NTP, DNS, DNSSEC

ISA - Laboratorní cvičení č.3

Vysoké učení technické v Brně

`https://github.com/nesfit/ISA/tree/master/dns`

Cíle laboratoře

- Základní seznámení se synchronizací času a protokolem NTP.
- Systém DNS.
- Zabezpečení DNSSEC.

Základní instrukce

- Přihlaste se do OS Kali Linux (F3), user/password `user`/`user4lab`.
- Otevřete si příkazovou řádku pro uživatele `user`.
- Otevřete si příkazovou řádku pro uživatele `root` příkazem `su` (switch user).
- V případě potřeby si otevřete další terminál v novém okně.
- Pro editaci konfiguračních souborů použijte libovolný editor (např. nano, mcedit, vim, gedit).
- Pracujte ve dvojicích, zkontrolujte, že máte počítač propojen přímým kabelem s přístupovým přepínačem (rozhraní eth0, zdířka E na patch panelu).
- Příkazem `systemctl stop NetworkManager.service` vypněte NetworkManager spravující rozhraní eth1 (doporučené, volitelně; zadání dále předpokládá, že je NetworkManager vypnutý).

Úkoly

1 NTP

Vašim úkolem je zajistit synchronizaci hodin počítače. V rámci bezpečnosti je důležité provozovat počítač se správným časem kvůli časovým razítkům označujícím platnost a neplatnost klíčů, certifikátů, podpisů apod.

1. Zkontrolujte obsah konfiguračního souboru `/etc/ntp.conf`. Ujistěte se, že je nastavena synchronizace proti serveru `ntp.fit.vutbr.cz`.
2. Zapněte démona pro NTP (`systemctl start ntp.service`). Sledujte postup synchronizace příkazem `watch ntpq -p`. Nemusíte čekat na dokončení synchronizace, nechte okno spuštěné a pokračujte dalšími úkoly.

2 Konfigurace DNS serveru

1. Pracujte ve dvojicích. Jeden z dvojice bude konfigurovat server (PC A) a druhý počítač bude mít roli klienta (PC B). Propojte rozhraní eth1 (zdířka D na patch panelu) mezi počítači ve dvojici křížovým kabelem.

2. Nastavte statické IP adresy na rozhraní eth1 vašich počítačů v souboru `/etc/network/interfaces`:

   ```
   auto eth1
   iface eth1 inet static
       address 192.168.X.X
       netmask 255.255.255.0
   ```

   Restartujte síťování `systemctl restart networking.service`. Otestujte konektivitu příkazem `ping`. Pokud konektivita nefunguje, ověřte nastavení IP adres `ip a s` a zamyslete se, co je špatně.

3. Upravte konfigurační soubor `/etc/bind/named.conf.options` (parametr `listen-on` v sekci `options`) tak, aby PC A naslouchalo i na adrese, která mu byla přiřazena staticky na rozhraní eth1. Inspirovat se můžete již uvedeným paramterem `listen-on-v6`, případně v manuálových stránkách.

4. Na PC A rozšiřte konfiguraci DNS serveru tak, aby obsluhoval požadavky pro vámi zvolenou doménu, např. `isa.cz`. Doména bude obsahovat:

   - Nameserver pojmenovaný `server`.
   - Záznam typu A pro `server`.
   - Záznam typu A pro jméno `pcb`, který odkazuje na IP adresu počítače PC B.
   - Záznam typu CNAME pro jméno `pca` ukazující na `server`.

   Inzerujte platnost vámi zasílaných záznamů 1 hodina. Jako základ zónového souboru pro vaší doménu můžete použít soubor `/etc/bind/db.local`, či `/root/isa3/fit.vutbr.cz`. Needitujte však soubor přímo, ale zkopírujte jej do `/var/cache/bind`, nezapomeňte zaregistrovat zóny v souboru `named.conf.local` (potřebné informace jsou k dispozici v teoretické části).

5. V případě zájmu nakonfigurujte pro doménu překlad na adresy IPv6 (záznamy AAAA).

6. Nakonfigurujte reverzní překlad. Jako základ zónového souboru s reverzní zónou můžete použít např. `/etc/bind/db.127`. Zkopírujte soubor do `/var/cache/bind`, upravte jej a přidejte do souboru `named.conf.local`.

7. Spusťte DNS server příkazem `systemctl start bind9.service`. Ujistěte se, že v souboru `/var/log/syslog` nejsou žádné chyby. Případné chyby týkající se chybějící konektivity pomocí protokolu IPv6 ignorujte.

