obsahuje seznam jmenných serverů služby DNS

```
domain bata.cz
search bata.cz
nameserver 192.168.1.1
nameserver 192.168.1.101
```

Obr. 9. Soubor resolv.conf

6.2 Konfigurace jmenného serveru

Jmenný server slouží pouze k urychlení služby DNS v rámci lokální sítě. Hlavní konfigurační soubor je named.conf v adresáři /etc, ve kterém se uvádí, pro které zóny je jmenný server autoritativní a zde se také definují přístupová práva (kdo může k jmennému serveru přistupovat atd.).

```
include "/etc/bind/named.conf.options";

include "/etc/bind/rndc.key";

controls {
  inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "rndc-key"; };
};


// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

// Informace o DNS ns.doma
zone "bata.cz" {                                // konfigurační soubor named.doma
        type master;                    // (pojmenovat si ho můžete jakkoliv)
        file "/etc/bind/db.bata.cz";    // je uložen v cestě uvedené v directory
        allow-update { key rndc-key; };
};                                      // v tomto případě /var/named/named.doma

// Slouží ke zpětnému překladu. (IP -> jméno)
zone "1.168.192.in-addr.arpa" {         //toto jméno je složeno z IP adresy třídy C (192.168.1) a .in-addr.arpa
        type master;
        file "/etc/bind/db.rev";        // opět konfigurační soubor
};

include "/etc/bind/named.conf.local";
```

Obr. 10. Soubor named.conf

```
options {
        directory "/var/cache/bind";
                                                #If there is a firewall between you and nameservers you want
                                                #to talk to, you may need to fix the firewall to allow multiple
                                                #ports to talk.  See http://www.kb.cert.org/vuls/id/800113

                                                #If your ISP provided one or more IP addresses for stable
                                                #nameservers, you probably want to use them as forwarders.
                                                #Uncomment the following block, and insert the addresses replacing
                                                #the all-0's placeholder.

        auth-nxdomain no;    # conform to RFC1035

        query-source port 53;
                                                #Pro komunikaci s jinou DNS použij port 53

        forward first;
                                                #Pokud něco neví, ptá se nejprve serverů
                                                #uvedených ve forwarders

        forwarders {
        192.168.1.1;
        192.168.1.101;
        };

        cleaning-interval 40320;
                                                #Pročišťuje tabulku pomocného serveru od záznamů,
                                                #na které se nikdo neptal (v sekundách)

        listen-on-v6 { any; };
};
```

Obr. 11. Soubor named.conf.options

Když chceme provozovat caching-only server, neobejdeme se zcela bez zónových souborů. Zónové soubory jsou umístěny v adresáři /etc/bind.

```
$ORIGIN .
$TTL 86400      ; 1 day
bata.cz                 IN SOA  ns.bata.cz. root.bata.cz. (
                                2009042302 ; serial
                                10800      ; refresh (3 hours)
                                1800       ; retry (30 minutes)
                                1209600    ; expire (2 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      ns.bata.cz.
                        MX      10 mail.bata.cz.
$ORIGIN bata.cz.
$TTL 600        ; 10 minutes
tomas                   A       192.168.10.100
$TTL 86400      ; 1 day
petr            A       192.168.1.101
lukas                   A       192.168.1.100
martin                  A       192.168.1.111
$TTL 600        ; 10 minutes
miroslav                A       192.168.10.105
$TTL 86400      ; 1 day
ns                      A       127.0.0.1
pop3                    CNAME   server
server                  A       192.168.1.111
smtp                    CNAME   server
www                     CNAME   server
```

Obr. 12. Soubor db.bata.cz

Jestliže máme tyto soubory nakonfigurovány, pak to stačí pro zprovoznění caching-only jmenného serveru a pak už jen zbývá spustit službu named příkazem `/etc/init.d/named start`.

7 POSTGRESQL

Byla vytvořena databáze s názvem „dns“, do které byla importována data ze souboru SQL.

