DNSSEC a bezpečné DNS
DNSSEC část první aneb je potřeba začít od píky

Princip DNS

Jak tedy systém DNS pracuje? Obecně se na celý systém DNS dá pohlížet jako na decentralizovanou hierarchickou databázi, jejíž hlavní smysl je poskytovat informace nutné pro překlad jmen na IP adresy. Informace uložené v DNS se ukládají ve formě, která se nazývá Resource Record (dále též RR záznam). Protože data uložená v RR záznamech jsou spíše statického charakteru, mají samotné informace i specifikaci maximální doby platnosti záznamu, tzv. TTL (Time to Live). Celá struktura jednoho RR záznamu obsahuje vlastníka (owner), třídu (class), typ (type), ttl a data (rdata – resource data). Třídy byly v návrhu specifikovány dvě: IN – pro Internet, a CH – pro Chaos. Třída Chaos byla vytvořena pro experimentální účely a v praxi se s ní můžete potkat pouze při speciálním typu dotazu na verzi DNS serveru (CH TXT version.bind.). Mezi základní typy RR záznamu patří: A – pro IPv4 adresy, MX – pro směrování pošty, AAAA – pro IPv6 adresy, a další. Tyto typy jsou registrovány u organizace IANA a jsou postupně doplňovány dle potřeby dalších protokolů, které s DNS pracují. Např. DNSSEC zavádí hned několik nových typů RR záznamů. V datech jsou pak uložené informace, které se liší dle typu, již zmiňovaný A záznam obsahuje IPv4 adresu. A ještě jedna technická poznámka. RR záznamy jsou sdruženy podle názvu, třídy a typu do RRsetů, které sdílejí stejné TTL. Pokud například máte více poštovních serverů, tak nelze při zadávání MX záznamů mít pro každý server jiné TTL.

Programy, které s DNS pracují, můžeme rozdělit na dvě základní kategorie: klient a server. Klient je program, který umí procházet hierarchickou strukturu DNS a zjišťovat informace uložené v RR záznamech. Server je pak program, který umí odpovídat na dotazy klientů, pro konkrétní doménová jména, pro které je nakonfigurován. Protože by implementace obecného DNS klienta do každého programu byla složitá, je v systémové knihovně implementovaný jenom jednoduchý DNS klient (stub resolver), který se umí zeptat jenom na konkrétní jméno a očekává odpověď v jednoduché formě. Z tohoto důvodu může být DNS klient i serverem, který poskytuje informaci tomuto stub resolveru. DNS server, který se chová jako klient, budeme nazývat resolver nebo také rekurzivní DNS (rDNS). Protože rDNS má většinou (nikoli nutně) vyrovnávací paměť, do které si ukládá výsledky již provedených dotazů, můžeme se také setkat s označením kešující DNS. DNS server, který odpovídá na dotazy, nazveme autoritativní DNS (aDNS). A aby to nebylo příliš jednoduché, tak DNS server může vystupovat v obou rolích zároveň. Na dotazy na doménová jména, pro které je server autoritativní, odpovídá přímo – chová se jako aDNS, ostatní dotazy přeposílá jiným autoritativním serverům, tedy se chová jako rekurzivní DNS server. Nicméně tato konfigurace není doporučována a z různých důvodů je lepší mít tyto dvě role – rekurzivní a autoritativní – odděleny.

Pomalu jsme se propracovali přes definice k tomu, co se vlastně stane, když program potřebuje např. přeložit jmenný název na IP adresu. Překlad jména na IP adresu se provede voláním systémové funkce getaddrinfo, která vytvoří strukturu addrinfo. Struktura addrinfo umí IPv6 i IPv4 adresy a program ji může využít dále pro vytváření IP spojení. Co se vlastně všechno stane ve chvíli, kdy dojde k volání getaddrinfo? Otevřete prohlížeč a chcete se podívat na stránky vlády České Republiky, napíšete tedy do svého prohlížeče www.vlada.cz. Prohlížeč zavolá funkci:

```
getaddrinfo("www.vlada.cz", 80, &hints, &result).
```

Voláním této funkce se předá kontrola stub resolveru, který se nejprve podívá do lokálního souboru /etc/hosts (a jsme zpátky u hosts.txt) a pokud informaci nenalezne zde, zeptá se rDNS serverů nakonfigurovaných v /etc/resolv.conf. Pořadí dotazování lze nakonfigurovat v /etc/nsswitch.conf, v dnešních systémech se mezi tyto dvě metody zařazuje ještě multicast DNS. Ale pojďme zpátky k našemu dotazu na www.vlada.cz. rDNS server přijme dotaz a podívá se, zda-li odpověď nemá ve vyrovnávací paměti, pokud ano, vrátí rovnou tuto odpověď. V opačném případě začne hierarchicky od kořenové úrovně (jinak také root, v DNS je reprezentovaný nepovinnou tečkou úplně napravo v doménovém jméně) prohledávat DNS databázi. Aby vůbec mohl začít, musí mít staticky nakonfigurované DNS servery pro kořenovou úroveň. Takových serverů je v současné době 13. Náš rDNS se zeptá náhodně vybraného kořenového serveru na „www.vlada.cz“, tento server ovšem tuto informaci neví (protože je systém hierarchický, distribuovaný a decentralizovaný), ale ví, že správcem domény .cz je CZ.NIC, resp. jeho DNS servery. Proto odpoví „já to nevím, ale zeptej se serverů CZ.NICu“, prakticky to pak vypadá takto:

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

;; AUTHORITY SECTION:
cz.         172800  IN  NS  F.NS.NIC.cz.
cz.         172800  IN  NS  C.NS.NIC.cz.
cz.         172800  IN  NS  E.NS.NIC.cz.
cz.         172800  IN  NS  B.NS.NIC.cz.
cz.         172800  IN  NS  A.NS.NIC.cz.
cz.         172800  IN  NS  D.NS.NIC.cz.

;; ADDITIONAL SECTION:
A.NS.NIC.cz.        172800  IN  A   217.31.205.180
A.NS.NIC.cz.        172800  IN  AAAA    2001:1488:dada:176::180
B.NS.NIC.cz.        172800  IN  A   217.31.205.188
B.NS.NIC.cz.        172800  IN  AAAA    2001:1488:dada:184::188
C.NS.NIC.cz.        172800  IN  A   195.66.241.202
C.NS.NIC.cz.        172800  IN  AAAA    2a01:40:1000::2
D.NS.NIC.cz.        172800  IN  A   193.29.206.1
D.NS.NIC.cz.        172800  IN  AAAA    2001:678:1::1
E.NS.NIC.cz.        172800  IN  A   194.146.105.38
F.NS.NIC.cz.        172800  IN  A   193.171.255.48
F.NS.NIC.cz.        172800  IN  AAAA    2001:628:453:420::48
```

rDNS server si přečte, že se má zeptat serverů CZ.NICu a pošle dotaz jednomu z těchto serverů. Zde se situace opakuje a rDNS je odkázán na servery ns.vlada.cz a ns2.gts.cz, které obsluhují doménu vlada.cz. Výsledek vypadá takto:

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

;; AUTHORITY SECTION:
```

```
vlada.cz.       18000   IN  NS  ns.vlada.cz.
vlada.cz.       18000   IN  NS  ns2.gts.cz.

;; ADDITIONAL SECTION:
ns.vlada.cz.        18000   IN  A   212.47.23.110
```

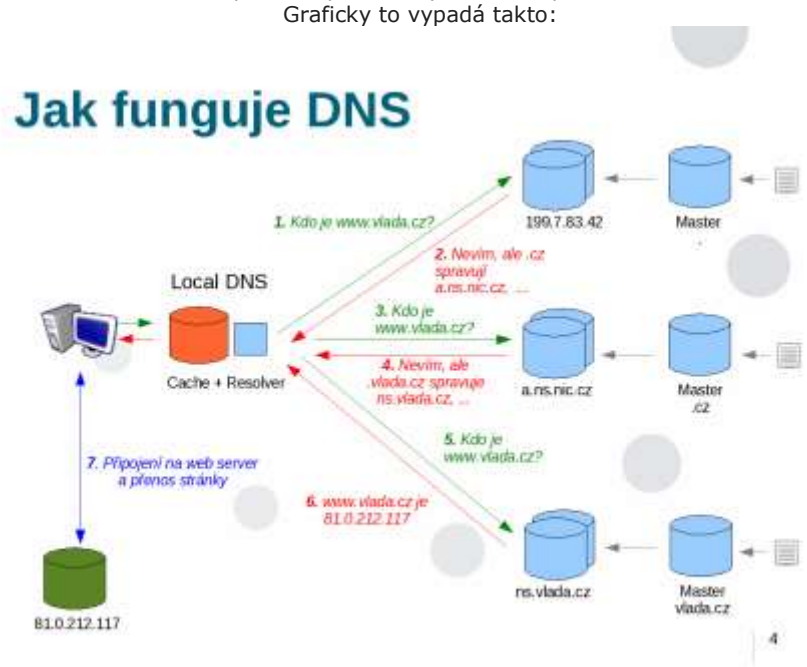
Opět je vyslán další DNS dotaz, tentokrát již na doménové servery domény vlada.cz, tedy například ns.vlada.cz. Tento server již požadovanou informaci ví a našemu dotazujícímu se serveru odpoví:

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

;; ANSWER SECTION:
www.vlada.cz.       600 IN  A   212.47.23.116
```

Nyní rDNS konečně dostal odpověď, kterou potřeboval, a na úplně původní dotaz stub resolveru může odpovědět s informací 212.47.23.110. Veškeré informace, které při získávání IP adresy pro server www.vlada.cz dostal, si uloží do vyrovnávací paměti, včetně informace o TTL. Příště, pokud dostane stejný dotaz v časovém limitu, který je pro www.vlada.cz 10 minut, bude moci odpovědět přímo z vyrovnávací paměti.

Graficky to vypadá takto:



V tuto chvíli je potřeba se ještě na chvíli zastavit u sekce ADDITIONAL. V této sekci posílá aDNS tzv. GLUE záznamy. Představme si situaci, kdy kořenové nameservery pošlou odpověď: „nedokáži ti odpovědět, co je www.vlada.cz, ale možná to ví a.ns.nic.cz." Pokud by v další sekci DNS zprávy nepřišly informace o IP adrese serveru a.ns.nic.cz, musel by rDNS zjistit IP adresu serveru a.ns.nic.cz sám. Jak by nejspíš vypadala odpověď kořenového serveru na dotaz ohledně IP adresy a.ns.nic.cz? „Nedokáži ti odpovědět, ale možná to ví a.ns.nic.cz". V tu chvíli bychom se dostali do nekonečné smyčky, a původní dotaz by zůstal nezodpovězen.

V další části seriálu o DNSSEC si řekneme něco o struktuře DNS zprávy, příznacích v DNS dotazech a DNS odpovědích, a omezeních a rozšířeních DNS protokolu.

Malý slovníček pojmů:

aDNS – Autoritativní DNS – DNS server, který ví informace o konkrétní zóně, tedy je pro tuto zónu autoritativní.
GLUE záznam – informace o IP adrese, která se posílá navíc v odpovědích, kdy je delegace zóny směrovaná do nameserverů uvnitř zóny.
rDNS – Rekurzivní DNS – Resolver – DNS server, který umí rekurzivně procházet DNS strom a odpovídat na dotazy klientů.
RR záznam – Resource Record – jednotlivá položka v DNS, obsahuje vlastníka, TTL, typ, třídu, a data záznamu.
TTL – Time To Live – údaj označující dobu platnosti, po kterou si může rDNS držet RR záznam ve vyrovnávací paměti.
Zóna – část DNS stromu, která je delegovaná na samostatné DNS servery.

DNSSEC a DNS zpráva pod drobnohledem

DNS servery mezi sebou komunikují pomocí DNS zpráv, které mají vždy standardní formát. Tento formát je v nejvyšší úrovni rozdělený na pět základních sekcí:

Pět sekcí DNS zprávy

Název sekce	Popis
Hlavička (Header)	Hlavička DNS zprávy
Dotaz (Question)	Dotaz pro DNS server
Odpověď	Odpověď DNS serveru

Pět sekcí DNS zprávy

Název sekce	Popis
(Answer)	
Autorita (Autority)	Sekce ukazující na autoritativní servery
Další (Additional)	Sekce obsahující další záznamy

Hlavička je v DNS zprávě vždy přítomna a obsahuje informace o příznacích zprávy, návratovou hodnoty odpovědi zprávy, a informaci o přítomnosti dalších částí. Pokud si vzpomenete na článek o útoku na DNS, se kterým tento rok přišel DAN KAMINSKY, tak ono zmiňované transakční ID se také nalézá v hlavičce zprávy. Bohužel hlavička DNS zprávy má fixní formát a velikost, z čehož vyplývá první omezení DNS protokolu, tak jak byl definovaný v RFC1035 – hlavička DNS zprávy může obsahovat pouze omezené množství příznaků. Jak lze toto omezení obejít, se dozvíte ke konci tohoto článku.

V tuto chvíli udělám malou odbočku k příznakům, které jsou definovány v hlavičce DNS zprávy. Jsou to QR, AA, TC, RD a RA. QR (QueRy) je příznak, který určuje, zda je zpráva dotaz nebo odpověď. AA (Authoritative Answer) je příznak, který vrací autoritativní servery u odpovědí na dotazy, které vedly do zón, které obsluhují. Dotaz, který položíte rekurzivnímu serveru, by nikdy neměl obsahovat tento příznak. TC (TrunCation) je příznak, který označuje, že DNS zpráva byla zkrácena, a dotazující se má zeptat znovu přes TCP protokol. RD a RA jsou dva příznaky, které spolu souvisí. Příznak RD (Recursion Desired) posílá klient (například stub resolver), který se ptá rekurzivního serveru a vyžaduje od něj, aby provedl rekurzivní doptávání na jeho dotaz. RA (Recursion Available) je pak příznak, který posílá zpátky dotazovaný server, aby dal najevo, že je ochotný toto rekurzivní doptávání provést. Pokud se zeptáte serveru, který je pouze autoritativní, příznak RA nebude nastaven.

Před přechodem k další sekci bych rád představil termín oktet. Nejedná se o skladbu pro osm nástrojů, v počítačové terminologii je oktet označení pro 8 bitů. Zde je důležité si uvědomit, že standardy RFC 1034 a 1035 vznikaly v době, kdy nebylo tak přirozené, že bajt má 8 bitů. Proto je místo slova bajt použit termín oktet.

Další sekcí DNS zprávy je dotaz. Dotaz zprávy obsahuje tři části – QNAME, QTYPE a QCLASS. QNAME (z Query Name) obsahuje doménové jméno, na které se dotazujeme. QNAME se skládá s jednotlivých labelů (label je vždy část mezi dvěma tečkami), resp. samotný label předchází jeho délka reprezentovaná jedním oktetem. Pro samotnou délku je využíváno pouze nižších 6 bitů, a z toho vyplývá, že maximální délka jednoho labelu je 63 oktetů (tj. samé jedničky na nižších 6 bitech). Celý QNAME je ukončen speciálním označením pro kořenovou zónu – labelem o délce 0. Maximální délka sekce je 255 oktetů včetně jejich oktetů s délkami labelů a včetně labelu pro kořenovou zónu. QTYPE a QCLASS odpovídají Typu a Třídě RR záznamu, nicméně jsou lehce rozšířeny o některé další hodnoty, např. 255 znamená: vrať mi libovolný DNS záznam (tzv. dotaz ANY).

Další tři sekce obsahují vždy pouze RR záznamy. Formát RR záznamu jsme si již představili v minulém článku. RR záznam obsahuje:

Obsah RR záznamu

Název	Popis
Vlastník (Owner Name)	Doménové jméno vlastníka RR záznamu
Typ (Type)	Typ RR záznamu (2 oktety)
Třída (Class)	Třída RR záznamu (2 oktety)
TTL	Délka platnosti záznamu (4 oktety)
Délka sekce RDATA (RDLENGTH)	Délka sekce RDATA (2 oktety)
RDATA	Data RR záznamu

Rozdíl je tedy jen v zápisu – doménové jméno vlastníka záznamu je rozloženo na jednotlivé labely, tak, jak to bylo rozebráno v sekci Dotaz pro QNAME, a samotná data záznamu v DNS zprávě předchází jejich délka vyjádřená 16bitovým číslem.

Pro úsporu místa v DNS zprávě byl vymyšlen mechanismus komprese doménových jmen. Pokud délka labelu na horních dvou bitech obsahuje jedničky, pak dolních šest bitů neobsahuje délku, ale ukazatel na předchozí výskyt doménového jména. Protože je doménové jméno rozkouskováno na jednotlivé oktety, může jeho zápis v DNS zprávě začínat jedním nebo několika labely (např. „www“) a končit ukazatelem na předchozí výskyt (např. „dnssec.cz“).

DNS odpověď bude vždy obsahovat alespoň jeden RR záznam, nebo v hlavičce DNS zprávy v sekci návratového kódu bude obsahovat důvod, proč nemohl být dotaz vyřízen. Mezi nejčastější důvody je neexistence doménového jména (NXDOMAIN), odmítnutí dotazu (REFUSED) nebo chyba na straně serveru (SERVFAIL). Jednotlivé sekce budou naplněny podle toho, zda server posílá přímo odpověď (sekce Odpověď), posílá-li informaci o delegaci na jiné servery (sekce Autorita) nebo posílá-li např. IP adresy DNS serverů, tzv. GLUE záznamy (sekce Další). DNSSEC následně tyto sekce využívá také pro informace o vlastních RR záznamech, ale k tomu se dostaneme v další sekci našeho seriálu.

Na závěr jsem vám slíbil něco o omezeních DNS protokolu. RFC 1035 jich definuje hned několik a některá jsme zmínili už v průběhu článku. Jsou to:

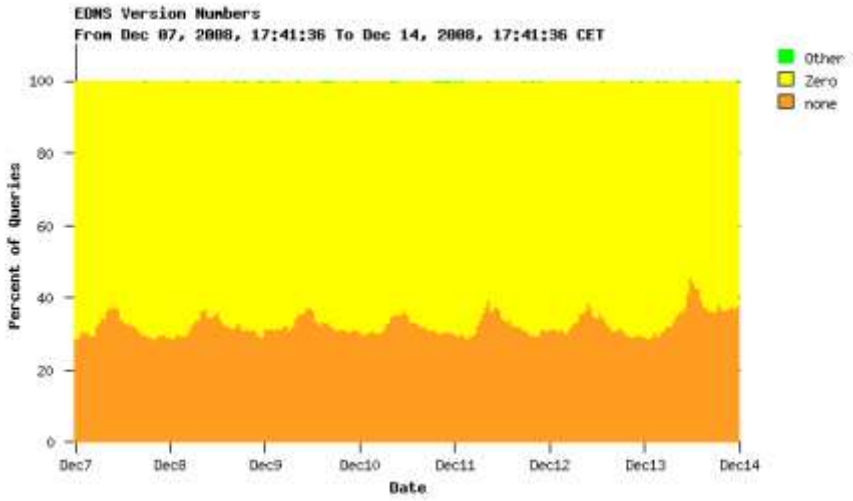
- maximální délka labelu je 63 oktetů
- maximální délka doménového jména je 255 oktetů
- TTL je 32bitové číslo
- pro zachování interoperability byl label striktně omezen na alfanumerické znaky a pomlčku
- maximální velikost UDP paketu je 512 oktetů
- počet použitelných příznaků v hlavičce DNS zprávy je omezený

Maximální velikosti labelu, doménového jména a TTL zůstaly zachovány do dnešních dní. TTL změnit asi už nepůjde a ani se nezdá, že by existovala potřeba toto pole zvětšit. Maximální délka labelu již také nejspíš zvětšit nepůjde. Nejméně problematická je maximální délka doménového jména. Toto omezení nevychází ze struktury DNS zprávy, ale bylo definováno, aby lidé, kteří implementují DNS protokol, měli jednodušší práci (pořád se pohybujeme v roce 1987). Dnes je toto omezení spíše historické. Velká část systémů má v sobě toto omezení „zadrátováno" a změnit toto omezení by byl úkol téměř nadlidský.
Omezení znaků v labelu již dávno neplatí. Plně osmibitové znaky do něj sice stále vkládat nelze, ale např. SRV záznamy používané pro směrování různých služeb obsahují podtržítko na začátku záznamu a nikdo se nad tím nepozastavuje. Čistě teoreticky lze do DNS stromu ukládat libovolná binární data, protože návrh byl hodně obecný, nicméně prakticky to moc nebude fungovat.
Poslední dvě omezení – maximální velikost DNS zprávy posílané přes UDP a počet příznaků v hlavičce – byly vyřešeny ve standardu pojmenovaném Extension Mechanisms for DNS, zkráceně také EDNS0, který je popsán v RFC2671. Toto RFC definuje speciální typ RR záznamu pojmenovaný OPT, který se „schovává" v sekci Další a přetěžuje jednotlivé pole RR záznamu takto:

OPT záznam

Název pole	Popis
Vlastník	Vždy kořenová zóna
Typ	OPT
Třída	Maximální akceptovaná velikost UDP paketu
TTL	Rozšířený návratový kód a verze EDNS
RDATA	Rozšířené příznaky

Standard EDNS0 vznikl v roce 1999 a příští rok oslaví deset let. I proto je vcelku pozoruhodné, že více než 30 % DNS serverů, které se ptají autoritativních serverů pro doménu .cz, nepoužívá EDNS0. Z hlediska technologie DNSSEC je podpora EDNS0 nutná pro samotné fungování DNSSEC, protože jsou využívány rozšířené příznaky i větší velikost DNS zpráv.



Dnešní část seriálu věnovaná DNS zprávě končí a příště si už konečně povíme něco technologii DNSSEC.

DNSSEC a jeho historie

V zásadě se nejedná o chybu samotného DNS, ale chybu přílišné důvěry v systém DNS. Publikace této práce byla odložena z bezpečnostních důvodů o celé čtyři roky, ale jak sám autor poznamenává, nebylo to příliš moudré rozhodnutí, protože informace z této práce stejně prosákly na veřejnost a byly zaznamenány útoky, které využívaly tento nadbytek důvěry v doménový systém.
Po zveřejnění tohoto nedostatku se na půdě IETF, což je organizace, která stojí za většinou standardů používaných na internetu, začíná uvažovat nad zabezpečením systému DNS. Mezitím EUGENE KASHPUREFF objevuje další zranitelnost v současných implementacích DNS serverů (tedy v té době především serveru Bind), která umožňuje v DNS podvrhnout libovolný záznam pomocí sekce „Další". V roce 1997 pak používá tuto chybu k celosvětovému nebo spíše celointernetovému přesměrování stránek registrátora InterNIC na stránky své firmy AlterNIC. Jako ochrana proti tomuto typu útoku byl vymyšlen systém správních oblastí (bailiwick), který definuje, jaká data budou v sekci „Další" akceptována. Jako následující krok ještě v témže roce vzniká první dokument popisující kryptografické zabezpečení systému DNS – RFC 2065. Tento dokument je pak během dvou dalších let rozpracován a v roce 1999 je v RFC 2535 publikována první verze systému DNSSEC. DNSSEC definuje do systému DNS nové záznamy, které kryptograficky zajišťují integritu dat poskytovanou DNS servery.
První implementace DNSSECu dle RFC 2535 je připravena v DNS serveru Bind, bohužel další dva roky nasazení DNSSECu stagnuje a v roce 2001 je vypracována studie, která konstatuje, že tato první verze systému DNSSEC je nevhodná k nasazení, protože libovolná výměna podepisovacího klíče vyžaduje jednak komplexní komunikaci s nadřazeným DNS serverem a jednak se tato změna musí promítnout na všech podřízených DNS serverech. Tyto požadavky první verzi DNSSEC odsunuly na vedlejší kolej a začíná se pracovat na verzi nové pracovně nazývané DNSSECbis. DNSSECbis definuje úplně nové záznamy, které nejsou kompatibilní s původní verzí DNSSECu a rozšiřuje DNS o nový druh záznamu – DS, který má výrazně zjednodušit komunikaci s nadřazeným serverem. V průběhu let 2002 až 2003 je tato nová verze DNSSECu implementována v DNS serveru Bind a

ukazuje se, že je na rozdíl od své předchůdkyně životaschopná. V roce 2004 je podpora DNSSECbis implementována ve dvou nezávislých DNS serverech (Bind a NSD 2.x) a čeká se jen na standardizaci. Ta proběhne až v roce 2005, kdy jsou v březnu vydány dokumenty RFC 4033, RFC 4044 a RFC 4035. V říjnu 2005, tedy jen o pár měsíců později, implementuje DNSSEC první TLD doména – švédská .se. Letos v září 2008 došlo k podpisu české TLD .cz. Více informací DNSSEC v české doméně .cz naleznete na stránkách www.dnssec.cz.

Ani DNSSECbis se ovšem neobešel bez problémů. V návrhu protokolu se počítá s tím, že je zapotřebí mít i kryptograficky ověřené odpovědi o neexistenci určitého záznamu. Protože DNSSEC podepisuje DNS záznamy a odpověď o neexistenci doménového jména, je standardně realizována nastavením návratového kódu RCODE na hodnotu NXDOMAIN (viz. předchozí článek o struktuře DNS zprávy). Z tohoto důvodu vznikl DNS záznam NSEC, který pro určité doménové jméno říká, jaký záznam jej abecedně následuje. V případě dotazu na neexistující záznam DNS server vrátí jména, která jsou „okolo" a tato odpověď již může být kryptograficky podepsána. Tento princip ovšem otvírá jeden nepříjemný důsledek – tímto způsobem se dá projít celý doménový prostor, jedná se tzv. zone walk. Návrh řešení existuje v dokumentu RFC 5155 schváleném letos na jaře, který definuje nový druh záznamu NSEC3, kde je informace o dalším záznamu ukryta pomocí hashovacího algoritmu. Další zajímavá vlastnost NSEC3 je tzv. Opt-Out, který specifikuje mechanismus, kdy po přidání nezabezpečené delegace, není nutné přepočítávat (a znovu podepsat) celý řetěz NSEC3 záznamů, což zjednodušuje náročnost správy především u velkých domén. NSEC3 je v současné době podporovaný DNS servery NSD 3 v roli autoritativního serveru a Unbound pro rekurzivní použití. DNS server Bind bude podporu NSEC3 obsahovat od verze 9.6, která je momentálně v přípravě a poslední vydaná verze je 9.6rc2. Tato verze Bind bude také obsahovat jednodušší správu podepsaných domén. Od verze 9.7 by pak správa podepsaných domén měla být ještě výrazně jednodušší.

Dnes se píše rok 2008 a pomocí systému DNSSEC je chráněno pouze pět národních domén: .se, .bg, .pr, .br a od letošního září také doména česká .cz. Nicméně jistý posun lze pozorovat i u domén generických – nařízení administrativy Spojených států, které bylo vydáno letos, ukládá správci domény .gov, aby do konce ledna 2009 podepsal doménu, kterou používají vládní úřady a organizace ve Spojených státech. Tento podzim také Národní telekomunikační úřad Spojených států (NTIA) vydal žádost o komentáře k podepsání kořenové zóny. Dokumenty a komentářů si můžete přečíst na webových stránkách www.ntia.doc.gov/DNS/dnssec.html. Z generických domén je podepsána také doména .museum, ale tento proces proběhl poněkud v tichosti, takže bohužel nelze říci, jestli to byl úmysl nebo omyl. Celkově by se dalo by se říci, že situaci okolo DNSSEC hodně pomohl útok na DNS, který poprvé odhalil BEZPEČNOSTNÍ VÝZKUMNÍK DAN KAMINSKY. A věci se konečně začaly hýbat.

Jak funguje DNSSEC?

Dnes již klasickým způsobem, jak ověřit pravost informací, je digitální podpis. Každý z nás se s digitálním podpisem v nějaké formě na internetu již setkal, ať už se jednalo o přístup na zabezpečené stránky přes HTTPS, digitální podpis v emailu přes X.509 certifikáty nebo OpenPGP. DNSSEC přináší digitální podpis do světa DNS. Důležité je si uvědomit, že rozšíření DNSSEC bylo navrhováno s ohledem na maximální zpětnou kompatibilitu.

DNSSEC klíč

Základním RR typem, o který DNSSEC rozšířil protokol DNS, je DNSKEY. Záznam DNSKEY může vypadat například takto:

```
dnssec.cz.      600 IN DNSKEY 257 3 5 (
AwEAAc4x/KbNECb+dpDDBSvyxfTlvUxXyC3EAqCnXDp4
+IxjfmwCm1QfB/VIMfqQ2bSsEu51BPk/38dBG01COvE5
tYit/Ux8gIuDgZiJx+ldZ9OAJ3Lnm7v/q5+gy2LSzW46
p6U45WHmGnDZ4c3uhzcfooXmQsW4UmIw+zDc2ePADy3M
bkr3Srll3XDny1OHoW6Ch4o8qC+ezzRDSEnhrtpon+r9
4sqXF50k6OLaqCRB3q9iaGUgviTVfZWJIlvZOwvxxpbH
SDd6QThM/CZBzcx/8JHAwP7MJcUQYS8XvBwRdaAfVDuE
FjUj6IF+vgn8PI1ipQUrF8L0OAHf1dHBou1XjuE=
) ; key id = 17398
```

RDATA záznamu DNSKEY obsahují příznaky klíče (257), typ protokolu (3 = DNSSEC), použitý algoritmus (5 = RSASHA1) a data veřejné části klíče (AwEA…juE=). Z DNSKEY klíče se dá získat ještě jeden údaj, který je spíše jen informativní a tím je keytag (nebo také id) klíče – 16-bitové číslo používané pro rychlou identifikaci. Obecné doporučení (nikoli však nutnost) je používat dva druhy klíčů – ZSK (Zone Signing Key) a KSK (Key Signing Key). Z názvů těchto klíčů vyplývá, že první typ bude použit pro podpis samotného obsahu zóny a druhý typ se bude používat pouze pro podepisování klíčů. Toto rozdělení je čistě praktické – výměna klíčů není triviální operace a proto byla i v rámci klíčů zavedena jedné zóny hierarchie.

KSK je klíč, který může být silnější (má větší počet bitů), výsledný podpis je větší, podepisování tímto klíčem je výpočetně náročnější a také validace podpisů vytvořených tímto klíčem je výpočetně náročnější. Proto je tento klíč použit pouze pro vytvoření jediného podpisu v celé zóně a to podpisu všech DNSKEY záznamů. Díky větší síle tohoto klíče může být v zóně publikován a používán delší dobu bez ohrožení bezpečnosti. KSK se od ZSK liší pouze jedním bitem v příznacích klíče (257 je KSK a 256 je ZSK).

ZSK je pak klíč, který je slabší, a používá se pro podpis všech záznamů v zóně (včetně DNSKEY). Protože je klíč slabší, musí být měněn častěji, ale díky hierarchii mezi KSK a ZSK, neznamená výměna ZSK žádnou interakci s dalšími subjekty. V jednom z dalších dílů seriálu si ukážeme, jak a proč je potřeba klíče měnit.

DNSSEC podpis

Nyní si ukážeme další nový RR typ, který je potřeba pro vlastní digitální podpis pomocí DNSSEC – typ záznamu RRSIG. Pokud si ještě vzpomenete na první díl našeho seriálu o technologii DNSSEC, tak jsme hned na začátku mluvili o RRSetech, což jsou všechny takové RR záznamy, které mají všechny údaje kromě RDATA stejné. Termín RRSet je pro DNSSEC důležitý, protože digitální podpis se vytváří pro RRSet a nikoli pro jednotlivé RR záznamy. DNS odpověď s DNSSEC může vypadat například takto:

```
$ dig +norec +multi +dnssec www.dnssec.cz @b.ns.nic.cz

; <<>> DiG 9.5.0-P2 <<>> +norec +multi +dnssec www.dnssec.cz @b.ns.nic.cz
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19348
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
```

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

;; ANSWER SECTION:
www.dnssec.cz.      600 IN A 217.31.205.50
www.dnssec.cz.      600 IN RRSIG A 5 3 600 20090127010003 (
                    20081228010003 58773 dnssec.cz.
                    rkrCtJFuRt+jCuUEnMB8eKO90DEsYXCE8QP5vn1zc1E8
                    r+NS+KVUgicJ4QFdGlb8qoQYCDFE0yVdYYEc2nybn9gQ
                    /cx4rKGRCZ3SAGGFLgjOnip60qI6ESlWDqGu5kvgJPGo
                    wpmXsWqBd1mApd8N9DQGLiRt7U6RsRCshBnbKA4= )

;; AUTHORITY SECTION:
dnssec.cz.      600 IN NS b.ns.nic.cz.
dnssec.cz.      600 IN NS a.ns.nic.cz.
dnssec.cz.      600 IN RRSIG NS 5 2 600 20090127010003 (
                20081228010003 58773 dnssec.cz.
                ZY4WqF4SkEWUaA0Wqgu517q4yy9tZgnJe4r3DATI6ecT
                cXKIMXUjDl6Gc3jZqtw55DWVDEH5lb2jnSglLUMsBRBQ
                dm45b4r+r45x1OyP2Obtg5LjXkmVdQVTqOBmfL3hzUqt
                uoSafDmYVN0HFwqTNVfkaRotaSpvXBNXU43Z1cE= )

;; Query time: 18 msec
;; SERVER: 2001:1488:dada:184::188#53(2001:1488:dada:184::188)
;; WHEN: Sun Dec 28 18:18:21 2008
;; MSG SIZE  rcvd: 435
```

V sekci Odpověď vidíme samotný A záznam a k němu příslušný RRSIG záznam, sekce Autorita obsahuje příslušné DNS servery pro doménu dnssec.cz a podpis tohoto RRSetu. Pokud se podíváme na obsah RDATA v RRSIG záznamu, tak objevíme tyto položky:

Položky v RRSIG záznamu

A	Typ podepsaného záznamu
5	Použitý algoritmus (5 – RSASHA1)
3	Počet labelů podepisovaného doménového jména
600	TTL původního záznamu
20090127010003	Datum konce platnosti podpisu
20081228010003	Datum počátku platnosti podpisu
58773	Keytag klíče použitého pro vytvoření podpisu
dnssec.cz.	Vlastník klíče použitého pro vytvoření podpisu (jméno zóny)
rkrC...KA4=	Digitální podpis

Klientská strana tedy dostala o jeden RR záznam navíc a k čemu je tento záznam dobrý? Pokud máte správně nakonfigurovaný rekurzivní DNS server, aby prováděl validaci DNSSEC podpisů (dále také validující resolver), může tento ověřit, že data nebyla v průběhu transportu změněna nebo kompletně podvržena. Toto je poměrně důležitá informace – validace podpisů se vždy provádí na straně klienta a podpis je vždy předpočítán dopředu, takže samotný DNSSEC nezatěžuje autoritativní DNS servery. Z uživatelského hlediska je obsah záznamu RRSIG nezajímavý – pokud hledáme chybu, tak můžeme zkontrolovat údaje jako jsou datum počátku a konce platnosti, keytag klíče a vlastníka klíče. Ověření platnosti samotného digitálního podpisu není v silách normálních smrtelníků.