8. V souboru `/etc/dhcp/dhclient.conf` nastavte na obou počítačích použití vlastního serveru DNS běžícího na PC A, na konec souboru přidejte řádek
   `prepend domain-name-servers 192.168.X.X;`
   Na obou počítačích restartujte síťování: `systemctl restart networking.service`. Ověřte nastavení IP adres a obsah souboru `/etc/resolv.conf`.

9. Ověřte, že PC A používá pro překlad lokální DNS server (`dig localhost`). Adresu dotazovaného serveru udává položka *SERVER* v poslední části odpovědi.

10. Ověřte, že PC B používá pro překlad lokální DNS server a dostává od něj odpověď.
11. Ověřte zda lze z obou počítačů přeložit dopředné i zpětné záznamy.
12. Odchyťte si komunikaci v rámci DNS na obou počítačích programem `wireshark`. Analyzujte si jednotlivé položky a porovnejte je s výpisem programu `dig`.

3 Jednoduchý útok na DNS

1. Na PC B spusťte prohlížeč IceWeasel. Zobrazte si webové stánky v doménách `www.fit.vutbr.cz` a www.seznam.cz. Všimněte si klíče v pravé části adresního řádku.
2. Na PC A vytvořte zónové soubory pro domény `fit.vutbr.cz` a `seznam.cz`. Nastavte IP adresu počítače `www` v obou doménách na adresu PC A. Zónové soubory najdete připraveny ve složce `/root/isa3`. Zónové soubory přidejte do souboru `/etc/bind/named.conf.local` a restartujte Bind.
3. Na PC A se v příkazové řádce přesuňte do adresáře `/root/isa3`. Spusťte zde jednoduchý webový server příkazem `python -m SimpleHTTPServer 80`.
4. Na PC B ověřte, že se doménová jména `www.seznam.cz` a `www.fit.vutbr.cz` překládají na IP adresu PC A.
5. Na PC B si zobrazte webové prezentace v doménách `www.fit.vutbr.cz` a `www.seznam.cz`. Všimněte si klíče v pravé části adresního řádku.

4 Zabezpečení DNS – DNSSEC

1. Vygenerujte Key Signing Key (KSK) a Zone Signing Key. Zkontrolujte, že jste vytvořili dva páry klíčů.
2. Přidejte záznamy typu DNSKEY do zónového souboru (podle uvážení přímo, či pomocí direktivy `$INCLUDE`).
3. Podepište zónový soubor příkazem `dnssec-signzone`.
4. Upravte konfiguraci `named.conf.local` tak, aby se jako zónový soubor použil soubor obsahující podepsané záznamy.
5. Zkontrolujte nasazení DNSSEC. Je nasazení úplné?

5 Ukončení práce v laboratoři

- Počítač vypněte dávkou `/root/isa3/clean`.

Teorie

1 Network Time Protokol

NTP [4] umožňuje synchronizovat čas mezi uzly v síti. Tento protokol se dokáže vypořádat s proměnlivou dobou přenášení paketu po síti. NTP organizuje servery hierarchicky do úrovní. Tyto úrovně se nazývají *stratum*. Nejnižší hodnota 0 označuje samotný zdroj přesného času (např. GPS). Stratum 1 pak označuje servery, které jsou synchronizovány právě s referenčním zdrojem. Stratum 2 jsou servery synchronizovány se servery stratum 1, atd. [5].

Implementaci protokolu NTP zajišťuje balík aplikací *ntp*. Tento balík se skládá z několika aplikací. V rámci laboratorního cvičení se použijí aplikace `ntpd`, `ntpq` a případně `ntpdc`.

1.1 ntpd

NTPD je aplikace, která běží na pozadí a neustále provádí kontrolu času s nastavenými servery a případně upravuje lokální čas. Úprava lokálního času se provádí úpravou rychlosti běhu lokálního času. Pokud je lokální čas pozadu, resp. se předbíhá, tak se *zrychlují*, resp. *zpomalují* systémové hodiny. Tento způsob úpravy času znamená, že pokud je čas odchýlen o několik minut, bude nějakou dobu trvat, než dojde k jeho srovnání. Na druhou stranu se tak zabrání skokové změně a navíc čas se nikdy neposune do minulosti. Pokud je lokální čas odchýlen o více než 1000 sekund (necelých 17 minut) aplikace to vyhodnotí jako chybu a skončí. Pokud se tak stane, objeví se zpráva v systémovém logu. Zda aplikace běží, lze zjistit například příkazem `ntpq -p` (aplikace skončí s hláškou *ntpq: read: Connection refused* pokud není ntpd spuštěn).