Na Obr. 9 je zobrazena databáze „dns“ a tabulky, které byly potřeba při vypracování této práce.

```
ivo@iTchy-Debian:~$ psql dns
Welcome to psql 8.3.7, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

dns=# \dt
                Seznam relací
 Schéma |    Jméno     |   Typ   | Vlastník
--------+--------------+---------+----------
 public | deviceplaces | tabulka | postgres
 public | networks     | tabulka | postgres
(2 rows)

dns=#
```

Tab. 1. Databáze – tabulky

Jednotlivé tabulky s daty jsou zobrazeny na Obr. 14 a Obr.15. Tabulky jsou vloženy jen pro ukázku, jinak obsahovaly data s několika sty řádky.

iddeviceplace	idnetwork	iddevice	address	hostname	state	statestamp	stateuser
4	12	50	192.168.56.67	pos11	3	2007-05-18 14:38:18+02	2
5	12	52	192.168.56.68	pos12	3	2007-05-18 14:38:28+02	2
6	13	54	192.168.56.3	pos11	1	2007-05-18 16:37:50+02	2
10	65	105	192.168.58.131	pos11	1	2007-06-04 10:07:12+02	1081
13	64	108	192.168.59.243	pos11	1	2007-06-04 10:14:00+02	1081
14	64	109	192.168.59.244	pos12	1	2007-06-04 10:14:00+02	1081

Tab. 2. Tabulka „deviceplaces“

idnetwork	domainname	netaddress	statestamp	public_host
1	test	1.1.1.1/32	2007-01-01 00:00:00+01	
2	test2	2.2.2.2/32	2007-01-01 00:00:00+01	
148	gdansk-galeria-baltycka.bata.pl	192.168.69.16/28	2007-10-05 16:04:34+02	bata53141.digitop.com.pl
163	doplnit	192.168.67.112/28	2007-11-30 14:16:32+01	bata53126.digitop.com.pl
160	bialystok-biala.bata.pl	192.168.69.144/28	2007-12-05 12:06:36+01	bata53144.digitop.com.pl
186	warszawa-targowek.bata.cz	192.168.66.80/28	2008-01-28 11:20:34+01	bata53107.digitop.com.pl
97	plzen-tesco.bata.cz	192.168.53.112/28	2007-06-11 13:12:54+02	90.176.67.142

Tab. 3. Tabulka „networks“

ZÁVĚR

Tato bakalářská práce měla za cíl vypracovat projekt pro firmu BAŤA, akciová společnost, který se zabýval replikací dat z relační databáze PostgreSQL do DNS serveru BIND. Informační systémy společnosti Baťa jsou postavené na platformě Unix . Tato práce se soustředila na použití systému Linux v distribuci Debian. Pro replikaci dat z databáze byl použit skriptovací jazyk Perl. Databáze obsahovala informace o pobočkách firmy Baťa v České Republice, Polsku a na Slovensku. Obsahovala také informace o jednotlivých zařízení na pobočkách (jednalo se o počítače, pokladny a jiné síťové zařízení). Potřebná data byla pomocí skriptu replikována do DNS serveru BIND. Tím, že DNS server obsahuje data o pobočkách a zařízení, se výrazně zjednodušila vzdálená správa těchto zařízení pomocí nástroje OpenSSH.

Dále je v této práci obsažen popis a konfigurace jednotlivých částí DNS serveru BIND, ať už se jedná o resolver, nebo o jmenný server. Popsán je i nástroj nsupdate, který slouží k dynamickým aktualizacím DNS serveru a jeho funkce je použita ve skriptu pro replikaci dat z databáze do DNS serveru. Na konec je popsána relační databáze PostgreSQL a práce s ní při replikaci dat pomocí Perlu a modulů DBD a DBI.

CONCLUSION

The core of this thesis was to elaborate a practical project for the BATA company, which dealt with data replication from relational database PostgreseSQL into a BIND DNS server. Vast majority of the BATA's IT systems are based on the Unix platform, while some parts run on GNU/Linux and therefore this work was focused on the Linux system, specifically the Debian distribution. We used the Perl scripting language for the replication scripts. The database contained public data about the BATA branches in the Czech Republic, Poland and Slovakia and also information about individual equipment on particular local offices (computers, checkout tills and other network equipment). Required data was replicated from source database into the BIND DNS server. The fact that the DNS server contains data about the branches' equipment significantly simplified remote administration of these devices using the OpenSSH tool.

In the second section, this work thoroughly describes configuration of individual parts of DNS server BIND – the resolver and the nameserver. It continues with description of the nsupdate tool, which is used for dynamic actualisation of the DNS server. Its functionality is required in the actual developed script for data replication.

Finally, we conclude with description of the mechanism involving the relational PostgreSQL database and its role in the data replication task with support of the Perl language, DBD and the DBI API modules.