V příkladu výše jsme se ptali přímo autoritativního serveru, který neprovádí validaci. Pokud stejný dotaz položíme nakonfigurovanému validujícímu resolveru, bude vypadat malinko jinak:

```
$ dig +multi +dnssec www.dnssec.cz @localhost

; <<>> DiG 9.5.0-P2 <<>> +multi +dnssec www.dnssec.cz @localhost
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61066
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
```

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

;; ANSWER SECTION:
www.dnssec.cz.      600 IN A 217.31.205.50
www.dnssec.cz.      600 IN RRSIG A 5 3 600 20090127010003 (
                    20081228010003 58773 dnssec.cz.
                    rkrCtJFuRt+jCuUEnMB8eKO90DEsYXCE8QP5vn1zc1E8
                    r+NS+KVUgicJ4QFdGlb8qoQYCDFE0yVdYYEc2nybn9gQ
                    /cx4rKGRCZ3SAGGFLgjOnip60qI6ESlWDqGu5kvgJPGo
                    wpmXsWqBd1mApd8N9DQGLiRt7U6RsRCshBnbKA4= )

;; AUTHORITY SECTION:
dnssec.cz.      600 IN NS a.ns.nic.cz.
dnssec.cz.      600 IN NS b.ns.nic.cz.
dnssec.cz.      600 IN RRSIG NS 5 2 600 20090127010003 (
                    20081228010003 58773 dnssec.cz.
                    ZY4WqF4SkEWUaA0Wqgu517q4yy9tZgnJe4r3DATI6ecT
                    cXKIMXUjDl6Gc3jZqtw55DWVDEH5lb2jnSglLUMsBRBQ
                    dm45b4r+r45x1OyP2Obtg5LjXkmVdQVTqOBmfL3hzUqt
                    uoSafDmYVN0HFwqTNVfkaRotaSpvXBNXU43Z1cE= )

;; Query time: 396 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 28 18:41:23 2008
;; MSG SIZE  rcvd: 435
```

V příznacích DNS zprávy ubyl příznak AA (Authoritative Answer), protože odpověď již není autoritativní, a přibyl příznak AD (Authenticated Data). Tento příznak indikuje, že validující resolver ověřil platnost DNSSEC podpisu a data v DNS odpovědi jsou správná a nebyla pozměněna. Tento příznak ovšem dostaneme pouze pokud použijeme volbu +dnssec na příkazové řádce nástroje dig, která v DNS dotazu nastaví příznak DO (DNSSEC OK). Pokud bychom tento příznak nenastavili, tak v DNS odpovědi nepoznáme, že RR záznamy byly validovány. Takto je zajištěna zpětná kompatibilita s klienty, kteří DNSSEC neznají. Možná vás v tuto chvíli napadlo – a jak tedy klient pozná, pokud byla data podvržena a digitální podpis nesouhlasí? V takovém případě se DNS odpověď vrátí s chybovým příznakem (RCode) SERVFAIL a špatná odpověď se ke klientovi vůbec nedostane. Návratový kód SERVFAIL bude nastaven i v případě, že „jen" vyprší časová platnost podpisu. I v takovém případě již není RRSIG podpis platný a nebude validován. Proto je potřeba podpisy v zónovém souboru pravidelně obnovovat.

Pro odlišení normální chyby serveru a chyby ve validaci DNSSEC podpisu byl zaveden speciální příznak CD (Checking Disabled), který nastavuje klient v dotazu na validující resolver. Tento příznak způsobí návrat dat v DNS odpovědi i v případě, že podpis není validní. Více o tom, jak tento příznak použít, si řekneme v některém z dalších dílů našeho seriálu, který bude věnován hledání chyb.

Podpis neexistujícího záznamu

V předchozím odstavci jsme si ukázali, jak vypadá podpis pomocí DNSSEC. Jistě jste si všimli, že podepsané jsou RR záznamy. Pokud si vzpomenete na předchozí díly seriálu, tak jsme si ukazovali, že v případě, že dotazovaný záznam neexistuje, je v DNS odpovědi nastaven návratový kód RCode na hodnotu NXDOMAIN. DNS odpověď, která v sobě neobsahuje žádná data, ovšem nemůže být podepsána. A v tuto chvíli nastupuje další typ RR záznamu – NSEC:

```
www.dnssec.cz.      600 IN NSEC dnssec.cz. A AAAA RRSIG NSEC
www.dnssec.cz.      600 IN RRSIG NSEC 5 3 7200 20090127010003 (
                    20081228010003 58773 dnssec.cz.
                    ZrZv9lLNNFkjnhJ+9gLl9kZUI9LHP9r+qNBqTyJq3gSo
                    7DpnmCI4tNHdpJKM0cKYf7nuZ1vBebNtiBEMPpdv/Z3K
                    MbnF7GWxSOBltvx3cHBa/OHov1ZhPyVyxvE17NvMANo2
                    K0654YZu/o8YsfDeImcQkT/gngclIWEBwV3ly/Y= )
```

NSEC je RR záznam, který ve svých RDatech obsahuje informaci o následujícím záznamu v setřízené zóně a informaci o všech existujících typech pro vlastníka záznamu. Při dotazu na neexistující záznam vrací autoritativní DNS server takový NSEC záznam, který je před a za dotazovaným doménovým jménem v případě kompletní neexistence takového doménového jména, nebo přímo NSEC záznam se shodným vlastníkem v případě neexistence konkrétního typu RR záznamu.

V letošním roce bylo standardizováno rozšíření NSEC záznamu – NSEC3, které odstraňuje jednu kritizovaných vlastností NSEC záznamu: možnost jednoduché iterace celou zónou. O NSEC3 psal nedávno na Lupě Pavel Satrapa ve článku NSEC3 – DNSSEC, který nic nevyzradí.

Haló, já jsem podepsal!

Ve chvíli, kdy je zóna podepsána, musíte nějakým způsobem dát vědět svému okolí, že jste podepsali a pro podpis používáte ten a ten konkrétní KSK klíč. Mohli byste svůj klíč poslat všem subjektům, které by chtěly vaši zónu validovat, nebo použijete hierarchický systém DNS a použitý klíč publikujete do nadřazené zóny. Publikace v nadřazené zóně se děje pomocí posledního nového typu RR záznamu – záznamu o bezpečné delegaci DS (Delegation Signer):

```
dnssec.cz.      1800 IN DS 17398 5 1 (
                BBDDDD272C4D81EF941C722CEF79A848B543502D )
dnssec.cz.      1800 IN RRSIG DS 5 2 1800 20090105140535 (
                20081206140535 4092 cz.
                dN2nO7C3vKDqf1Q0e+Ulijsp8orlYWD95PpjyssHcUAK
                Tya8bkwDz4B86KSyapFO+j6N1dqXRzwxi3dE3IPDKxzO
                pVG+oTTnJZakqLgxEaRf4H69sqcWmImVMPoHEHM/Y/p/
                zXvUPZFZoSQH74ztQYf1XRQ3rP7liEdBO8TOu9o= )
```

DS záznam je hodně jednoduchý. Obsahuje keytag klíče (17398), algoritmus klíče (5 – RSASHA1) a hashovací algoritmus DS záznamu (1 – SHA1). Následuje hash vytvořený z vlastníka DS záznamu a DNSKEY RDATA. Tento záznam je publikován a podpsán v nadřazené zóně – v tomto případě bude tento záznam v zóně národní domény .cz.
DS záznamem končí dnešní část seriálu o DNSSEC. V dalších dílech se konečně přesuneme od teorie k praxi a naučíme se konfigurovat DNS servery Bind, Unbound a NSD.

Konfigurace DNSSEC validujícího resolveru

Protože se v dnešním díle budeme zabývat praktickou konfigurací, je důležité zmínit, jaké DNS servery budeme používat. První DNS server, pro který si ukážeme konfiguraci, bude Bind. Druhým používaným DNS serverem bude nováček na poli DNS serverů – server Unbound. Unbound je rekurzivní DNS server, který vznikl jako alternativa k DNS serveru Bind a je vyvíjen neziskovou organizací NLNet Labs. Najdete jej buď přímo ve své distribuci nebo je možné si stáhnout zdrojové kódy na adrese www.unbound.net. Ke kompilaci je zapotřebí knihovna ldns, která je buď přibalena do zdrojových kódu serveru Unbound, nebo ji najdete na adrese www.nlnetlabs.nl/ldns/. Knihovna ldns v adresáři examples/ obsahuje i sadu ukázkových programů pro práci s DNS a DNSSEC a v adresáři drill/ je program drill, který je možné používat jako alternativu k programu dig. K úspěšnému používání DNSSEC není zapotřebí instalovat ani knihovnu ldns a ani DNS server Unbound – příslušné pasáže ve článku prostě přeskočte.
Technologie DNSSEC je podporována v různých verzích DNS serveru Bind a ve všech stabilních verzích DNS serveru Unbound. Ve všech následujících příkladech konfigurace se budu snažit ukázat konfiguraci pro všechny používané verze, pokud se konfigurace liší.

Zapínáme DNSSEC validaci

Zapnout DNSSEC validaci je ta nejjednodušší část z celého článku. Pokud používáte DNS servery Bind ve verzi 9.5 a vyšší nebo Unbound, nemusíte nic dělat. DNSSEC validace je implicitně zapnuta. Bind ve verzi 9.4 má implicitně zapnutou pouze podporu DNSSEC, ale validace je implicitně vypnuta. Bind ve verzi 9.3 má implicitně podporu DNSSEC vypnutou.
Pokud byste chtěli DNSSEC validaci explicitně zapnout/vypnout, pro server Bind 9.5 a 9.4 se jedná o dvě konfigurační volby. Konfigurační volba `dnssec-enable boolean;` v sekci `options {};` zapíná/vypíná DNSSEC. Neznamená to nic jiného, než že DNS server posílá DNS dotazy se zapnutým příznakem DO (DNSSEC OK). Druhá konfigurační volba `dnssec-validation boolean;` opět v sekci `options {};`zapíná/vypíná DNSSEC validaci a je platná pouze pro Bind verze 9.4 a vyšší. Bind ve verzi 9.3 zná pouze první konfigurační volbu a DNSSEC validace se nedá samostatně vypínat a zapínat.
Pro server Unbound je možné DNSSEC validaci zapnout nebo vypnout pomocí direktivy `module-config: "validator iterator"`. Vypnutí DNSSEC validace se provede nastavením této direktivy pouze na hodnotu, tedy `module-config: "iterator"`.

Důvěryhodné klíče

Validující resolver nyní má povolenu validaci podpisů, nicméně neví, kterým klíčům může důvěřovat a kterým ne. Konfiguraci DNSSEC klíčů je zapotřebí provést ručně. Po ruční konfiguraci klíčů bude možné, aby validující resolver ověřoval podpisy pro všechnya doménová jména, která jsou přímo v zóně nakonfigurovaného klíče, nebo k nim vede bezpečná cesta přes DS záznamy. Tedy, pokud má validující resolver nakonfigurovaný klíč pro zónu .cz a subdoména dnssec.cz zveřejní svůj klíč pomocí DS záznamu, bude moci tento validující resolver ověřovat podpisy i v zóně dnssec.cz.
Konfigurace klíčů je velmi jednoduchá. Pro server Bind se jednotlivé klíče přidávají do samostatné sekce trusted-keys {};. Musíte přidat klíč nebo více klíčů pro každou doménu, pro kterou chcete ověřovat DNSSEC podpisy a zároveň tato doména není bezpečně delegována z nadřazené zóny, pro kterou jste klíč již přidali.
Příklad konfigurace pro zónu .cz bude vypadat takto:

```
trusted-keys {

"cz." 257 3 5
"XXXXAdo9fGLzCyxz1yTlsHCT7JpHrg0q/yOlvDNg39n/gAUzg6H/5X9p
jW6mpecJuZirIcPcRw5E7E8uR8g2ztH4uztoc/7ss01s3rTnEgXfilbd
psEdXEuxIfhq+w6zL6PvCcE3qRSzsrc2//x/SXjWp8yeT4YY3W3kvB4Z
g5ld0a8bAHBYo4ZY9x7a3qnqOhqunXSG8EfRPD9koUMgWCjdnFNR89L1
5Bkzh+q1J7phTHIY5akKf3YnIB/5BnKmGBC7DimK4uSBLiBA3DLxHnvL

ffMT5XtKKHuQ/uZ4IxHWqR2cpHz/6e2WaQvOVILwd0gk9lTCildBGjC7
eNxOMniXXXX=" ; // nepouzivat

};
```

Autoři DNS serveru Unbound se snažili situaci ulehčit, takže Unbound podporuje přesně ten samý formát konfigurace. Sekci trusted-keys {}; uložíte do samostatného souboru a konfigurační direktivou `trusted-keys-file: ""` řeknete, ze kterého souboru má klíče načítat. Unbound podporuje i další direktivy, ale vzhledem k tomu, že se nejčastěji setkáte právě s tímto formátem, nechávám jejich studium za domácí úkol.
Výše uvedený DNSSEC klíč je opravdu jenom příklad a je schválně znehodnocen. Důležité je si uvědomit, že pokud přidáváme klíč do konfigurace DNS serveru, měli bychom ho získat nějakou bezpečnou cestou. Klíče pro zónu .cz můžete získat na stránkách www.nic.cz/dnssec/, kde bude vždy zveřejněn aktuální klíč (klíč v tomto článku zastará), a je možné si ověřit jednak SSL certifikát stránky a jednak PGP podpis DNSSEC klíče. Další DNSSEC klíče je možné získat na webových stránkách sdružení RIPE NCC (klíče pro reverzní delegace a pro ENUM domény), na stránkách švédského doménového registru (klíč pro doménu .se) a na stránkách brazilského doménového registru.
Až dojde k podepsání kořenové zóny, bude možné tyto jednotlivé klíče z konfigurace odstranit a spravovat pouze klíče pro kořenovou zónu. Do doby, než bude kořenová zóna podepsána, je zapotřebí spravovat jednotlivé klíče. Důležité je také se přihlásit do poštovních konferencí, které jsou většinou uvedeny na stránkách s klíči, abyste byli informováni o případných změnách klíče a validace vám nepřestala fungovat. O výměnách klíčů si budeme povídat v následujících dílech seriálu, zatím bych jenom upozornil na to, že pokud máte v konfiguraci špatný klíč, není možné odpovědi od autoritativního DNS serveru správně zvalidovat, a validující resolver vyhodnotí všechny odpovědi jako špatně podepsané a na všechny dotazy z takovéto domény a všech jejich subdomén bude vracet chybový kód SERVFAIL.

Ověření funkčnosti

Nyní máme v konfiguraci zapnutou DNSSEC validaci a přidané klíče pro doménu .cz. Jako další krok je potřeba nezapomenout konfiguraci DNS serveru znovu načíst nebo DNS server restartovat.
Jako poslední krok je zapotřebí ověřit, že jsme vše udělali správně a náš rekurzivní DNS server se stal validujícím resolverem. Jednoduchý příklad jsme si již uvedli v minulém díle, ale není na škodu si ho ukázat znovu:

```
$ dig +multi +dnssec www.dnssec.cz @localhost

; <<>> DiG 9.5.0-P2 <<>> +multi +dnssec www.dnssec.cz @localhost
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61066
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.dnssec.cz.     IN A

;; ANSWER SECTION:
www.dnssec.cz.      600 IN A 217.31.205.50
www.dnssec.cz.      600 IN RRSIG A 5 3 600 20090127010003 (
                    20081228010003 58773 dnssec.cz.
                    rkrCtJFuRt+jCuUEnMB8eKO90DEsYXCE8QP5vn1zc1E8
                    r+NS+KVUgicJ4QFdGlb8qoQYCDFE0yVdYYEc2nybn9gQ
                    /cx4rKGRCZ3SAGGFLgjOnip60qI6ESlWDqGu5kvgJPGo
                    wpmXsWqBd1mApd8N9DQGLiRt7U6RsRCshBnbKA4= )

;; AUTHORITY SECTION:
dnssec.cz.      600 IN NS a.ns.nic.cz.
dnssec.cz.      600 IN NS b.ns.nic.cz.
dnssec.cz.      600 IN RRSIG NS 5 2 600 20090127010003 (
                20081228010003 58773 dnssec.cz.
                ZY4WqF4SkEWUaA0Wqgu517q4yy9tZgnJe4r3DATI6ecT
                cXKIMXUjDl6Gc3jZqtw55DWVDEH5lb2jnSglLUMsBRBQ
                dm45b4r+r45x1OyP2Obtg5LjXkmVdQVTqOBmfL3hzUqt
                uoSafDmYVN0HFwqTNVfkaRotaSpvXBNXU43Z1cE= )

;; Query time: 396 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 28 18:41:23 2008
;; MSG SIZE  rcvd: 435
```

Místo @localhost použijte adresu vašeho resolveru. V příznacích DNS zprávy se musí objevit příznak AD.
Další možnost jak ověřit, že DNS server byl správně nakonfigurován, je dotazem na špatně podepsanou doménu. Příklad takové domény je rhybar.cz.