Démona je možné spustit příkazem:

```
systemctl start ntp.service
```

Aby se předešlo problému v případě, kdy např. hardwarové hodiny jdou špatně a při vypnutí může dojít k odchýlení lokálního času o více než 17 minut, je možné vynutit okamžité nastavení času příkazem

```
ntpd -qg
```

(`-q` pro vynucení okamžitého nastavení času a `-g` pro vypnutí kontroly kdy je rozdíl větší než 1000 sekund).

1.1.1 Konfigurace

Základním konfiguračním souborem je `/etc/ntp.conf`. Tento konfigurační soubor obsahuje mnoho konfiguračních voleb. Nastavení serverů NTP zajišťuje konfigurační volba *server*, která má jeden parametr (adresu nebo jméno NTP serveru). Tato volba se může vyskytovat opakovaně, klient pak využívá větší počet serverů a tím lze dosáhnout vyšší přesnosti. Volba server mimo jiné znamená, že lokální čas se může nastavit podle uvedeného serveru, ale nemůže tomu být naopak. V jiných případech může být volba *server* nahrazena volbou *peer*, která umožňuje, aby se i server synchronizoval podle lokálních hodin. Tato volba je užitečná, pokud je více ekvivalentních serverů, k zajištění, že se budou synchronizovat navzájem mezi sebou. Dalšími možnostmi jsou *broadcast* a *manycastclient*. Tyto možnosti využívají broadcastového nebo skupinového vysílání (pokud je nutné synchronizovat velký počet uzlů může tato možnost šetřit síťové zdroje). Server v této konfiguraci vysílá na broadcastovou nebo multicastovou adresu informace o správném čase v pravidelných intervalech a klienti zpracovávají tyto informace.

Jinou užitečnou volbou je *restrict*, která slouží pro řízení přístupu. Aplikací `ntpd` mohou být zaslány různé požadavky přes síť (viz aplikace `ntpq`/`ntpdc`). Je vhodné dovolit některé dotazy jen z určitého uzlu nebo podsítě. Využití této volby může být také užitečné v případě, že se pro nastavování času

využívá broadcastu nebo multicastu a není žádoucí, aby takto vyslanou informaci klienti akceptovali z libovolného zdroje. Základní tvar tohoto příkazu je:

```
restrict <adresa> [mask <maska>] [<jeden či více příznaků>]
```

Příznak definuje omezení pro danou adresu/síť:

- ignore – zahazovat všechny pakety
- nomodify – povolí pouze dotazy, požadavky měnící stav serveru jsou zahazovány
- noquery – zakáže dotazy pomocí `ntpq` a `ntpdc`, synchronizace času není ovlivněna
- notrust – zahazovat neautentizované pakety

Další příznaky a jejich popis lze nalézt v manuálových stránkách `man ntp.conf`.

Pokud budou aplikace `ntpq` a `ntpdc` používané i pro změnu konfigurace, pak je nezbytné nastavit autentizaci. Protokol nabízí možnost využití symetrické i asymetrické kryptografie. Pro použití symetrické kryptografie jsou k dispozici tyto volby:

- keys – tato volba má jeden parametr, který udává název souboru, který obsahuje používané klíče (obvykle `/etc/ntp.keys`,
- trustedkey – výčet klíčů, kterým se bude důvěřovat,
- requestkey – seznam klíčů, které mohou být použity aplikací `ntpdc`,
- controlkey – seznam klíčů, které mohou být použity aplikací `ntpq`.

Formát souboru `/etc/ntp.keys` má následující tvar:

```
<číslo klíče> <typ> <heslo>
```

Typ může nabývat čtyř hodnot. Hodnoty `S` a `N` používají bitový formát a běžně se neužívají. Hodnoty `A` a `M` používají textový řetězce délky 1 až 8 znaků a určují způsob zašifrování při přenosu. Nejčastěji se užívá možnost `M`, která značí použití DES nebo MD5. Příklad souboru pak může vypadat následovně:

```
1 M heslo1
2 M secret
3 M passwd
```

Konfigurace v `/etc/ntp.conf` potom vypadá:

```
keys /etc/ntp.keys
trustedkey 1 2 3
requestkey 2
controlkey 1 3
```

Toto značí, že důvěryhodné jsou všechny tři klíče. Aplikace `ntpdc` se může autentizovat pouze klíčem 3 a aplikace `ntpq` klíči 1 a 3.

1.2 Aplikace ntpdc a ntpq

Oba dva programy nabízejí v podstatě podobné možnosti konfigurace ntp serveru. Rozdílů je mezi těmito programy několik. První, který byl již uveden výše, je v tom, že každá z těchto aplikací může používat jinou množinu klíčů. Podstatnější rozdíl z uživatelského hlediska je v tom, že aplikace `ntpdc` má přesně definovaný výčet příkazů, které lze aplikovat, a proto může měnit jen to, co je v aplikaci definováno. Naproti tomu `ntpq` disponuje příkazem `:config`, kterému se jako parametry předávají konfigurační volby, které se používají v souboru `/etc/ntp.conf`. Další rozdíl je ve formátu zpráv, pomocí kterých komunikují se serverem.