SEZNAM POUŽITÉ LITERATURY

Monografie:

[1] NEMETH E., GARTH S., TRENT R. H. – „Linux – kompletní příručka administrátora“, Computer Press, Brno (2004), 828s., ISBN 80-7226-919-4

[2] SATRAPA P. – „Perl pro zelenáče“, Neocortex, Praha (2001), 224s., ISBN 80-86330-02-8

[3] PROKOPOVÁ Z. – „Databázové systémy MySQL + PHP“, Univerzita Tomáše Bati ve Zlíně, Zlín (2006), 126s., ISBN 80-7318-486-9

Internetové odkazy :

[4] LINUXZONE.CZ – server o Linuxu pro programátory, administrátory a fanoušky, [online], [cit. 2009-05-10]. Dostupné z WWW: http://www.linuxzone.cz/index.phtml?ids=9&idc=427

[5] CS.WIKIPEDIA.ORG – Wikipedie, otevřená encyklopedie, [online], [cit. 2009-05-10]. Dostupné z WWW: http://cs.wikipedia.org/wiki/PostgreSQL

[6] CPAN.ORG – Comprehensive Perl Archive Network, [online], [cit. 2009-05-10]. Dostupné z WWW: http://cpan.org

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

DNS	Systém doménových jmen.
BIND	Software pro DNS server.
DBI	Rozhraní Perlu pro databáze.
DBD	Ovladač pro databázové rozhraní
IP	Adresa, která jednoznačně identifikuje počítač v síti.
RFC	Normy popisující internetové protokoly.
FQDN	Plně kvalifikované doménové jméno.
TTL	Definuje životnost jednotlivých záznamů v DNS serveru.
ISC	Organizace, která se podílela na vytváření DNS serveru BIND.
BSD	Unixová distribuce.
DHCP	Aplikační protokol pro automatické přidělování IP adres.
TSIG	Autentizace chránící záznamy na DNS serveru.
IEFT	Organizace podílející se na vývoji bezpečné komunikace mezi servery.
HMAC-MD5	Šifrovací algoritmus.
ACID	Databázová norma.
OpenSSH	Unixový nástroj pro vzdálenou správu počítačů přes síť.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: SKRIPT PRO REPLIKACI DAT Z PSQL DO DNS

```
#!/usr/bin/perl -w

use DBI;
use strict;
use warnings;
use utf8;
use Net::DNS::Update;
use Net::DNS;

my $dbname    ="dns";
my $user      ="";
my $password  ="";
my @row       ="";
my $hostname  ="";
my $domainname="";
my $address   ="";
my $key_name  = "rndc-key";
my $key       = "plkxpqI8Pw7KcPW+wFmgtQ==";
#Net::DNS version
#print Net::DNS->version, "\n";

#man DBD::Pg
#$dbh = DBI->connect($data_source, $username, $password)
my $dbh = DBI->connect("dbi:Pg:dbname=$dbname",$user, $password,
                           { RaiseError => 1, AutoCommit => 0 });

my $sth = $dbh->prepare("SELECT n.domainname,d.address,d.hostname FROM
networks n JOIN deviceplaces d ON n.idnetwork=d.idnetwork WHERE
n.domainname IS NOT NULL AND d.address IS NOT NULL AND d.hostname IS NOT
NULL");
   $sth->execute();
      while ( @row = $sth->fetchrow_array ) {
            print "Domena\t$row[0]\n";
            print "Adresa\t$row[1]\n";
            print "Hostname\t$row[2]\n";

            # Create the update packet.
            my $update = Net::DNS::Update->new("$row[0]");

            # Prerequisite is that no A records exist for the name.
            $update->push("pre" => nxrrset("$row[2].$row[0]. A"));


            # Add A record for the name.
            $update->push("update" => rr_add("$row[2].$row[0]. 86400 A
$row[1]"));

            $update->sign_tsig($key_name, $key);

            # Send the update to the zone's primary master.
            my $res = Net::DNS::Resolver->new;
            $res->nameservers("primary-master.$row[0]");

            my $reply = $res->send($update);

            # Did it work?
            if ($reply) {
                if ($reply->header->rcode eq 'NOERROR') {
                    print "Update succeeded\n";
                } else {
                    print "Update failed: ", $reply->header->rcode, "\n";
```

```
                }
            } else {
                print "Update failed: ", $res->errorstring, "\n";
            }
        }

$dbh->disconnect;
#__END__
```

PŘÍLOHA P II: RESOLV.CONF

```
domain bata.cz
search bata.cz
nameserver 192.168.1.1
nameserver 192.168.1.101
```

PŘÍLOHA P III :NAMED.CONF

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on
the
// structure of BIND configuration files in Debian, *BEFORE* you
customize
// this configuration file.
//
// If you are just adding zones, please do that in
/etc/bind/named.conf.local

include "/etc/bind/named.conf.options";

include "/etc/bind/rndc.key";

controls {
  inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "rndc-key"; };
};