```
$ dig +dnssec +multi www.rhybar.cz @localhost

; <<>> DiG 9.4.2-P2 <<>> +dnssec +multi www.rhybar.cz @localhost
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 14766
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.rhybar.cz.     IN A

;; Query time: 115 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Jan  4 18:10:10 2009
;; MSG SIZE  rcvd: 42
```

Pro jistotu pak ještě zkontrolujte ten samý dotaz s příznakem CD při dotazování, jestli se nejedná o jinou chybu.

```
$ dig +cd +dnssec +multi www.rhybar.cz @localhost

; <<>> DiG 9.4.2-P2 <<>> +cd +dnssec +multi www.rhybar.cz @localhost
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58454
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.rhybar.cz.     IN A

;; ANSWER SECTION:
www.rhybar.cz.      600 IN A 217.31.205.50
```

```
www.rhybar.cz.      600 IN RRSIG A 5 3 600 20081030080058 (
                    20080930080058 5172 rhybar.cz.
                    XVkut4l9mw2MhodZFIOD2L57AU2u+I6wGVlK1fr6w5lo
                    cFC5NIe8ukw79jYdOCH3WwFgSMscumIz1sGqRPrN/Crh
                    XiU0ymFGFju9x/k10lv6SGS6lslgnZluet04CyibGQ2H
                    BnwTx7qK3j+bNzxKLvjpn7DY9f+YKB8F2FtwNOc= )

                    ;; AUTHORITY SECTION:
rhybar.cz.          14369 IN NS b.ns.nic.cz.
rhybar.cz.          14369 IN NS a.ns.nic.cz.
rhybar.cz.          600 IN RRSIG NS 5 2 600 20081030080058 (
                    20080930080058 5172 rhybar.cz.
                    XNIBK/CmsKZsw6lT2iAa5g+TLOVxPx39N7vOxqW5lafa
                    C56EuCZxUEmZT6ECvU/WzvQIqE1vqN4X6N/Z+5QTXKM3
                    zcT+UhayyiLNRNwdlkmG0xo/+bYACj85lhyB3UGJ+vpR
                    Wg2VVJOy9RLMyIka/S7nVYpgKUZFAZuxue6K17o= )

                    ;; Query time: 5 msec
                    ;; SERVER: 127.0.0.1#53(127.0.0.1)
                    ;; WHEN: Sun Jan  4 18:10:36 2009
                    ;; MSG SIZE  rcvd: 435
```

DLV registr

Protože v současnosti podporuje DNSSEC jen velmi málo TLD registrů a správa individuálních klíčů ve větším množství není reálná, vznikl alternativní způsob správy DNSSEC klíčů pro jednotlivé podepsané domény. DLV registr je podepsaná zóna, která obsahuje speciální DLV záznamy, které jaksi suplují neexistující DS záznamy v nadřazené zóně. Validující resolver, který je nakonfigurovaný pro funkci s DLV registrem, se nejprve podívá do nadřazené zóny, zda tato obsahuje DS záznam, a pokud jej tam nenajde, zkusí se podívat, zdali neexistuje DLV záznam v DLV registru pro doménové jméno, které vznikne spojením původního dotazu a názvu DLV registru. Pokud ano, bude takovýto záznam použit pro ověření důvěryhodného klíče pro ověřovaný podpis. DLV registr technicky může provozovat kdokoliv, ale prakticky se používá pouze DLV registr provozovaný společností ISC – dlv.isc.org. Příklad DLV záznamu v tomto registru by mohl vypadat takto:

```
dnssec.cz.dlv.isc.org.  1800 IN DS 17398 5 1 (
BBDDDD272C4D81EF941C722CEF79A848B543502D )
```

Konfigurace DLV registru není nikterak složitá. Nejprve musíte nakonfigurovat důvěryhodný klíč pro DLV registr (ten najdete na stránkách www.isc.org/ops/dlv/včetně kompletního příkladu konfigurace) a následně do sekce options {}; přidáte direktivu dnssec-lookaside string trust-anchor string;. Výsledná konfigurace bude vypadat takto:

```
options {
dnssec-lookaside "." trust-anchor dlv.isc.org.;
};

trusted-keys {
"dlv.isc.org." 257 3 5 "AwEA...FsJm=";
};
```

Nezapomeňte konfigurace znovu načíst a ověřit, že validace funguje. Ověření můžete například provést tak, že z konfigurace odstraníte klíč pro doménu .cz, a vyzkoušíte příklady uvedené výše.

Konfigurace pro server Unbound vypadá následovně:

```
server:
dlv-anchor-file: <nazev_souboru>
```

V souboru, na který ukazuje konfigurace, jsou obsaženy všechny aktuální klíče ve formátu RR záznamů – buď DS nebo DNSKEY.

Na závěr dnešního dílu bych pro jistotu zopakoval dvě důležité věci – nakonfigurované klíče je zapotřebí získat nějakou bezpečnou cestou a pokud zapnete DNSSEC validaci, je zapotřebí hlídat změny v klíčích. Nejlépe tak, že se přihlásite do poštovní konference dnssec-announce@lists.nic.cz. V příštím díle seriálu o DNSSEC si ukážeme opačnou stranu – konfiguraci autoritativního DNS a podepisování jednotlivých zónových souborů.

DNSSEC na autoritativním serveru

Na začátek bych opět upřesnil, jaké nástroje a DNS servery budeme používat. Již klasicky prvním DNS serverem bude Bind. Druhým serverem, který je možné pro DNSSEC na straně autoritativního serveru použít, je NSD, opět vyvinuté v organizaci NLNet Labs jako alternativa k nejpoužívanějšímu DNS serveru Bind. Oba DNS servery nejspíše najdete přímo ve vaší distribuci. Pokud ne, tak Bind lze stáhnout ze stránek Internet Systems Consorcium (www.isc.org), které jej vyvíjí, a NSD lze stáhnout ze stránek projektu na NLNet Labs.

Pro kompilaci tentokrát není zapotřebí knihovna ldns zmiňovaná v minulém díle, ale je dobré tuto knihovnu a příklady z adresáře examples/ mít nainstalové, protože některé utility lze šikovně využít pro rozšířenou práci se zónovým souborem, případně pro kontrolu podpisu. Důležité je použít minimálně verzi 1.4.0. Předchozí verze obsahovaly chybu v nástroji na podepisování zónového souboru. Knihovnu ldns také možná najdete ve své oblíbené distribuci (v Debianu je to balík ldnsutils), pokud ne, tak na stránkách projektu najdete zdrojové kódy.

Zapnutí DNSSEC na autoritativním serveru

Zapnutí podpory na autoritativním serveru je ještě jednodušší než jeho zapnutí na rekurzivním serveru. Pokud používáte Bind ve verzi 9.4 a vyšší nebo server NSD nemusíte dělat vůbec nic. Podpora je standardně zapnuta. Pouze pro verzi Bind řady 9.3 je zapotřebí v sekci options {}; pomocí konfigurační volby dnssec-enable boolean; explicitně zapnout podporu technologie DNSSEC.

Vygenerování klíčů

Ve čtvrté části našeho seriálu jsme si řekli o dvou druzích klíčů KSK a ZSK. Jak tyto klíče vygenerujeme? Můžeme použít buď nástroje, které jsou standardně distribuovány s DNS serverem Bind, nebo utility z balíku nástrojů ldns.

Pro oba klíče použijeme algoritmus RSASHA1 a jako zdroj náhodných čísel budeme používat /dev/urandom, který se na rozdíl od /dev/random neblokuje při nedostatku entropie. ZSK klíč bude mít 512 bitů (minimum pro RSASHA1 je 512 a maximum 4096) a KSK klíč bude mít 1024 bitů.
Pro vygenerování KSK a ZSK klíčů použijeme tyto příkazy, např. pro zónu dnssec.cz:

```
# dnssec-keygen -a RSASHA1 -b 1024 -r /dev/urandom -f KSK dnssec.cz
dnssec-keygen -a RSASHA1 -b  512 -r /dev/urandom dnssec.cz
```

Parametr -a určuje algoritmus, parametr -b počet bitů, parametr -r zdroj náhodných dat, a nakonec parametr -f KSK říká, že vygenerovaný klíč bude mít v příznacích nastavený SEP (Secure Entry Point) bit, který říká, že klíč je Key Signing Key.
Alternativně lze klíče vygenerovat pomocí utility ldns-keygen:

```
# ldns-keygen -a RSASHA1 -b 512 -r /dev/urandom -k dnssec.cz
ldns-keygen -a RSASHA1 -b 512 -r /dev/urandom dnssec.cz
```

Parametry -a a -b mají shodný význam, pouze parametr pro vygenerování KSK klíč se z -f KSK změnil na pouhé -k.
Oba příkazy na standardní výstup vypíší název vygenerovaného klíče ve formátu K<zona>+<alg>+<keytag> (např. Kdnssec.cz.+005+56944) a na disku vygenerují dva soubory K<zona>+<alg>+<keytag>.key a K<zona>+<alg>+<keytag>.private (soubory Kdnssec.cz.+005+56944.key a Kdnssec.cz.+005+56944.private). V souboru s příponou .key je veřejná část klíče a v souboru s příponou .private je část soukromá, kterou je zapotřebí chránit, a kterou budete potřebovat pro podpis zóny.
Obsah veřejné části klíče vypadá takto:

```
dnssec.cz. IN DNSKEY 256 3 5 AwEAA[...]8Gm0=
```

V případě generování pomocí ldns-keygen obsahuje navíc ještě TTL a v komentáři pak keytag, typ klíče a počet bitů:

```
dnssec.cz. 3600 IN DNSKEY 256 3 5 AwEAA[...]rHE= ;{id = 35181 (zsk), size = 512b}
```

Veřejné části všech klíčů, kterými budeme chtít podepisovat, je zapotřebí přidat do zónového souboru (z mnoha praktických důvodů je dobré mít zónový soubor pojmenovaný stejně jako zóna). Toto je možné udělat dvěma způsoby:
1. Direktivou $INCLUDE <nazev_souboru>, tedy například:

```
$ cat >> dnssec.cz << EOF
$INCLUDE Kdnssec.cz.+005+23163.key
$INCLUDE Kdnssec.cz.+005+56944.key
EOF
```

2. Nebo přímo přidáním obsahu těchto souborů do souboru se zónou:

```
cat Kdnssec.cz.+005+23163.key >> dnssec.cz
cat Kdnssec.cz.+005+56944.key >> dnssec.cz
```

Obsah soukromé části vypadá pro algoritmus RSASHA1 (pro algoritmus DSA je obsah souboru až na první dva řádky odlišný) takto:

```
Private-key-format: v1.2
Algorithm: 5 (RSASHA1)
Modulus: zq0CE[...]bQ==
PublicExponent: AQAB
PrivateExponent: clEr[...]WQ==
Prime1: 6N2ob[...]RpM=
Prime2: 4zVG[...]uv8=
Exponent1: tqJi[...]MhE=
Exponent2: 3h9I[...]LXU=
Coefficient: 3Xz6[...]/1I=
```

Z uživatelského hlediska je obsah tohoto souboru nezajímavý a klidně by obsah mohl být binární.
Podepsání zónového souboru
Nyní máme vygenerované dva klíče připravené pro podpis zónového souboru a pravděpodobně máme i samotný zónový soubor, který chceme podepsat.
Obsah zónového souboru může vypadat například takto:

```
dnssec.cz.  3600    IN SOA  a.ns.nic.cz. hostmaster.nic.cz. 2008101419 10800 3600 1209600 7200
www.dnssec.cz. 3600 IN  AAAA    2001:1488:0:3::2
www.dnssec.cz. 3600 IN  A   217.31.205.50
dnssec.cz. 3600 IN AAAA 2002:1488:0:3::2
dnssec.cz. 3600 IN A    217.31.205.50
dnssec.cz. 3600 IN NS   b.ns.nic.cz.
dnssec.cz. 3600 IN NS   a.ns.nic.cz.
dnssec.cz. 3600 IN DNSKEY 256 3 5 AwEA[...]7VE= ;{id = 40965 (zsk), size = 512b}
dnssec.cz. 3600 IN DNSKEY 257 3 5 AwEAA[...]CZc= ;{id = 42307 (ksk), size = 512b}
```

Příkaz dnssec-signzone má poměrně mnoho parametrů, ale v té nejjednodušší variantě ho můžeme zavolat pouze zavoláním:

```
# dnssec-signzone dnssec.cz
```

Důležité je, aby zónový soubor měl shodný název s názvem zóny, obsahoval veřejné části klíčů, kterými chceme zónu podepsat, a tyto klíče byly v aktuálním adresáři. Zavolání tohoto příkazu způsobí vygenerování nového souboru: dnssec.cz.signed. Tento nový soubor bude obsahovat původní obsah souboru dnssec.cz a navíc bude obsahovat nově vygenerované záznamy NSEC a RRSIG pro každý RRSet:

```
; File written on Sun Jan 11 20:14:32 2009
; dnssec_signzone version 9.5.0-P2
dnssec.cz.      3600    IN SOA  a.ns.nic.cz. hostmaster.nic.cz. (
                                2008101419 ; serial
                                10800      ; refresh (3 hours)
                                3600       ; retry (1 hour)
                                1209600    ; expire (2 weeks)
                                7200       ; minimum (2 hours)
                                )
                        3600    RRSIG   SOA 5 2 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                YEkGZPifd0B3uLER9g7C/zni+j3f7RE0OKJK
                                e6Q3rMnKtwbY6ShZjLQiQHEnvvMmARFOk2Kr
                                FImaDil20jGY/g== )
                        3600    NS  a.ns.nic.cz.
                        3600    NS  b.ns.nic.cz.
                        3600    RRSIG   NS 5 2 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                jW5vlsY4s85s7DV5bEJYuz0CuQqwnrb8kg69
                                WhGy+ftvckH2aQhUWwwnlLJgatxkLmYR2k0+
                                SkR6Wg8ND0fkzA== )
                        3600    A   217.31.205.50
                        3600    RRSIG   A 5 2 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                YPBL0dEVelyBpTzEb/HEratPRNeaR5ncItcm
                                nYPXLqmwmXg20KVygDU+XhYFUnj3CgcJTZMj
                                q0Bm20keYBCOCQ== )
                        3600    AAAA    2002:1488:0:3::2
                        3600    RRSIG   AAAA 5 2 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                ibdbXaCCfstnWPKK0xzrG+Su76xMwt2zR9oh
                                KrwfjBEYBC28i5iABwS3fa2s5FzBdAh62Y3T
                                xqIp2pCOJGmI/w== )
7200    NSEC    www.dnssec.cz. A NS SOA AAAA RRSIG NSEC DNSKEY
                        7200    RRSIG   NSEC 5 2 7200 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                ydJOiwN0Xj0DBLb4uIVkYlS24AiNIEAm1Ltk
                                lXM9CaRahmPX/1jC/vw6Um9tg0XO8mlbH7yn
                                wCUA+J6e5mARbg== )
                        3600    DNSKEY  256 3 5 (
                                AwEAAc/dbg0TRovNLCAEEUcSJnN4kxbGulUD
                                6XvkLWDRV+r2i7e0JqAlWU6tOJjwF9tE17If
                                TOnUtauBw40nazdk7VE=
                                ) ; key id = 40965
                        3600    DNSKEY  257 3 5 (
                                AwEAAb390x3z9UOpmpiYayEsbQ1/svbboJ5L
                                kttKljfFiMZY7T2dYQb+dLPDOkHvIjWu7ILc
                                EUewaolsKnz4vNy3CZc=
                                ) ; key id = 42307
                        3600    RRSIG   DNSKEY 5 2 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                FvK7y7v2/CA91ooU4cYKgN4sXQF/yFYhbZP+
                                eCPEwKEnBSddziNqfV65j7mtesXdDUHMIqkZ
                                e1IuhpfijCmmrA== )
                        3600    RRSIG   DNSKEY 5 2 3600 20090210181432 (
                                20090111181432 42307 dnssec.cz.
                                cGBcC2ctsU42um1oAOuqowTJ9KbbhbZUVhoL
                                RSAP0kQRfY6VUt/WpRHDAthC9km2oEJCBolN
                                9mZB3B9l951iRA== )
                www.dnssec.cz.      3600    IN A    217.31.205.50
                        3600    RRSIG   A 5 3 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                Urb1gJ7fnwvRFz064jfgccmACrG/CX9o5Y4f
                                ga/97WufyLFwxmPPxn5PLCSQAf1PzFkkP7Sb
                                NMU1jX4/EPpISA== )
                        3600    AAAA    2001:1488:0:3::2
                        3600    RRSIG   AAAA 5 3 3600 20090210181432 (
                                20090111181432 40965 dnssec.cz.
                                JjcqL5SYD6vGJuMXP0BQIzA5rRAfggd0sXT7
                                tElex1xZyx3tJ8euBeTdx0szX4CA0TCWhntF
                                AsKshOCfOw++cw== )
                        7200    NSEC    dnssec.cz. A AAAA RRSIG NSEC
                        7200    RRSIG   NSEC 5 3 7200 20090210181432 (
                                20090111181432 40965 dnssec.cz.
```