Obě aplikace používají pro komunikaci s aplikaci `ntpd` zasílání zpráv přes síťové rozhraní, bez ohledu na to, zda běží lokálně nebo vzdáleně. Hlavní výhoda však spočívá v tom, že tyto aplikace dovolují měnit konfiguraci `ntpd` za běhu. Možnost změny parametru přes síť může být nežádoucí a proto, je-li tato možnost povolena, je vhodné omezit pomocí volby *restrict* přístup pro změnu pouze přes loopback rozhraní.

Příklad výstupu volání `ntpq -p`:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+rhino.cis.vutbr 248.205.243.78   3 u  425 1024  377   10.070   -4.280  10.575
+tik.cesnet.cz   195.113.144.238  2 u  622 1024  377    4.796   -3.464  16.528
*tak.cesnet.cz   .GPS.            1 u  364 1024  377    5.008   -3.625   6.228
```

Výpis obsahuje následující hodnoty:

remote Klient se synchronizuje vůči třem serverům: rhino.cis.vutbr.cz, tik.cesnet.cz a tak.cesnet.cz. Hvězdičkou je označený primární zdroj času, plusem sekundární zdroje času.

refid Primární zdroj času pro vzdálený server NTP.

stratum Počet skoků od vzdáleného serveru k přesnému zdroji času (1 znamená, že zdroj přesného času je přímo připojen, 16 znamená, že vzdálený server je nedosažitelný).

when Počet sekund od posledního kontaktu se serverem.

poll Počet sekund mezi jednotlivými dotazy protokolem NTP.

reachability Úspěšnost posledním 8 pokusů o kontakt vzdáleného serveru v osmičkové soustavě. (1 znamená pouze poslední pokus uspěl, 377 znamená všech 8 posledních pokusů uspělo).

delay, offset, jitter Zpoždění, posun a jitter – charakteristiky vzdáleného zdroje času, podle kterého se volí primární a sekundární zdroje času.

2 Domain Name System

Cílem DNS je zajistit překlad mezi doménovým jménem a IP adresou. Dříve se jednalo především o překlad hostname na IP adresu a naopak, tedy záznamy typu `A`, `AAAA`, `PTR` [6, 8]. Dalším známým typem je `MX`, který deleguje zodpovědnost za příjem e-mailové pošty dané domény. Tento typ se od předchozích tří odlišuje, protože se nejedná o adresaci hosta, ale služby. Podobný význam mají i záznamy typu `SRV`, které lze v současnosti využít pro služby SIP a XMPP [3].

Mimo tyto typy záznamů existují i další. Některé jsou definovány již v původním návrhu, jiné přidané později nebo význam původních typů vyžit pro další služby (např. `TXT` využito pro distribuci veřejného klíče podepisování emailu DKIM [1]).

2.1 Klient

Aby klient věděl, na který server se obrátit s požadavkem na přeložení doménového jména na IP adresu či opačně, je součástí systému tzv. resolver. Tento resolver má několik konfiguračních souborů, z nichž jmenujme `/etc/hosts`, `/etc/resolv.conf` a `/etc/host.conf`.

První z těchto souborů se používá pro statický překlad doménového jména na IP adresu. Formát souboru a další popis lze nalézt v manuálových stránkách `man hosts`.

Dynamický překlad, neboli překlad pomocí DNS serveru, vyžaduje existenci konfiguračního souboru `/etc/resolv.conf`, který obvykle obsahuje jedenkrát volbu `search` a jednu nebo více voleb `nameserver`.

search definuje tzv. searchlist, neboli seznam domén (max. 6), které se budou prohledávat, pokud je požadavek na překlad neúplného doménového jména. Tedy má-li tato volba podobu `search fit.vutbr.cz`, pak při pokusu přeložit jméno *merlin* se při neúspěchu pokusí systém také přeložit jméno *merlin.fit.vutbr.cz*.

nameserver se uvádí právě s jedním parametrem, který definuje IP adresu DNS serveru, který se systém pokusí kontaktovat. Může se jednat jak o IPv4 tak o IPv6 adresu.

Další možnosti, které může soubor obsahovat, lze nalézt v manuálových stránkách `man resolv.conf`.

V souboru `/etc/host.conf` lze definovat, v jakém pořadí se předchozí dvě volby využijí. Standardně se nejprve zkouší statický překlad a poté dynamický. Toto výchozí nastavení umožňuje lokální předefinování překladu z doménového jména na IP adresu.

Pokud se síť na klientovi konfiguruje dynamicky, pak je soubor `/etc/resolv.conf` upravován automaticky. Při statické konfiguraci musí být obsah souboru upraven ručně.

2.2 Konfigurace serveru

Konfigurace DNS se stává ze dvou částí – konfigurace vlastností samotné aplikace a konfigurace obsluhovaných zón. Existuje mnoho různých variant implementací. V laboratoři se bude pracovat s implementací od ISC – BIND.

Konfigurační soubor **named.conf** se na počítačích v laboratořích nachází ve složce `/etc/bind/`. V Linuxové distribuci Kali tento soubor obsahuje vložené 3 soubory (pro konfiguraci voleb, konfiguraci lokálních zón a konfiguraci výchozích zón).

Další možnosti konfiguračního souboru naleznete v manuálových stránkách `man named.conf`.

2.2.1 Zóna typu hint

Asociuje se s nejvyšší doménou v rámci celé hierarchie. Pokud přijde na server požadavek, prohledá seznam spravovaných zón ostatních typů (*master*, *slave*, *forward*). Pokud záznam nenajde, vybere jeden z kořenových serverů definovaný právě v tomto typu zóny.

Soubor, který je vyžadován pro konfiguraci této domény lze získat například programem dig:

```
dig +norec NS . @a.root-servers.net > /etc/bind/db.root
```

2.2.2 Lokálně obsluhovane zóny

Tyto zóny lze rozdělit do dvou skupin – **loopback adresy** a **„prázdné zóny"**. RFC 6303 [2] je z kategorie *best practice* a definuje, které zóny by měl být schopen DNS server obsloužit sám a tím redukovat dotazy přeposílané na další servery. Ve většině případů jsou to zóny pro reverzní záznamy privátních IP adres.

2.2.3 Vlastní zóna

Aby server začal překládat vlastní zónu, je třeba přidat definici zóny do souboru `named.conf.local` a vytvořit zónový soubor. Pokud se přidává zónový soubor pro doménu, je vhodné přidat i zónu pro reverzní záznamy.

```
zone "moje.domena.cz" {
  type master;
  file "/var/cache/bind/moje.domena.cz.zone";
};

zone "0.168.192.in-addr.arpa" {
  type master;
  file "/etc/bind/master/192.168.0.rev";
};
```

Pro každou z těchto zón se tvoří samostatný zónový soubor, jehož základní tvar má podobu:

```
$TTL 5m

@ IN  SOA <fqdn autoritativniho serveru>. <email správce>. (
  1   ; seriové číslo
  10h ; obnovovací interval pro sekundární servery
  10m ; prodleva, po které se sekundární server pokusí znova kontaktovat
      ; autoritativní server, pokud se předchozi spojení nezdařilo
  1w  ; jestliže se sekundárnímu serveru nepodaří kontaktovat autoritativní
      ; server během této doby, přestane vracet záznamy pro tuto doménu
  1h  ; původně výchozí TTL (nahrazeno $TTL), nově určuje
      ; NEGATIVNÍ TTL -- pro chybové odpovědi (např. NXDOMAIN)
  )

@ IN  NS <fqdn autoritativního serveru>
@ IN  NS <fqdn sekundárního serveru>

následuje seznam dalších záznamů
```

Typ záznamu **SOA** definuje parametry zóny – jméno autoritativního serveru a email správce domény. Všimněte si, že znak '`@`' má zvláštní význam (zastupuje název domény). Z tohoto důvodu se v emailu správce nahrazuje znak '`@`' za znak '`.`' (tečka).

Záznam typu **NS** by měl být vždy obsažen alespoň jednou a to právě pro autoritativní server. Počet sekundárních serverů je libovolný. Tyto záznamy by také měly byt obsaženy v nadřazené doméně, aby bylo možné nalézt server zodpovědný za danou poddoménu. Tedy kořenová doména obsahuje `NS` záznamy pro domény nejvyšší úrovně *.cz, .com, .eu, ...* a ty zase pro jednotlivé poddomény *google.com, seznam.cz, ...*

V zónovém souboru se rozlišuje mezi relativním a absolutním jménem. O jaký typ jména jde, se rozlišuje podle zakončení. Končí-li tečkou, je to jméno absolutní (plně kvalifikované). V opačném případě jde o jméno relativní a za to se vždy doplní obsah definován v `$ORIGIN`. Jestliže není definován, použije se název, který je definován u názvu zóny v souboru `named.conf`.

Další záznamy mají obecný tvar:

```
jméno  ttl  třída  typ  <na typu závislá data>
```

Jméno běžně bývá ve tvaru relativním. Pokud není hodnota TTL jiná než je výchozí pro celou zónu, pak se může vynechat. Třída záznamu se běžně používá pouze `IN` (Internet). Kromě typu SOA a NS, se v laboratorním cvičení použijí záznamy typu `A`, `PTR` a volitelně `AAAA`. Typ `AAAA` se chová stejně jako záznam typu `A`, který se používá pro IPv4. Typově závislými daty je v tomto případě adresa IPv6, resp. IPv4. Příklad:

```
host  IN A  192.168.0.1
```

```
host  IN AAAA  fd12:1234::1
```

Typ `PTR` nerozlišuje, zda se jedná o IPv6 nebo IPv4 adresu. Pro tento případ se typově závislá data chovají stejně jako jméno na začátku záznamu. V tomto případě se ale vždy zapisuje v plném tvaru. Další zvláštností PTR záznamu, který je vidět i v názvu zóny, je, že se hodnoty zapisují v obráceném pořadí a každá hodnota je oddělená tečkou. Obrácené pořadí je kvůli vyhodnocování záznamů od konce a tečka mezi každým číslem dovoluje snadnější delegaci podsítě na jiný DNS server. Příklad:

```
1 IN PTR host.moje.domena.cz.
```

Pro IPv6 můžete využít:

```
$ORIGIN 0.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN PTR host.moje.domena.cz.
```

Oproti tomu záznamy

```
$ORIGIN 0.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN PTR host
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN PTR host.moje.domena.cz
```

jsou neplatné, protože na dotaz by vrátily:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa. IN PTR
                                                    host.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa. IN PTR
                                host.moje.domena.cz.0.0.0.0.0.0.0.4.3.2.1.2.1.d.f.ip6.arpa.
```

Příklad PTR záznamu pro IPv4 je:

```
19.176.229.147.in-addr.arpa. IN PTR merlin.fit.vutbr.cz.
```

Po každé změně zónového souboru by se vždy mělo upravit i sériové číslo zóny a to tak, aby bylo vždy větší než předchozí hodnota. Také je nutné aplikaci načíst změněnou konfiguraci.

```
systemctl restart bind9.service
```

3 DNS Security Extensions (DNSSec)

DNS je službou, která si z inulosti nese různé problémy. Jedním z nich je nezabezpečení této služby vůči útokům s podvržením záznamu. DNSSEC rozšiřuje službu DNS o bezpečnostní prvky, jejichž cílem je zabránit útokům využívající podvržené záznamy v DNS. Obrana spočívá v zavedení asymetrické kryptografie pro podepisování zón. Přesněji řečeno se podepíše každý záznam v zóně a tyto podpisy se zveřejní jako speciální typ záznamu. Protože není možné, aby resolver měl uložený veřejný klíč od každé zóny, používá se metoda sestavení důvěryhodného řetězce s tím, že resolver v tomto případě potřebuje mít uložen pouze veřejný klíč nejvyšší domény. Aby tento systém fungoval, musí nadřazená doména obsahovat nejen záznamy typu `NS` ale i záznam typu `DS` – speciální typ záznamu obsahující hash klíče a název domény. Principiálně samozřejmě může mít resolver uložen veřejné klíče více zón.

Pro vytvoření klíčů a podepsaní zóny není třeba zpracovávat každý záznam samostatně. Součástí balíčku *bind* jsou aplikace, kterým stačí zadat vhodné parametry, a ty zajistí vytvoření všech potřebných souborů. Pak již jen stačí správně nakonfigurovat aplikaci `named`, aby tyto soubory používala.

3.1 Klíče

Není to nezbytně nutné, ale obecně se doporučuje používat v DNSSec dva typy klíčů:

- Key Signing Key (KSK) a
- Zone Signing Key (ZSK).