// prime the server with knowledge of the root servers
zone "." {
      type hint;
      file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
      type master;
      file "/etc/bind/db.local";
};

// Informace o DNS ns.doma
zone "bata.cz" {                    // konfigurační soubor named.doma
      type master;                  // (pojmenovat si ho můžete jakkoliv)
      file "/etc/bind/db.bata.cz";  // je uložen v cestě uvedené v
                                    //directory
      allow-update { key rndc-key; };
};                                  // v tomto případě
/var/named/named.doma

// Slouží ke zpětnému překladu. (IP -> jméno)
zone "1.168.192.in-addr.arpa" {     //toto jméno je složeno z IP
                                    //adresy třídy C (192.168.1) a .in-
                                    //addr.arpa
      type master;
      file "/etc/bind/db.rev";      // opět konfigurační soubor
};

include "/etc/bind/named.conf.local";
```

PŘÍLOHA P IV: ZÓNOVÝ SOUBOR DB.BATA.CZ

```
$ORIGIN .
$TTL 86400  ; 1 day
bata.cz                 IN SOA      ns.bata.cz. root.bata.cz. (
                        2009042302 ; serial
                        10800      ; refresh (3 hours)
                        1800       ; retry (30 minutes)
                        1209600    ; expire (2 weeks)
                        604800     ; minimum (1 week)
                        )
                    NS    ns.bata.cz.
                    MX    10 mail.bata.cz.
$ORIGIN bata.cz.
$TTL 600    ; 10 minutes
tomas               A     192.168.10.100
$TTL 86400  ; 1 day
petr        A       192.168.1.101
lukas               A     192.168.1.100
martin                    A     192.168.1.111
$TTL 600    ; 10 minutes
miroslav            A     192.168.10.105
$TTL 86400  ; 1 day
ns                  A     127.0.0.1
pop3                CNAME server
server                    A     192.168.1.111
smtp                CNAME server
www                 CNAME server
```

PŘÍLOHA P V: NAMED.CONF.OPTIONS

```
options {
      directory "/var/cache/bind";

      // If there is a firewall between you and nameservers you want
      // to talk to, you may need to fix the firewall to allow multiple
      // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

      // If your ISP provided one or more IP addresses for stable
      // nameservers, you probably want to use them as forwarders.
      // Uncomment the following block, and insert the addresses
      // replacing
      // the all-0's placeholder.

      // forwarders {
      //    0.0.0.0;
      // };

      auth-nxdomain no;    # conform to RFC1035

      query-source port 53;
      // Pro komunikaci s jinou DNS použij port 53

      forward first;
      // Pokud něco neví, ptá se nejprve serverů
      // uvedených ve forwarders

      forwarders {
      192.168.1.1;
      192.168.1.101;
      };

      cleaning-interval 40320;
      // Pročišťuje tabulku pomocného serveru od záznamů,
      // na který se nikdo neptal (v sekundách)

      listen-on-v6 { any; };
};
```

PŘÍLOHA PVI: TABULKA DEVICEPLACES

iddeviceplace	idnetwork	iddevice	address	hostname	state	statestamp	stateuser
4	12	50	192.168.56.67	pos11	3	2007-05-18 14:38:18+02	2
5	12	52	192.168.56.68	pos12	3	2007-05-18 14:38:28+02	2
6	13	54	192.168.56.3	pos11	1	2007-05-18 16:37:50+02	2
10	65	105	192.168.58.131	pos11	1	2007-06-04 10:07:12+02	1081
13	64	108	192.168.59.243	pos11	1	2007-06-04 10:14:00+02	1081
14	64	109	192.168.59.244	pos12	1	2007-06-04 10:14:00+02	1081

PŘÍLOHA PVII: TABULKA NETWORKS

idnetwork	domainname	netaddress	statestamp	public_host
1	test	1.1.1.1/32	2007-01-01	
2	test2	2.2.2.2/32	2007-01-01	
148	gdansk-galeria-baltycka.bata.pl	192.168.69.16/28	2007-10-05	bata53141.digitop.com.pl
163	doplnit	192.168.67.112/28	2007-11-30	bata53126.digitop.com.pl
160	bialystok-biala.bata.pl	192.168.69.144/28	2007-12-05	bata53144.digitop.com.pl
186	warszawa-targowek.bata.cz	192.168.66.80/28	2008-01-28	bata53107.digitop.com.pl
97	plzen-tesco.bata.cz	192.168.53.112/28	2007-06-11	90.176.67.142