```
QElCh1q5zY1MIzwMpqnEpJ9k+fleQncCTbw4
XJn2xtkKAWZhEr2nx9JABO8MWF1deMuWSaFP
qAQRgEchw/43yA== )
```

Nyní si uveďme trochu maximalističtější variantu tohoto příkazu (argumenty jednotlivých parametrů jsou zároveň standardní hodnoty):

```
# dnssec-signzone -s +-3600 -e +2592000 -o dnssec.cz -f dnssec.cz.signed \
-r /dev/urandom -a -k Kdnssec.cz.+005+42307.key -N keep dnssec.cz \
Kdnssec.cz.+005+40965.key
```

Parametry -s a -e určují datum platnosti nově vygenerovaných podpisů a jejich argumenty můžou být buď absolutní ve formátu YYYYMMDDhhmmss nebo relativní se znakem + na začátku, parametr -o určuje název (origin) podepisované zóny, parametr -f určuje název výstupního souboru, parametr -r /dev/urandom jsme již používali při generování klíčů, parametr -a ověřuje platnost nově vygenerovaných podpisů. Důležitý je parametr -k Kdnssec.cz.+005+42307.key, který určuje KSK klíč, který má být použit pro podpis DNSKEY záznamů (v případě, že zónový soubor obsahuje více KSK klíčů).
Další zajímavý a užitečný parametr je -N keep, který určuje jaký formát má mít sériové číslo výstupního souboru. Kromě keep lze použít argument increment, který sériové číslo zvýší, nebo unixtime, který ho nastaví na aktuální unixový čas (počet sekund od začátku roku 1970). Oba dvě tyto alternativní volby se snaží předcházet problému, kdy je zóna považována za nezměněnou ve chvíli, kdy nedojde ke zvýšení sériového čísla. Následuje už jenom původní podepisovaný zónový soubor a teprve po něm jsou vypsané všechny ZSK klíče, které budou použity pro podpis všech RR záznamů v zónovém souboru.
Ekvivalent tohoto příkazu z balíku ldns vypadá takto:

```
# ldns-signzone -i 20090111193000 -e 20090110192959 -f dnssec.cz.signed \
-o dnssec.cz dnssec.cz Kdnssec.cz.+005+42307 Kdnssec.cz.+005+40965
```

Příkazy jsou hodně podobné, jen u začátku (-i) a konce (-e) platnosti vygenerovaných podpisů nelze použít relativní formát. Nástroj ldns-signzone také neumí manipulovat se sériovým číslem. KSK a ZSK klíče není potřeba na příkazovém řádku rozlišovat, dojde k automatické selekci dle nastaveného SEP bitu (256 vs. 257 v příznacích klíče).
Všimněte si, že výsledný soubor obsahuje jeden RRSIG záznam pro každý RRSet kromě RRSetu pro DNSKEY záznamy. Ten má podpisy dva – jeden podpis je pomocí KSK klíče a druhý pomocí ZSK klíče.
Použití podepsaného zónového souboru
Nyní máme vygenerovaný podepsaný zónový soubor. Co s ním? Aby mohl váš DNS server začít vracet podpisy pro dotazy, které mají nastavený DNSSEC OK (DO) příznak, musíte změnit název souboru v konfiguraci. Pro DNS server Bind je zapotřebí změnit direktivu:

```
zone "dnssec.cz" {
type "master";
file "dnssec.cz";
};
```

na

```
zone "dnssec.cz" {
type "master";
file "dnssec.cz.signed";
};
```

A novou konfiguraci načíst pomocí příkazu rndc reload. Po kontrole log souborů, že všechno proběhlo v pořádku, si můžete pogratulovat – právě jste začali poskytovat svou první DNSSEC podepsanou zónu.
Dnešní díl na tomto místě končí. V příštím díle si povíme něco o tom, jak se o podepsané domény starat.

Údržba DNSSEC klíčů

Vytvoření řetězu důvěry

Nyní máme svou zónu podepsanou, ale nikdo o tom, že jsme zónu podepsali, neví a validující resolvery nemají informaci o tom, jaký je správný klíč k této zóně. Krok, který je zapotřebí udělat jako další, spočívá v tom, že dáte všem stranám, kterým chcete, aby vaši zónu validovaly, informaci o vašem novém klíči. Existují tři způsoby, jak to udělat:
Klíč jim přímo nějakým způsobem předáte. E-mailem, poštou, telefonicky. Tato metoda není příliš flexibilní, a v podstatě ji má smysl používat jen ve speciálních případech, jako jsou například doménové registry.
Druhý způsob je možný pouze v případě, že nadřazená zóna podporuje DNSSEC. Při podpisu zóny vznikne na disku soubor s názvem dsset-<zona> a keyset-<zona>. V souboru dsset-<zona> jsou vypsány DS záznamy, které je zapotřebí umístit do nadřazené zóny. V souboru keyset-<zona> jsou pak vypsány KSK klíče, které byly použity pro podpis zóny. Obecně se dá říct, že jsou tyto dva soubory ekvivalentní. DS záznamy se vypočítávají z klíče a názvu zóny, a výsledný hash je umístěn do nadřazené zóny. Pro domény v zóně .cz je možné DS záznamy umístit do zóny .cz tak, že se obrátíte na svého registrátora, který musí podporovat DNSSEC, a zašlete mu KSK klíče. Systém si pak již ke každé doméně, která je takovýmto klíčem podepsána, spočítá DS záznamy a vygeneruje je do zóny .cz.
Třetí způsob je možný pro domény, jejichž nadřazená zóna DNSSEC nepodporuje. Jako dočasné řešení vznikl mechanismus, který se jmenuje Domain Lookaside Validation – DLV. Obecně funguje tak, že místo DS záznamu se do doménového stromu umisťuje DLV záznam, který se skládá z názvu domény a názvu DLV registru, a validující resolver pak tento záznam může použít v případě neexistujícího DS záznamu v nadřazené zóně. Mechanismu DLV bych se věnoval v jednom z následujících článků.

Údržba podepsané zóny

Údržba podepsané zóny se skládá ze údržby RRSIG podpisů a údržby klíčů.

Údržba podpisů

Pokud si vzpomenete na záznam RRSIG, tak si možná uvědomíte, že tento záznam v sobě obsahuje dobu platnosti, od a do kdy je podpis platný. To implikuje dvě věci: Jednak je zapotřebí, aby čas na rekurzivních i autoritativních serverech byl v pořádku a vygenerované podpisy měly platné datum, a na druhé straně pak nedocházelo k chybám ve validaci, a jednak, že je zapotřebí podpisy pravidelně obnovovat a to minimálně před dobou konce platnosti mínus největší TTL RRSIG záznamu v zóně.

Údržba klíčů

DNSSEC klíče jsou veřejně dostupné ve formě DNSKEY záznamů a zároveň jsou dostupná data vygenerovaná při podpisu RR záznamů. Je to v podstatě analogická situace k webovým certifikátům. Pokud jste někdy s certifikáty přišli do styku, tak víte, že certifikát má nějakou platnost a je zapotřebí ho pravidelně měnit. Není za tím jen ekonomický zájem certifikačních autorit, ale čím déle je certifikát veřejně dostupný, tím větší je riziko, že jej někdo rozlouskne. Podobně je to i u DNSSEC klíčů s jediným rozdílem. Žádná položka ve veřejné části certifikátu neříká, jakou má klíč platnost, technicky vás tedy nic nenutí klíče měnit. Nicméně je dobré klíč čas od času vyměnit.

Proč ‚čas od času'? Protože konkrétní čas není pevně daný, ale měl by reflektovat důležitost podepsaných dat. Je jasné, že klíče k soukromé doméně, kde provozujete blog pro své kamarády, nebude zapotřebí měnit tak často, jako klíče od zpravodajského portálu, televize, burzy, banky nebo jiné externí autority, kde je důvěryhodnost informací, které proudí ze služeb provozovaných na konkrétním doménovém jméně a které koncoví uživatelé zasílají, velmi důležitá. A to ze dvou důvodů. Jednak klíče k nedůležité doméně asi nikdo prolamovat nebude, bylo by to plýtvání prostředky, a jednak v případě prolomení klíče nebude napáchaná škoda, tak velká. Důležité si je také uvědomit, že prolomení nebo zcizení klíče ještě automaticky neznamená, že útočník může podvrhovat DNS odpovědi. Technicky se tímto pouze dostane do situace před nasazením technologie DNSSEC – ještě stále musí nějakým způsobem podvrhnout DNS zprávu s podepsanými RR záznamy.

Výměna klíčů v zóně (rollover)

RRSIG záznamy a DNSKEY záznamy mohou být (a budou) ve vyrovnávací paměti uloženy nezávisle na sobě. Proto je zapotřebí dbát na to, abychom se nedostali do situace, kdy ve vyrovnávací paměti resolveru bude RRSIG záznam, ke kterému už neexistuje DNSKEY v zóně na autoritativním serveru. Stejně tak naopak je zapotřebí zajistit, aby ve vyrovnávací paměti resolveru „nezbyl" RRset DNSKEY záznamů, a v zóně na autoritativním serveru už budou RRSIG záznamy nového klíče. Pokud by došlo k jedné z těchto situací, došlo by k přerušení řetězu důvěry a podpisy by nebyly validní – resolver by místo odpovědí, které byly validovány, začal vracet DNS odpovědi s chybovým kódem SERVFAIL.

Teorii máme za sebou pojďme k samotnému mechanismu výměny klíčů. Způsoby výměny DNSSEC klíčů jsou dvě – před-publikace a dvojitý podpis. Jak jsme si řekli výše, oba dva způsoby musí zajistit, aby nedošlo k přerušení řetězu důvěry. Obě dvě metody se dají použít pro výměnu KSK i ZSK klíčů, ale z praktických důvodů se metoda před-publikace používá pro výměnu ZSK klíčů a metoda dvojitého podpisu pro výměnu KSK klíčů.

Před-publikovaní klíčů

Samotný název metody naznačuje způsob, jakým bude výměna klíčů probíhat. Nový DNSSEC klíč je přidaný do zóny nějakou dobu předtím, než bude použit pro podepisování zóny, a teprve po uběhnutí této lhůty dojde k tomu, že jej bude možné začít použít pro podepisování. Tato metoda do zóny nepřidává žádné nové podpisy. Dojde pouze k přidání jednoho RR záznamu DNSKEY.

Krok po kroku bude výměna vypadat následovně:

1. vygenerujeme nový ZSK klíč:

```
dnssec-keygen -a RSASHA1 -b  512 -r /dev/urandom dnssec.cz
Kdnssec.cz.+005+02271
```

2. nově vygenerovaný klíč přidáme do zónového souboru:

```
cat >> dnssec.cz << EOF
;; KSK klice
$INCLUDE Kdnssec.cz.+005+42307.key

;; ZSK klice
$INCLUDE Kdnssec.cz.+005+02271.key
$INCLUDE Kdnssec.cz.+005+40965.key
EOF
```

3. podepíšeme zónu s explicitním uvedením klíčů, které chceme použít (42307 a 40965 jsou klíče z minulého článku):

```
dnssec-signzone -r /dev/urandom -k Kdnssec.cz.+005+42307.key \
-N increment dnssec.cz Kdnssec.cz.+005+40965.key
```

4. po podepsání zónového souboru je zapotřebí počkat, až se zóna rozdistribuuje na všechny nameservery a následně počkat dobu, která je rovná nebo vyšší TTL záznamů DNSKEY (a jejich podpisů).
5. nyní máme jistotu, že o novém ZSK klíči budou vědět všechny validující resolvery, a můžeme zónu podepsat novým klíčem:

```
dnssec-signzone -r /dev/urandom -k Kdnssec.cz.+005+42307.key \
-N increment dnssec.cz Kdnssec.cz.+005+02271.key
```

6. v této fázi musíme opět počkat. Doba čekání musí být zdola omezená maximálním TTL v celé zóně a shora omezená minimální dobou platnosti ze všech RRSIG záznamů v zóně.
7. po uplynutí této doby můžete ze zónového souboru odstranit odkaz na starý ZSK:

```
cat >> dnssec.cz << EOF
;; KSK klice
$INCLUDE Kdnssec.cz.+005+42307.key

;; ZSK klice
$INCLUDE Kdnssec.cz.+005+02271.key
EOF
```

8. a zónu znovu přepodepsat:

```
dnssec-signzone -r /dev/urandom -k Kdnssec.cz.+005+42307.key \
-N increment dnssec.cz Kdnssec.cz.+005+02271.key
```

Drobná varianta této metody spočívá v tom, že máme v zóně umístěné vždy dva klíče – aktivní a pasivní, tedy až po bod 4. jsme vše již provedli na začátku. Vygenerování nového ZSK klíče a jeho přidání do zóny, pak proběhne na konci celého cyklu místo na jeho začátku.

Dvojitý podpis

Tato metoda se od před-publikování liší v tom, že budeme pro podpis zónového souboru používat oba dva klíče – starý i nově vygenerovaný. Název „dvojitý" podpis je poněkud zavádějící, protože klíčů může být i více, ale pro jednoduchost zůstaneme u tohoto názvu a budeme si předvádět použití této metody pouze na dvou klíčích. Tato metoda je časově rychlejší, ale pokud bychom ji používali pro výměnu ZSK, neúměrně by nafoukla zónový soubor. Proto je vhodná pro výměnu KSK klíčů, kdy dochází jen k dvojitému podpisu RR záznamu nesoucího informaci o používaných klíčích – DNSKEY. Pro výměnu KSK lze použít také metodu před-publikování, ale z důvodů jednoduchosti a rychlosti je vhodnější metoda dvojitého podpisu.

Jak tedy postupovat při výměně KSK:

1. vygenerujeme nový KSK:

```
dnssec-keygen -a RSASHA1 -b 1024 -r /dev/urandom -f KSK dnssec.cz
Kdnssec.cz.+005+60787
```

2. nově vygenerovaný klíč přidáme do zónového souboru:

```
cat >> dnssec.cz << EOF

;; KSK klice
$INCLUDE Kdnssec.cz.+005+42307.key
$INCLUDE Kdnssec.cz.+005+60787.key

;; ZSK klice
$INCLUDE Kdnssec.cz.+005+40965.key
EOF
```

3. podepíšeme zónu s explicitním uvedením obou KSK klíčů – nového i starého, které chceme použít (42307 a 40965 jsou klíče z minulého článku):

```
dnssec-signzone -r /dev/urandom -k Kdnssec.cz.+005+42307.key \
-Kdnssec.cz.+005+60787.key -N increment dnssec.cz \
Kdnssec.cz.+005+40965.key
```

4. nyní nastává důležitá fáze. Nyní je zapotřebí počkat, až se zóna rozdistribuuje na všechny podřízené nameservery plus doba TTL RR záznamu DNSKEY. Souběžně s tím je zapotřebí poslat nový KSK klíč do nadřazené zóny, a to stejným postupem, jakým jste do nadřazené zóny umístili původní KSK klíč – obrátit se na správce nadřazené zóny. V případě domény ze zóny .cz, je zapotřebí se obrátit na svého registrátora. Po té, co je klíč v nadřazené zóně ve formě DS záznamu umístěn, je zapotřebí počkat, až vyprší TTL tohoto DS záznamu.

5. ve chvíli, kdy už všechny validující resolvery mají ve vyrovnávací paměti nové DS i DNSKEY záznamy, můžeme odstranit původní KSK klíč:

```
cat >> dnssec.cz << EOF
;; KSK klice
$INCLUDE Kdnssec.cz.+005+60787.key

;; ZSK klice
$INCLUDE Kdnssec.cz.+005+40965.key
EOF
```

6. a zónový soubor podepsat pouze novým KSK klíčem:

```
dnssec-signzone -r /dev/urandom \
-Kdnssec.cz.+005+60787.key -N increment dnssec.cz \
Kdnssec.cz.+005+40965.key
```

V tomto díle seriálu jsme si ukázali jednu důležitou věc – je zapotřebí se o svoje domény starat. A to ostatně platí i v případě domén, které nemají s technologií DNSSEC nic společného. V příštím díle seriálu si ukážeme, jak hledat chyby pomocí dostupných nástrojů.

DNSSEC: I have no root and I must scream

Po podpisu kořenové zóny bude stačit v konfiguraci validujícího resolveru mít pouze klíč pro kořenovou zónu. V ideální případě všechny ostatní domény budou podepsány tak, že k použitým klíčům povede řetěz důvěry od kořenového klíče. V současnosti však kořenová zóna podepsána není a proto je zapotřebí tuto situaci nějakým způsobem vyřešit. Pokud pomineme možnost DNSSEC validaci vůbec nepoužívat, tak nám zbývají tři cesty, jak vytvořit řetěz důvěry.

Způsob první – Jednotlivé klíče

Způsob první už jsme si ukázali v předchozích článcích. Do konfigurace validujícího resolveru přidáme jednotlivé klíče – pevné body důvěry – pro všechny domény (resp. ostrovy důvěry), které chceme validovat. Tento způsob je nejjednodušší, lze jej použít i na doménová jména, která nejsou veřejná, a administrátor má věci pevně pod kontrolou. Z výhod plynou i nevýhody – je zapotřebí se o individuální klíče pečlivě starat. Pokud by administrátor zapomněl nakonfigurovat nové klíče v případě jejich výměny na autoritativním serveru, došlo by k zneplatnění všech podpisů a doména by se jevila jako nedostupná. Tento systém je také velmi nepraktický – pokud bychom přidávali klíč pro každé doménové jméno, brzy by mohlo dojít k tomu, že bychom v konfiguraci validujícího resolveru měli klíčů příliš mnoho a údržba by přestala být zvladatelnou.

Způsob druhý – Domain Lookaside Validation

S druhým způsobem přišla společnost ISC – autor DNS serveru Bind. Mechanismus DLV funguje tak, že se validující resolver podívá „bokem". V praxi to funguje tak, že nejprve se validující resolver snaží vytvořit řetěz důvěry mezi podpisem a nakonfigurovanými pevnými body důvěry – tedy hledá DS záznam v nadřazené zóně. Pokud DS záznam není v nadřazené zóně nalezen, připojí ke jménu validované domény název DLV registru a hledá DLV RR záznam pro tuto nově vzniklou kombinaci.

V případě nálezu DLV záznamu je výsledek dotazu použit stejně, jako kdyby měl validující resolver k dispozici DS záznam v nadřazené zóně.
Komunikace validujícího resolveru s DLV registrem může vypadat například takto:
1. Zóna .cz je podepsaná a v DLV registru
2. Je vytvořen DNSSEC dotaz na A záznam jména www.dnssec.cz a je poslán kořenovým nameserverům
3. Není nalezen DS záznam v kořenové zóně pro zónu .cz
4. Validující resolver se zapnutou DLV validací bude hledat DLV záznam cz.dlv.isc.org

```
;; ANSWER SECTION:
cz.dlv.isc.org.      3319 IN DLV 7978 5 1 (
9B6C3898470914CDDA98D0CC001688CB32C17A09 )
cz.dlv.isc.org.      3319 IN RRSIG DLV 5 4 3600 20090228094531 (
20090129094531 9912 dlv.isc.org.
GyjKws8ZveMSbDqL/DboeC7MHQmiqTiZDijMoaZd3B2k
n9UxJPXlG9TPxDH2Ve5EgNOEPIV1Bm5YUzJ64JQFwF/l
iIBsAMIFu5k41hj5oODsTuLryrUcKMBMCgJfJWBDxi02
BXcRxfXUbbbCmI5fwEOFRsgDar/aKB6iVW3/Weg= )
```

5. Tento DLV záznam bude použit jako DS záznam pro zónu cz
6. Je vytvořen DNSSEC dotaz na A záznam jména www.dnssec.cz a je poslán autoritativním serverům pro zónu .cz
7. V zóně .cz je nalezen DS záznam pro zónu dnssec.cz
8. Tento DS záznam je použit pro validaci podpisu A záznamu v zóně dnssec.cz.

Konfigurace DLV registru pro DNS server Bind 9.3.3 a vyšší vypadá takto:

```
options {
// zapneme DLV validaci
dnssec-lookaside . trust-anchor dlv.isc.org.;

};
trusted-keys {
dlv.isc.org. 257 3 5
"BEAAAAPHMu/5onzrEE7z1egmhg/WPO0+juoZrW3euWEn4MxDCE1+lLy2
brhQv5rN32RKtMzX6Mj70jdzeND4XknW58dnJNPCxn8+jAGl2FZLK8t+
1uq4W+nnA3qO2+DL+k6BD4mewMLbIYFwe0PG73Te9fZ2kJb56dhgMde5
ymX4BI/oQ+cAK50/xvJv00Frf8kw6ucMTwFlgPe+jnGxPPEmHAte/URk
Y62ZfkLoBAADLHQ9IrS2tryAe7mbBZVcOwIeU/Rw/mRx/vwwMCTgNboM
QktUdvNXDrYJDSHZws3xiRXF1Rf+al9UmZfSav/4NWLKjHzpT59k/VSt
TDN0YUuWrBNh";
};
```

V sekci options řekneme, od jakého místa v DNS hierarchii se má použít alternativní validace přes DLV registr a jaký DLV registr budeme používat. V současné době je v provozu pouze jeden DLV registr – provozuje ho společnost ISC na adrese dlv.isc.org. Aby mohly být DLV záznamy považovány za důvěryhodné, je zapotřebí, aby k nim také vedla cesta přes řetěz důvěry. Proto je zapotřebí do konfigurace přidat aktuální klíč DLV registru. Pozor! Klíč použitý v příkladu byl aktuální v době psaní tohoto článku, a tento klíč se v pravidelných intervalech mění. Platný klíč vždy najdete na adrese secure.isc.org/ops/dlv/. Na stejné adrese také najdete podrobný popis toho, jak používání DLV registru nakonfigurovat.
Pro DNS server Unbound 1.1.0 a vyšší je konfigurace ještě jednodušší, ze stránek secure.isc.org/ops/dlv/ si stáhneme aktuální klíč pro zónu dlv.isc.org:

```
$ wget -O /etc/unbound/dlv.isc.org.key http://ftp.isc.org/www/dlv/dlv.isc.org.key
```

Na výše zmíněných stránkách je dostupný i PGP podpis tohoto klíče a je velmi vhodné si jej ověřit.
Do konfigurace unbound.conf pak přidáme jednu řádku v sekci server:

```
# dlv-anchor-file: "/etc/unbound/dlv.isc.org.key"
```

Načteme novou konfiguraci a DLV validace je tímto zapnuta.
Velká výhoda DLV registru je množství domén, které začnete validovat ve chvíli, kdy DLV registr začnete používat. Mezi nevýhody bych uvedl – žádná kontrola nad tím, jaké klíče chcete nebo nechcete validovat, a provoz jediného DLV registru není v rukou autorit, které mají na starosti správu kořenové zóny, ale konsorcia ISC. DLV registr také není příliš škálovatelný – DLV klíče sice lze rozdělit do více podzón, neexistuje automatizované rozhraní pro přidávání/měnění/mazání klíčů z DLV registru. I přes to všechno je DLV registr dobrým způsobem, jak začít s validací DNSSEC podpisů – v ideálním případě musíte hlídat pouze jeden klíč v konfiguraci. Bohužel v DLV registru nejsou ani všechny podepsané TLD domény – např. švédská doména .se v DLV registru uložena není, a pokud ji chcete validovat, musíte přidat její klíč do konfigurace ručně.

Způsob třetí – IANA

Třetí způsob v sobě zahrnuje celkem dvě metody – nicméně obě jsou provozovány organizací IANA, která je (zjednodušeně řečeno) zodpovědná za obsah kořenové zóny. IANA již nějakou dobu provozuje testovací prostředí na podepisování kořenové zóny a toto testovací prostředí je veřejně přístupné. Naleznete jej na adrese ns.iana.org/dnssec/status.html. Na těchto stránkách lze také najít ukázkovou konfiguraci pro Bind 9:

```
options {
...
dnssec-enable yes;
dnssec-validation yes;
...
};

zone "." {
```

```
type hint;
file "dnssec.root";
};

trusted-keys {
"." 257 3 5 "AwEAAff8EiNa/S3wovNzPUmuBqe1pSjnNoen
cXDNMpmjTgngGMPct+8KDKxM6FwvPSRx15gN
RyRQfzSPU0WshDNkBV2TMtVpzqn/dsurbmTo
ixRzLyLK2Kd2adg5o5yS/gaTgCo0HVBmIruS
N3FVI2ugCWJBFLkFGHLvMJ0BTSYVqWGwQIzp
EPKCbKN+L9nrLcvJRCWG59Yq6BUsSEKlzSK3
jMhYQs6y5IiCGAVol+3VyjN93/lXkeUG6u7d
lQsyiY9fxfeUvmn004y0TjAgjZqdwKZB0K9M
A7qcALG3Tw2TXEdQsn9aY3DzNii3YEBidzER
mY7n4hIUri1r59MnuNJq2x0=";
"." 257 3 5 "AwEAAb+qUOkdZKCP0Qn/4TxJy2XD07xOckKj
wwAHOE/Hn3rLGy0RpmVYCOrmJbVtfyC6i8SQ
sRRKj6YUVlNg7EJ9gjK6rTiDlYMxSc0hFsoG
I8qfAfmsjwClVh86rSIJvruvbcRsQURp/gvJ
EdOaEHlA3JEIHS3cRR5AjKeF1IQdsGYtJMBM
2VMtrgHKgOPZjzfM6bPyg+H9uMBwOm2HkSiE
geAw2vXEHp0eNM2sOxUQMYYPkoywa28oxP4v
dUI7ht4I8etlq3gNCEuBJjV4347ZQ0VHIwDw
JSVmYBzc4f3REfEZS7h6fR33wPVQIw9NNi9p
Cy7JRqzEGwHVft/re6SRqE0=";
};
```

Soubor dnssec.root pak bude obsahovat následující obsah:

```
. 3600 IN NS ns.iana.org.
ns.iana.org 3600 IN A 208.77.188.32
```

Touto konfigurací způsobíme přepsání konfigurace kořenových nameserverů na pouhý jeden. V případě výpadku toto serveru by (po vypršení záznamů ve vyrovnávací paměti resolveru) přestalo DNS být dostupné. Proto je tato konfigurace vhodná pouze pro testovací účely.

Žhavou novinkou chladného měsíce ledna je spuštění beta verze Interim Trust Anchor Repository na adrese itar.iana.org. ITAR je dočasné úložiště klíčů pro kořenovou zónu do doby než bude podepsaná kořenová zóna a DS záznamy budou uloženy přímo v ní. Z toho plynou určitá omezení – klíče do ITARu můžou vkládat pouze TLD operátoři. V současné době je ITAR v testovacím provozu a obsahuje klíče pro tři TLD – švédskou, brazilskou a také tu naši českou. Použití klíčů z ITAR registru pro DNS server Bind je v současnosti poněkud krkolomné, protože ITAR negeneruje soubor ve formátu trusted-keys, tudíž jeho použití znamená trochu toho skriptování na konverzi z DS záznamů do konfigurace Bindu. Master soubor je poskytován i ve speciálně naformátovaném souboru XML, který bude velmi snadné přes XSLT transformaci dostat do požadovaného formátu ve formě trusted-keys {}; direktivy.

Unbound podporuje formát souboru anchors.mf. Tento soubor je ke stažení na výše uvedených stránkách (včetně synchronizace přes protokol rsync):

```
$ rsync -rvP rsync://rsync.iana.org/itar/ /etc/unbound/itar/
$ cd /etc/unbound/itar/ && gpg --verify anchors.mf.sig anchors.mf
gpg: Signature made Sun 01 Feb 2009 01:45:06 AM CET using DSA key ID 81D464F4
gpg: Good signature from "IANA Trust Anchor Repository <itar@iana.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1642 571A 578F 0DF2 6F82  785F F47D FB30 81D4 64F4
```

Zapnutí validace se provede následující direktivou:

```
trust-anchor-file: "/etc/unbound/itar/anchors.mf"
```

Opět i o této službě platí, že je zatím vhodná k použití pouze v testovacím provozu. V případě použití v produkčním prostředí bych doporučil spíše ruční kontrolu nově stažených klíčů a následné úpravy v konfiguraci DNS serverů dělat také ručně. Nicméně předpokládám, že v průběhu první poloviny tohoto roku dojde ke stabilizaci a spuštění repozitáře ITAR do produkce.

To je pro dnešek vše a v příštím díle seriálu o technologii DNSSEC se dostaneme k již avizovanému hledání problémů v DNSSEC validaci.

Hotovo: DNS je podepsáno

Přibližně v roce 2007 začínají sílit tlaky na podepsání kořenové zóny. Pracovní skupina DNS organizace RIPE sdružující evropské ISP, poskytovatele obsahu a další internetové subjekty, se na konferenci RIPE 54 v Talinnu domlouvá, že naformuluje otevřený dopis vyzývající k podpisu kořenové zóny. Tento otevřený dopis je v červnu 2007 zaslán na ICANN a obsahuje opatrnou formulaci s žádostí o podpis kořenové zóny v realistickém termínu. Nezávisle na této výzvě spouští ICANN ten samý měsíc skrze servisní organizaci IANA testovací provoz podpisu kořenové zóny. Tento testovací provoz je v tuto chvíli stále v provozu, nicméně je nutné podotknout, že se skutečným podpisem kořenové zóny toho nyní má jen velmi málo společného, a jakékoli využití kromě ukojení vlastní zvědavosti nedoporučuji.

Stejně jako boží mlýny i ICANN mele pomalu ale jistě, což je ostatně v případě této organizace dobře, a o rok později v červnu 2008 vydává dokument DNSSEC @ ICANN, ve kterém popisuje svůj záměr podepsat kořenovou zónu. Na tento krok následně navazuje Ministerstvo obchodu Spojených států amerických zastupované úřadem NTIA (National Telecommunications and Information Administration), když v říjnu 2008 publikuje žádost o komentáře k procesu podpisu kořenové zóny technologií DNSSEC. Uzávěrka komentářů byla stanovena na 24. listopadu 2008, a sešlo se jich celkem úctyhodný počet – 53 komentářů od zainteresovaných jednotlivců až po TLD včetně CZ.NICu.

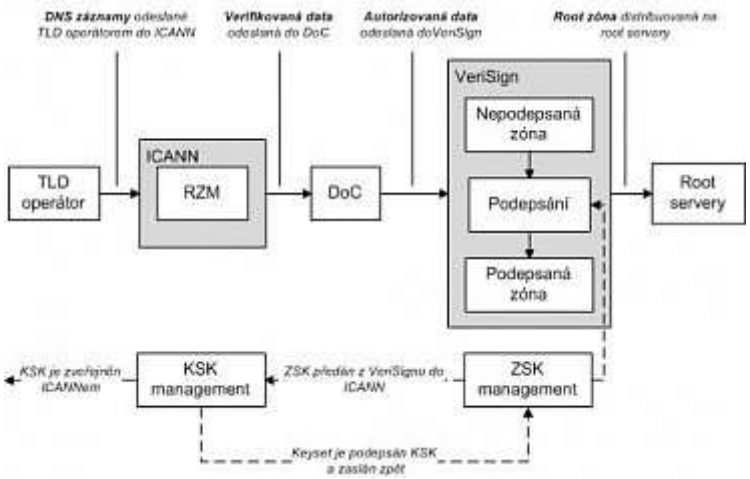
Na základě návrhu procesu kořenové zóny, který společně vyhotovili ICANN a Verisign (technický správce kořenové zóny), a těchto procesů vzniká v říjnu 2009 dokument Testing and Implementation Requirements for the Initial Deployment of DNSSEC in the Authoritative Root Zone, který obsahuje mimo jiné specifikaci minimální úrovně fyzického zabezpečení, požadavek na audit celého procesu, požadavky na certifikace použitého hardware (především pak HSM – Hardware Security Module), požadavek na použití algoritmu RSASHA1 s doporučeným využitím novějších algoritmů z rodiny SHA-2 a také popis některých rolí, resp. požadavek na zdvojení některých rolí.

Vydání tohoto dokumentu se specifikací předchází konference RIPE 59 v Lisabonu, kde Matt Larson ze společnosti Verisign a Joe Abley z ICANNu ohlašují na úterním plenárním zasedání implementaci DNSSECu v kořenové zóně Status Update: Implementing DNSSEC at the Root (video). Toto ohlášení bylo uvítáno bouřlivým potleskem, který je bohužel na videu zkrácen. O dva dny později v rámci pracovní skupiny DNS má Joe Abley prezentaci nazvanou Incremental Deployment of a Signed Root, kde popisuje konkrétní kroky vedoucí k podepsání kořenové zóny. Pojďme se nyní podívat, jaký byl plán a jak nakonec proběhl.

Pro začátek je zapotřebí si ujasnit role jednotlivých aktérů procesu:

- ICANN má v držení KSK klíče kořenové zóny. KSK klíč je určen pouze k podpisu dalších klíčů, tedy jsou jím podepsané záznamy DNSKEY – KSK i ZSK.
- IANA stejně jako do teď bude přijímat změny v nastavení DNS serverů pro jednotlivé TLD, jedná se o změny v NS záznamech. Nově bude možné podávat i požadavky na změnu DS záznamů.
- DoC NTIA formálně schvaluje veškeré změny v kořenové zóně a do jejích pravomocí spadne i schvalování změn v DS záznamech.
- VeriSign jako technický správce zónového souboru bude mít v držení ZSK klíče kořenové zóny. Tímto klíčem jsou podepsány všechny ostatní záznamy v zóně.

Celý proces podpisu lze schematicky vyjádřit takto:



Od této chvíle se proces podpisu kořenové zóny začal mít konkrétní obrysy. Prvním krokem byl začátek podepisování kořenové zóny 1. prosince 2009. V tuto chvíli se jednalo pouze o interní proces mezi ICANNem a VeriSignem. Toto interní testování bylo zapotřebí, aby byly nalezeny všechny možné problémy, které by mohly nastat.

V roce 2010 celý proces pokračoval postupnou publikací tzv. DURZ zóny. Zkratka DURZ znamená Deliberately Unvalidateable Root Zone, volně přeloženo jako „naschvál nevalidovatelná kořenová zóna". Tento krok byl zapotřebí pro otestování kompatibility s internetovým prostředím. Publikací tohoto falešného DNSSEC klíče došlo ke zvětšení odpovědí DNS zpráv ze serverů, které umí s DNSSECem pracovat, tedy posílají dotazy s nastaveným příkazem DNSSEC OK (DO bit). Klíč v DURZ kořenové zóně vypadal takto:

```
. 3600 IN DNSKEY 256 3 5 (\\
AwEAAa++++++++++++++++++++++++++++++\\
++THIS/KEY/AN/INVALID/KEY/AND/SHOULD\\
/NOT/BE/USED/CONTACT/ROOTSIGN/AT/ICA\\
NN/DOT/ORG/FOR/MORE/INFORMATION+++++\\
++++++++++++++++++++++++++++++++++++\\
++++++++++++++++++++++++++++++++++++\\
++++++++++++++++++++++++++++++++++++\\
++++++++++++++++++++++++++++++++++++\\
++++++++++++++++++++++++++++++++++++\\
++++++++++++++++++++++/=\\
) ; Key ID = 6477\\
```

Text uvnitř klíče říká, že tento klíč není validní, a nemá být používán. Mimochodem zajímavé je, že i přesto, že je v tomto klíči umístěn anglický text jedná z pohledu DNSSECu o validní RSASHA1 klíč. Podpisy v kořenové zóně (záznamy RRSIG) byly také falešné a byly vytvořeny za pomocí jiného klíče, než který byl publikován (Key ID = 41248).

Postupné nasazování na jednotlivé kořenové nameservery bylo zvoleno z toho důvodu, aby případný výpadek ovlivnil pokud možno co nejmenší skupinu uživatelů. Jako první přišel na řadu DNS server L.root-server.net, který je provozován samotným ICANNem. K publikaci DURZ podepsané zóny došlo 27. ledna 2010 a na DNS provozu se tato publikace projevila především ve velikosti UDP paketů:

Jelikož nebyly zaznamenány žádné problémy, pokračovalo se dle plánu. Dalším důležitým krokem byla publikace DURZ na kořenovém nameserveru A.root-server.net. Panovala obava, že by podepsání prvního nameserveru v pořadí mohlo způsobit problémy některým špatně implementovaným DNS klientům, kteří jej používají jako primární. Toto podepsání proběhlo v pořádku 10. února 2010 a naštěstí také nebyly zaznamenány žádné problémy.

Posuňme se rychle v čase kupředu. V březnu se nám před očima mihly servery M, I, D, K a E, v dubnu pak B, H, C, G a F. Náš virtuální stroj času se zastavil na datu 5. května 2010. Tou dobou probíhá v Praze konference RIPE 60, a je milou shodou okolností, že k publikaci DURZ podepsané zóny na posledním kořenovém nameserveru M.root-server.net dochází právě na

konferenci jejímž hostitelem byl právě CZ.NIC. O den později má Dave Knight z ICANNu prezentaci Last Root Server DURzed – Did the sky fall?, kde na grafech ukazuje lehké zvýšení počtu dotazů přes TCP, ale jinak se nic podstatného nestalo.

Po publikaci DURZ podepsané zóny započala usilovná práce na finální zprávě o testovacím procesu. Tato zpráva byla vydána 28. května 2010 a obsahuje shrnutí dosavadní práce na projektu a doporučení k nasazení podpisu kořenové zóny do produkce. K plnému nasazení podepsané kořenové zóny zbývají na začátku června tři kroky. První důležitý krok se odehrál na první KSK ceremonii, která proběhla 16. června 2010 na východním pobřeží USA v malém městečku Culpeper blízko Washingtonu, D.C.. Na této ceremonii byly inicializovány HSM moduly, vygenerován historicky první KSK klíč. Tímto klíčem byly podepsány ZSK klíče na první tři měsíce a podepsané klíče byly předány zástupci VeriSignu. Více o této KSK ceremonii se můžete dozvědět ze samostatného článku.

Druhá KSK ceremonie proběhla tento týden na západním pobřeží v El Segundu, které leží blízko Los Angeles. Na této druhé KSK ceremonii byly inicializovány další dva HSM moduly a byl do nich naimportován KSK klíč vygenerovaný před měsícem v Culpeperu. Této druhé KSK ceremonie se za CZ.NIC zúčastnil Ondřej Filip a jeho reportáž si můžete přečíst v článku Proč je podpis kořenové zóny tak důležitý pro budoucnost DNS? Tímto krokem má ICANN k dispozici dvě nezávislé lokality, kde může pomocí KSK klíče podepisovat další ZSK klíče. Toto se bude dít čtyřikrát do roka a CZ.NIC bude tento důležitý proces i nadále sledovat.

Po zprovoznění obou lokalit na východním i západním pobřeží USA již nic nebránilo pokračovat v plánu a 15. července byla publikována produkční podepsaná kořenová zóna. V tuto chvíli vám již nic nebrání otevřít si na oslavu této velké události láhev dobrého sektu a nakonfigurovat vlastní resolver tak, aby validoval kořenovou zónu. Instrukce jak na to, naleznete na stránkách CZ.NICu www.dnssec.cz na stránkách ICANNu věnovaných podpisu kořenové zóny www.root-dnssec.org, kde také naleznete provozní dokumenty a jednotlivé zprávy z podpisu kořenové zóny. Pokud vás téma DNSSECu a DNS zaujalo, tak další informace také můžete najít v našem seriálu DNSSEC a bezpečné DNS nebo na blogu CZ.NICu, kde se těmto tématům dlouhodobě věnujeme.

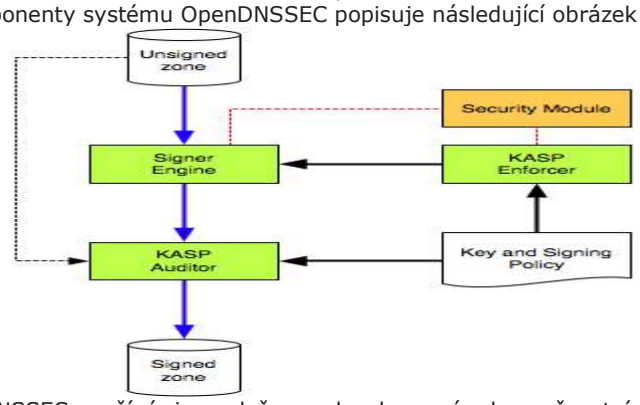
Bezpečné domény s OpenDNSSEC

Provozování podepsané domény je poněkud náročnější na údržbu. To proto, že podpisy mají časově omezenou platnost. Pokud není zóna včas znovu podepsána, stane se nevalidní. Navíc, ačkoli to není přímo vyžadováno, je vhodné čas od času vyměnit klíče, kterými jsou podpisy generovány. Celé této problematice se podrobně věnují starší články DNSSEC na autoritativním serveru a Údržba DNSSEC klíčů.

Právě k údržbě podepsané DNS zóny slouží projekt OpenDNSSEC. Z vnějšího pohledu jde o software, který na vstupu přebírá nepodepsané zónové soubory, a na výstupu generuje podepsané, přímo vhodné pro načtení autoritativním DNS serverem. Sám se přitom stará o generování a pravidelné obměny klíčů a podpisů.

Architektura OpenDNSSEC

Jednotlivé komponenty systému OpenDNSSEC popisuje následující obrázek z dokumentace:



Veškeré klíče, které OpenDNSSEC používá, jsou uloženy v hardwarovém bezpečnostním modulu (HSM). Používá se standardizované rozhraní PKCS#11, takže je možné použít zařízení od různých výrobců. Pro účely testování či malých domén, kde by bylo použití HSM zbytečný luxus, existuje podprojekt SoftHSM, implementující softwarové HSM, které ukládá klíče v databázi SQLite.

Vlastní podepisování zónových souborů má na starosti komponenta *Signer Engine*. Činí tak na základě objednávek komponenty *KASP Enforcer*. Jejím úkolem je vynucovat nastavené politiky klíčů a podpisů (Key and Signing Policy). Když to politika vyžaduje, automaticky vygeneruje nové klíče, provede výměnu klíčů a na závěr staré vymaže. V případě výměny KSK, kdy je zapotřebí vložit nové klíče do nadřazené zóny, úkoluje pomocí logů administrátora.

Před tím, než jsou podepsané zónové soubory vystaveny v systému DNS, provede *KASP Auditor* jejich validaci. Představuje doplňkovou ochranu před chybami v zónových souborech a implementaci předchozích komponent. Teprve v případě, že auditor shledá zónový soubor validní, v souladu politikou i zdrojovým souborem, je předán DNS serveru.

Instalace

OpenDNSSEC je k dispozici ve formě balíčků pro hlavní distribuce, instalace by tedy neměla představovat zvláštní problém. Drobnou potíž může představovat jen instalace úložiště SoftHSM, je třeba přidělit souboru s databází klíčů správná oprávnění, aby k němu mohl přistupovat jen uživatel, pod jehož účtem běží OpenDNSSEC. Můžeme použít třeba doporučený postup distribuce Gentoo. Nejprve nakonfigurujeme cestu k databázovému souboru v souboru /etc/softhsm.conf:

```
0:/var/lib/opendnssec/softhsm_slot0.db
```

Následně SoftHSM inicializujeme a změníme vlastníka databázového souboru:

```
# softhsm --init-token --slot 0 --label OpenDNSSEC
The SO PIN must have a length between 4 and 255 characters.
Enter SO PIN: 1234
The user PIN must have a length between 4 and 255 characters.
Enter user PIN: 1234
The token has been initialized.
# chown opendnssec:opendnssec /var/lib/opendnssec/softhsm_slot0.db
```

Konfigurace

Konfigurační soubory se nachází standardně v cestě /etc/opendnssec/. Hlavní konfigurační soubor conf.xml obsahuje rozumné výchozí hodnoty, takže jej nejspíše nebude nutné příliš upravovat, tedy za předpokladu, že vystačíte s použitím SoftHSM. Za zmínku dále stojí parametr Interval v konfiguraci *enforceru*. Určuje, jak často bude kontrolováno, zda by si zóna nezasloužila nové podepsání nebo výměnu klíče.

Nastavení politiky klíčů a podpisů se skrývá v souboru kasp.conf. Voleb je poměrně mnoho a detailní popis by zřejmě vydal na samostatný článek. Důležité je, že výchozí politika je nastavena rozumně až paranoidně, takže podpisy jsou platné jen 7 dnů (běžně se používá 30 dnů), tři dny před koncem platnosti se obnovují. Pro autentizaci popření existence záznamů se používá moderní NSEC3, takže nehrozí vyzrazení obsahu zóny procházením NSEC řetězců. Životnost klíčů ZSK je nastavena na 14 dnů, v případě KSK je to jeden rok. Součástí politiky je také nastavení hodnot TTL pro záznamy, které podepisováním vzniknou, stejně jako informace o TTL u DS záznamů nadřazené domény – to proto, aby mohl *enforcer* správně načasovat výměnu KSK.

Další konfigurační soubor zonelist.xml obsahuje seznam zón, které bude OpenDNSSEC obhospodařovat. Nemusíme jej však konfigurovat ručně, zóny je možné zadávat i odebírat interaktivně za běhu.

První spuštění

Pro evidenci konfigurace a stavu klíčů a podpisů u jednotlivých zón program používá databázi SQLite. Tu je třeba před prvním spuštěním vygenerovat. Nejdříve ale zkontrolujeme syntaktickou korektnost konfiguračních souborů:

```
# ods-kaspcheck
/etc/opendnssec/conf.xml validates
/etc/opendnssec/kasp.xml validates
# ods-ksmutil setup
*WARNING* This will erase all data in the database; are you sure? [y/N] y
SQLite database set to: /var/lib/opendnssec/kasp.db
fixing permissions on file /var/lib/opendnssec/kasp.db
zonelist filename set to /etc/opendnssec/zonelist.xml.
kasp filename set to /etc/opendnssec/kasp.xml.
Repository SoftHSM found
No Maximum Capacity set.
RequireBackup NOT set; please make sure that you know the potential problems of using keys which are not recoverable
/etc/opendnssec/conf.xml validates
/etc/opendnssec/kasp.xml validates
Policy default found
Info: converting P1Y to seconds; M interpreted as 31 days, Y interpreted as 365 days
```

Příkaz setup je možné spustit pouze jednou, při dalším spuštění by celou databázi přepsal a ztratila by se informace o doposud vytvořených klíčích a podpisech. Pokud později změníme konfiguraci, je třeba změny importovat do databáze příkazem ods-ksmutil update <conf|kasp|zonelist|all>.

Je-li databáze vytvořena, můžeme spustit *enforcer* a *signer*, buď pomocí init skriptů, nebo přímo příkazem ods-control start.

Přidání zónových souborů

Pokud se OpenDNSSEC úspěšně spustí, můžeme přidat zónové soubory. Nejjednodušší je, pokud se držíme konvence, že zónové soubory se jmenují stejně jako zóna, kterou reprezentují. Nejprve soubor bez podpisu vložíme do adresáře /var/lib/opendnssec/unsigned. Pak jej zavedeme do konfigurace pomocí:

```
# ods-ksmutil zone add -z example.net -p default
zonelist filename set to /etc/opendnssec/zonelist.xml.
SQLite database set to: /var/lib/opendnssec/db/kasp.db
Imported zone: example.net
```

Kromě zavedení zóny v databázi se také aktualizuje konfigurační soubor zonelist.xml, takže obsah konfiguračních souborů zůstává konzistentní s databází. Zóna se však ihned nepodepíše, nejprve si ji musí všimnout *enforcer*, který se probouzí v intervalech, konfigurovaných v souboru conf.xml (viz výše). Tuto akci můžeme urychlit restartem *enforceru*.

Provedeme-li později v zónovém souboru změnu, je třeba vždy požádat *signer* o znovupodepsání příkazem:

```
# ods-signer sign example.net
connecting to /var/run/opendnssec/engine.sock
Zone scheduled for immediate resign
```

Napojení na BIND

Podepsaný zónový soubor, který vznikne v adresáři /var/lib/opendnssec/signed je možné přímo předat autoritativnímu DNS serveru. V případě BINDu přidáme do konfiguračního souboru:

```
zone "example.net" {
type master;
file "/var/lib/opendnssec/signed/example.net";
};
```

Dále je důležité zajistit, aby se DNS server vždy dozvěděl, že je k dispozici nová verze podepsaného souboru. OpenDNSSEC umí při každém vystavení podepsaného zónového souboru zavolat příkaz, zadaný v konfiguračním souboru conf.xml

```
<NotifyCommand>/usr/sbin/rndc reload %zone</NotifyCommand>
```

Nezapomeňte, že po úpravě konfiguračního souboru je třeba promítnout změny do databáze příkazem ods-ksmutil update conf a také to, že daný příkaz se spouští s efektivním oprávněním uživatele, pod kterým běží *signer*. Aby mohl pomocí utility rndc ovládat BIND, je potřeba udělit mu práva ke čtení souboru /etc/bind/rndc.key, například přidáním souboru do skupiny opendnssec.

Spolupráce s nadřazenou zónou

Pokud OpenDNSSEC dospěje k názoru, že by bylo vhodné vyměnit KSK a tedy i podpis v nadřazené zóně, upozorní na to v syslogu:

```
ods-enforcerd: WARNING: KSK Retirement reached; please submit the new DS for example.net and use ods-ksmutil key ds-seen when the DS appears in the DNS.
```

To se týká i prvního zavedení nově podepsané zóny. Potřebný klíč připravený k publikaci v registrech, které používají tzv. KEYSETy (například registr domény cz), získáme níže uvedeným příkazem. Pro registry, které přijímají přímo DSSETy tyto získáme přidáním přepínače --ds.