Ve své podstatě se tyto klíče liší jen v jediném bitu (SEP – Secure Entry Point), který říká, že od tohoto místa může byt zahájeno sestavovaní důvěryhodného řetězce. V praxi se tyto dva klíče liší ještě další vlastností, která se určuje při jejích vytváření – délkou. Důvodem je vztah mezi délkou klíče, bezpečností a výpočetními nároky. Delší klíč je sice bezpečnější, ale zároveň má vyšší nároky na výkon a naopak. Protože k překladu DNS dochází velice často, není žádoucí, aby tato operace byla náročná na výpočetní zdroje. Proto byl zvolen postup, kdy klíč, který je často používán, je sice slabší, ale na druhou stranu jej lze měnit relativně často bez větších obtíží, a je-li čas nutný k jeho prolomení hrubou sílou delší, než délka jeho platnosti, pak jeho prolomení nepředstavuje problém – tímto klíčem je klíč pro podepisování zóny (ZSK). Druhý klíč (KSK) se použije pouze pro podepsání ZSK, což znamená, že se nepoužívá tak často. V důsledku tento klíč může být delší a proto i odolnější vůči prolomení. Výměna tohoto klíče je obvykle trošku složitější, neboť vyžaduje spolupráci se správcem nadřazené domény. Čas, po kterém je nutné klíče obměnit, není explicitně definován a stejně tak klíče neobsahují časové vymezení platnosti. Po jaké době provést výměnu klíčů je tedy v režii správce domény a víceméně to závisí i na frekvenci používaní klíče k podpisu. Dochází-li v doméně k častým změnám a tedy i k častému podepisování, je vhodné měnit klíče častěji.

Pro vygenerování klíčů je v balíku *bind* k dispozici aplikace `dnssec-keygen`, které stačí jediný parametr[1] – název domény. Další užitečné parametry jsou:

`-f KSK` pokud má mít vygenerovaný klíč nastaven SEP bit,

`-b <číslo>` pro nastavení délky klíče,

`-a <algoritmus>` pro výběr algoritmu a

`-r` kterým se nastaví zdroj náhodných dat (standardně se použije `/dev/random`, který je blokující, pokud nemá dostatek entropie a může být vhodné nahradit ho `/dev/urandom`).

Pro získaní seznamu dalších možností stačí spustit bez parametru. Po spuštění aplikace se správnými parametry je vypsán na výstup řetězec, který se skládá z názvu domény, číselného kódu použitého algoritmu a identifikačního čísla klíče. Kromě toho vytvoří dva soubory, jejichž jména začínají vypsaným řetězcem a končí příponami `.key` a `.private`. První z nich obsahuje veřejný klíč a ten druhý privátní. Podle toho by se také s těmito klíči mělo zacházet (omezit přístup k privátnímu klíči, apod.). Příkazy:

```
dnssec-keygen -r /dev/urandom -a RSASHA1 -b 2048 -f KSK example.com
dnssec-keygen -r /dev/urandom -a RSASHA1 -b 1024 example.com
```

vytvoří čtyři soubory, kde první pár je KSK a druhý ZSK. Oba klíče je třeba zveřejnit v zóně. Vypsáním obsahu souborů s příponou `.key` zjistíte, že záznamy již mají tvar požadovaný pro zápis do zóny. Obsah těchto souboru se může buďto zkopírovat do zónového souboru a nebo je připojit pomocí direktivy `$INCLUDE`, např:

```
cat Kexample.com.+005+06487.key >> /var/cache/bind/example.com
cat Kexample.com.+005+32883.key >> /var/cache/bind/example.com
```

Alternativně přidejte do souboru `/var/cache/bind/example.com` řádek:

[1] Závisí na implementaci, v některých systémech může požadovat více parametrů.

```
$INCLUDE Kexample.com.+005+06487.key;
$INCLUDE Kexample.com.+005+32883.key;
```

Nyní už jen stačí zónový soubor podepsat. To se provede příkazem `dnssec-signzone`, kterému se pouze předá parametrem název souboru, který obsahuje podepisovanou zónu. Pokud se název souboru neshoduje s názvem zóny, pak je třeba použít ještě parametr `-o`, kterým se určí název podepisované zóny. Pro podpis zóny se použijí automaticky ty klíče, jejichž veřejné části byly vloženy do zónového souboru (viz odstavec výše).

```
dnssec-signzone -o example.com labXX.zone
```

Parametr `-o` specifikuje origin a `labXX.zone` je jméno zónového souboru (vstupního souboru). Výstupem tohoto příkazu je soubor se stejným názvem jako vstupní soubor rozšířený o příponu *.signed*. Pokud je požadován jiný název výstupního souboru, lze toho docílit volbou `-f`. Takto vzniklý soubor má požadovaný tvar zónového souboru. Stačí jen upravit `/etc/namedb/named.conf`, aby místo původního zónového souboru (bez podpisů) začal používat nově vzniklý soubor, který obsahuje i podpisy záznamů.

V souboru podepsané zóny se objevily nové typy záznamů `DNSKEY`, `RRSIG` a `NSEC`. První z nich je typ, který nese data o veřejné části klíče. `RRSIG` obsahuje samotný podpis plus další informace – typ podepsaného záznamu, jeho TTL, časy od kdy a do kdy je podpis platný a ID klíče, kterým byl podpis proveden. Poslední ze záznamu (`NSEC`) obsahuje informaci o následujícím záznamu v abecedně seřazené zóně. Tento záznam se využívá při odpovědi pro dotaz na neexistující záznam. Protože součásti podpisu je i časové vymezeni jeho platnosti, je nutné provádět znovupodepsání zóny. To by se mělo provést dříve, než vyprší platnost starých záznamů (nutné počítat se zkrácením hodnotou TTL, po kterou mohou být staré záznamy uloženy v cache paměti).

V tuto chvíli je již zóna podepsána a je možné ji začít používat. Ovšem resolver, který bude ověřovat, že je zóna dobře podepsaná, nemá k dispozici veřejný klíč (z jiného bezpečného zdroje), a nadřazená zóna neobsahuje informaci o použitém klíči a není proto možné sestavit důvěryhodný řetězec. Zbývá požádat správce nadřazené zóny o zveřejnění `DS` záznamu. Tento DS záznam se vytváří pro KSK klíč. Při podpisu zóny vznikne také soubor začínající řetězcem *dsset-* a končící názvem zóny. Tento soubor obsahuje `DS` záznamy, které je potřeba zveřejnit. Tyto záznamy lze také získat příkazem `dnssec-dsfromkey`. Pokud se nepoužijí žádné volby, pak tento program standardně vypíše na výstup dva DS záznamy ve stejné formě, jako je ve výše uvedeném souboru (jeden s použitím algoritmu SHA-1 a druhý s SHA-256). První z nich by měl být zveřejněn povinně.

```
dnssec-dsfromkey <název souboru s KSK klíčem>
```

Výměna klíčů

Jak bylo uvedeno výše, používají se obvykle dva typy klíčů, kdy jeden je slabší a druhy silnější. Oba tyto klíče je třeba obměňovat, aby nedošlo ke kompromitování zóny. Slabší z klíčů, ZSK, je třeba měnit častěji, ale vzhledem k tomu, že se změnou tohoto klíče není svázaná nutnost změny v nadřazené zóně, je tento krok relativně jednodušší (alespoň po administrativní stránce).

Při vyměně klíčů je nutné zajistit, aby klienti měli vždy k dispozici klíč, kterým jsou data podepsána. Způsoby, jak toho dosáhnout jdou dva – buďto podepsat data starým i novým klíčem, nebo nejprve zveřejnit nový klíč, počkat až se klíč rozšíří, a pak jej začít používat k podepisování zóny. Varianta, kdy se data podepíší starým i novým klíčem, je sice rychlejší, ale rovněž znamená, že zónový soubor zvětší svou velikost (a to téměř dvojnásobně). Z tohoto důvodu se tento způsob používá pouze pro výměnu KSK klíčů. Druhý způsob vyžaduje více časů, ale předchází zvětšování zónového souboru – běžně se používá pro výměnu ZSK.

3.2 Ověřování podepsané zóny

Bind od verze 9.5 má validaci zapnutou implicitně. Liší se ovšem způsob zpracování veřejného klíče. V současné době již je podepsána zóna `root`, a není proto nezbytně nutné zabývat se distribucí klíčů pomocí DLV registrů.

Do verze 9.6 se používala direktiva *trusted-keys* v konfiguračním souboru, která měla obdobný formát jako `DNSSEC` záznam. Při použití této možnosti je nutné hlídat, zda nedošlo ke změně klíče root zóny a v tomto případě tento klíč obnovit. Od verze 9.7 se používá volba *managed-keys*, která má stejný formát, a funguje v podstatě stejně jako *trusted-keys*, ale navíc dovoluje použití automatického sledování změny klíče root zóny definované v RFC 5011 a nevyžaduje zásah operátora v případě obměny [7].

Reference

[1] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. DomainKeys Identified Mail (DKIM) Signatures. RFC 4871 (Proposed Standard), May 2007. Updated by RFC 5672.

[2] M. Andrews. Locally Served DNS Zones. RFC 6303 (Best Current Practice), July 2011.

[3] A. Gulbrandsen, P. Vixie, and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). RFC 2782 (Proposed Standard), February 2000.

[4] D. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. RFC 1305 (Draft Standard), March 1992. Obsoleted by RFC 5905.

[5] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905 (Proposed Standard), June 2010.

[6] P.V. Mockapetris. Domain names - implementation and specification. RFC 1035 (Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966.

[7] M. StJohns. Automated Updates of DNS Security (DNSSEC) Trust Anchors. RFC 5011 (Proposed Standard), September 2007.

[8] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi. DNS Extensions to Support IP Version 6. RFC 3596 (Draft Standard), October 2003.