```
# ods-ksmutil key export --zone example.net --keystate READY
SQLite database set to: /var/lib/opendnssec/db/kasp.db

;ready KSK DNSKEY record:
example.net.    3600    IN      DNSKEY  257 3 7 AwEAA…TeDj1494j/0= ;{id = 6489 (ksk), size = 2048b}
```

V případě rotace klíčů můžeme předchozí DS záznamy smazat. Poté, co se DS záznamy objeví v nadřazené zóně, je nutné tuto informaci předat *enforceru* použitím příkazu:

```
# ods-ksmutil key ds-seen -z example.net -x 6489
SQLite database set to: /var/lib/opendnssec/db/kasp.db
Found key with CKA_ID d61a44f44ad106c767a330bf9ae43f15
Key d61a44f44ad106c767a330bf9ae43f15 made active
Error: retiring a key would leave no active keys on zone, skipping...
```

Poslední chyba nás při první publikaci nemusí znepokojovat, *enforcer* se snaží automaticky vyřadit nejstaší klíč, což je však v případě publikace prvního klíče zároveň ten nejnovější. Při rotaci klíčů toto funguje korektně.

Závěrem

Vždy zkontrolujte, zda jsou podepsané soubory totožné s těmi, které nabízí DNS server a to včetně všech *slave* serverů. *Enforcer* provádí časování výměny klíčů na základě okamžiku jejich vystavení v souboru (maximální zdržení při kopírování na *slave* servery se konfiguruje v politice), takže pokud některý z DNS serverů nabízí starou verzi zóny, stane se zóna po čase nevalidní.

OpenDNSSEC dokáže také zavolat zvolený příkaz při potřebě publikace nových DS záznamů v nadřazené zóně. Pokud váš registrátor podporuje nějaký způsob strojové editace doménových záznamů, je možné i výměnu KSK plně automatizovat.

Moderní DNSSEC: eliptické křivky a nevinné lži

Podpora eliptických křivek v DNSSECu

Algoritmy založené na eliptických křivkách jsou aktuálním módním trendem v kryptografii, a tak není divu, že ani DNSSEC se jim nevyhýbá. Mezi hlavní výhody eliptických křivek patří rychlejší generování podpisů a především pak podstatně kratší délka klíče při stejné síle. Jak je možné ověřit například na serveru Keylength.com, ECDSA klíč délky 256 bitů poskytuje bezpečnost odpovídající RSA klíči délky kolem 3072 bitů. Takový klíč je tedy pravděpodobně bezpečnější i než 2048bitový RSA klíč samotné kořenové zóny. Malá velikost klíče i podpisů je obecně pro DNS výhodná, neboť s narůstáním objemu DNS zpráv dochází k jistým provozním problémům, jako je fragmentace, přechod na transportní protokol TCP nebo nárůst zesilovacího faktoru při kybernetických útocích.

Podporu pro eliptické křivky v DNSSECu zavádí RFC 6605, které vyšlo již v roce 2012. Konkrétně jsou definovány dva algoritmy a sice ECDSAP256SHA256 s číslem 13 a ECDSAP384SHA384 s číslem 14. Jak názvy napovídají, jedná se o použití eliptických křivek P-256, resp. P-384 podle specifikace FIPS 186–3 a hashovací funkce SHA256, resp. SHA384.

Jedinou, zato ale zásadní, nevýhodou použití ECDSA algoritmu v DNSSECu je jeho nepodpora staršími validátory. Ta vychází zejména z nejistoty ohledně existence patentů, kvůli které některé distribuce podporu eliptických křivek v OpenSSL záměrně vypínaly. Ještě v roce 2014 George Michaelson z laboratoří APNIC před jejich použitím varoval. Od té doby však nastal velký pokrok, validaci ECDSA podpisů zvládne ve výchozím nastavení většina aktuálních linuxových distribucí. Důležité také je, že i při nepodpoře ECDSA algoritmu validátorem nedojde k žádnému fatálnímu selhání, validátor zónu propustí bez validace stejně, jako by nebyla podepsána vůbec.

O tom, jak je na tom váš DNS resolver s podporou různých DNSSEC algoritmů se můžete snadno přesvědčit pomocí utilitky alg_rep. Ta provede úplný test všech definovaných DNSSEC algoritmů proti všem typům DS záznamů a oběma typům NSEC záznamů. Výsledek je zobrazen v přehledné tabulce:

```
Zone dnssec-test.org.  Qtype DNSKEY Resolver [2001:4860:4860::8888]
DS    : 1 2 3 4 | 1 2 3 4
ALGS  :   NSEC    |   NSEC3
alg-1 : S S S S | x x x x => RSA-MD5 OBSOLETE
alg-3 : - - - - | x x x x => DSA/SHA1
alg-5 : V V - V | x x x x => RSA/SHA1
alg-6 : x x x x | - - - - => DSA-NSEC3-SHA1
alg-7 : x x x x | V V - V => RSA-NSEC3-SHA1
alg-8 : V V - V | V V - V => RSA-SHA256
alg-10 : V V - V | V V - V => RSA-SHA512
alg-12 : - - - - | - - - - => GOST-ECC
alg-13 : V V - V | V V - V => ECDSAP256SHA256
alg-14 : V V - V | V V - V => ECDSAP384SHA384
V == Validates  - == Answer  x == Alg Not specified
T == Timeout S == ServFail O == Other Error
DS algs 1=SHA1 2=SHA2-256 3=GOST 4=SHA2-384
```

V ideálním případě budou ve všech polích tabulky znaky V, značící úspěšnou validaci, případně x pro neexistující kombinace algoritmu a NSEC záznamu. Pokud je v poli znak -, znamená to, že validátor daný podpis nepodporuje a data propustil bez ověření. Znak S pak značí návratový kód selhání serveru. Ve výše uvedeném příkladu takovou odpověď vrací zóny podepsané zastaralým protokolem RSA-MD5.

Nevinné lži při online podepisování

Zásadním návrhovým požadavkem, který do velké míry ovlivnil výsledný návrh protokolu DNSSEC, byla podpora offline podepisování, tedy to, aby autoritativní DNS server mohl pracovat zcela bez přístupu k privátnímu klíči, pomocí kterého jsou

podpisy vytvářeny. Splnění tohoto požadavku spolu s podporou pro tzv. *žolíkové*DNS záznamy a odolností proti útokům přehráním zaznamenané komunikace si vyžádalo poměrně komplexní systém NSEC záznamů, který byl ještě dále zkomplikován přechodem na NSEC3 záznamy za účelem znesnadnění procházení zónového souboru postupnými dotazy.

Důsledkem je, že zatímco pozitivní odpověď na DNS dotaz je přímočará – k dotazovaným záznamům se jednoduše připojí podpis – negativní odpověď musí obsahovat mnohem víc dat: kromě tradičního SOA záznamu a jeho podpisu ještě NSEC záznam prokazující, že dotazované jméno neexistuje a NSEC záznam prokazující, že neexistuje ani *žolíkový* záznam, který by pokrýval dotazované jméno, oba záznamy ještě doprovázené podpisy. Při použití NSEC3 jsou takové záznamy dokonce tři. Stejně tak jsou komplikované i pozitivní odpovědi vzniklé z *žolíkových*záznamů, kde je třeba kromě podepsané odpovědi doručit i podepsaný NSEC důkaz, že opravdu neexistuje lepší záznam a bylo nutné použít *žolík*. Celou problematiku hezky shrnuje informativní RFC 7129. Důsledkem takto složitého systému je kromě nárůstu délky zpráv také spousta implementačních chyb, způsobujících například problémy při řetězení resolverů a validaci žolíkových záznamů.

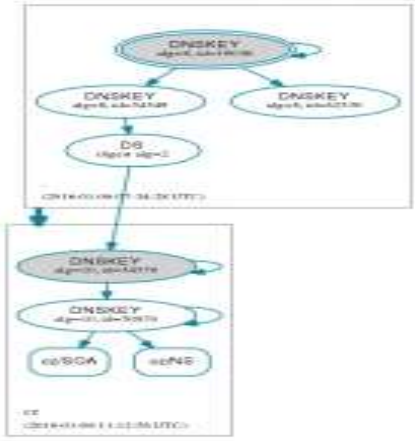
Offline podepisování však není *jediný možný způsob nasazení DNSSEC*. Online podepisování může být výhodné jak pro menší instalace, tak i pro velké sítě CDN, které obsah DNS odpovědi mění v reálném čase na základě nejrůznějších faktorů. Zde je třeba zdůraznit, že přítomnost privátního klíče na DNS serveru není nic strašidelného; úplně stejně fungují snad všechny kryptografické protokoly, které znáte, namátkou třeba HTTPS nebo SSH. Přítomnost privátního klíče pak umožní zásadním způsobem zjednodušit obsah negativních odpovědí, protože odpověď obsahuje přímo podepsaný dotaz.

Tento postup se obecně nazývá NSEC/NSEC3 *white lies*, česky tedy *nevinné lži*. Zatímco v při tradičním podepisování by dotaz na neexistující jméno jablko vrátil NSEC záznam s nejbližšími existujícími záznamy, třeba banán NSEC pomeranč, v režimu nevinných lží server vygeneruje minimální NSEC záznam, pokrývající pouze dotazované jméno, například tedy jablkn NSEC jablkp. Takový postup implementoval samotný DAN KAMINSKY ve svém DNSSEC proxy serveru Phreebird a dočkal se i standardizace v RFC 4470.

S ještě zajímavější variantou *nevinných lží* přišla společnost CloudFlare. Jejich implementace online podepisování je zvláštní tím, že na jakýkoli dotaz odpoví pozitivně, tedy s návratovým kódem NOERROR. Pokud dotazovaná data neexistují, raději předstírají, že dané jméno sice existuje, ale nemá žádný záznam dotazovaného typu. Tímto způsobem, obsahuje negativní odpověď pouze jeden SOA a jeden NSEC záznam. Spolu s použitím ECDSA algoritmu je tak možné i negativní odpovědi stlačit pod délku zprávy 512 bajtů a ušetřit tak případné problémy s velkými DNS zprávami.

Podepisování jedním klíčem

Od počátku DNSSECu bylo doporučováno používání dvojice klíčů pro podepisování zónového souboru. Slabší klíč ZSK podepisuje všechny záznamy v zóně, silnější klíč KSK podepisuje pouze záznam DNSKEY s veřejnou částí klíče ZSK. Otisk klíče KSK je publikován v DS záznamu nadřazené zóny, čímž je sestaven řetěz důvěry.



Vizualizace hierarchie KSK a ZSK klíčů nástrojem DNSViz.net

Tímto způsobem se řeší požadavek na dostatečně krátké podpisy, aby objem DNS dat příliš nenarostl, zároveň s protichůdným požadavkem na dostatečně bezpečné klíče, aby je nebylo nutné měnit příliš často, neboť každá výměna vyžaduje (typicky manuální) komunikaci s operátorem nadřazené zóny. Obvyklá současná praxe je použití 1024bitového RSA klíče jako ZSK a 2048bitového RSA klíče jako KSK, přičemž ZSK je obvykle měněn automaticky každý měsíc, KSK manuálně po roce, dvou, nebo vůbec. Zajímavé je to zejména v kontrastu s TLS certifikáty, kde ještě nedávno existovaly kořenové certifikáty používající 1024bitové RSA klíče, vydané s platností 10 let.

Příchod eliptických křivek umožňuje i správu klíčů výrazným způsobem zjednodušit. 256bitový ECDSA klíč je silnější než běžné současné KSK klíče a zároveň generuje podpisy kratší než současné ZSK klíče. Odpadá zde tedy požadavek na častou výměnu ZSK klíče a nic nebrání podepisování celé zóny jedním klíčem, který stačí manuálně jednou za čas vyměnit.

Používání původní dvojice samostatných KSK a ZSK klíčů samozřejmě není zakázáno. Třeba dříve zmíněná společnost CloudFlare používá dvojici klíčů z důvodu bezpečnosti – ZSK klíče jsou přítomny přímo na DNS serverech ve všech uzlech, zatímco KSK klíče jsou v bezpečném kontrolovaném prostředí. V případě kompromitace ZSK klíče je možné tento vyměnit a revokovat, aniž by to vyžadovalo komunikaci s nadřazenou zónou.

Další křivky na obzoru

Standardizované křivky P.256 a P.384, které jsou v současné době jediné dvě podporované v DNSSECu, byly kryptology Danielem J. Bernsteinem a Tanjou Lange v nedávném výzkumu označeny jako *nebezpečné*. Zřejmě se nejedná se o nějaké bezprostřední riziko, rozhodně však není od věci podporovat i jiné křivky, takové, které zmiňovaná zpráva označuje jako bezpečné.

Jednou z nich je i Bernsteinova křivka Curve25519. Podporu pro její použití v DNSSECu v tuto chvíli NAVRHUJE V IETF ONDŘEJ SURÝ, stejně jako další křivku Ed448. Není však možné očekávat reálnou nasaditelnost těchto nových křivek dříve než za několik let.

DNSSEC s BIND 9.9 snadno a rychle

Podpora automatického DNSSEC podepisování byla v serveru BIND přitomna již dříve, bylo ale vždy nutné přepnout zónu do dynamického režimu, což s sebou mohlo nést některé problémy. Ve verzi 9.9 se ale poprvé objevila funkce zvaná *inline signing*, která umožňuje zapnout podepisování a přitom zachovat původní zónové soubory beze změn.

Výchozí situace

Nasazení si prakticky předvedeme na Debianu Jessie, který obsahuje balíček bind9ve verzi 9.9.5. Přepokládejme, že server slouží jako autoritativní server pro zónu example.com, která používá obyčejný textový zónový soubor umístěný v /etc/bind/example.com. V konfiguračním souboru /etc/bind/named.conf.local je pak následující definice zóny:

```
zone "example.com" {
type master;
file "/etc/bind/example.com";
};
```

Krok 0: dostatek entropie

Ještě než začneme zónu podepisovat, je dobré se ujistit, že systém bude mít k dispozici dostatek entropie, aby mohl generovat náhodná čísla skutečně náhodně. Všechny dále zmíněné nástroje používají jako zdroj náhodných čísel zařízení /dev/random, které při nedostatku entropie proces vytváření klíčů nebo podpisů blokuje až do chvíle, kdy entropie naroste zpět. Množství dat, které zařízení /dev/random vyrábí, můžeme ověřit nástrojem pv, který vypisuje každou sekundu počet bajtů, které protekly rourou:

```
# pv -nb /dev/random > /dev/null
192
192
192
^C
```

Opakující se hodnota značí vyčerpanou zásobu entropie. Ve většině případů si můžeme pomoci instalací démona haveged, který získává další entropii z nejrůznějších vnitřních stavů procesoru:

```
# apt-get install haveged
# pv -nb /dev/random > /dev/null
1586646
3219499
4800458
^C
```

Krok 1: vygenerujeme klíč

Automatické podepisování v BINDu dokáže generovat a obnovovat podpisy, klíče je však třeba generovat a případně měnit ručně. Abychom snížili míru ruční práce na minimum, použijeme k podpisu zóny pouze jediný klíč s algoritmem ECDSA P-256 a hashovací funkcí SHA256. Síla tohoto algoritmu by měla být větší než síla RSA klíče o délce 2048 bitů, takže jej není nutné vyměňovat dříve než za několik let. Zároveň jsou podpisy tímto algoritmem tak krátké, že si můžeme dovolit použít jej k podepisování každého záznamu zónového souboru.

```
# mkdir /etc/bind/keys
# cd /etc/bind/keys
# dnssec-keygen -a ECDSAP256SHA256 -fK example.com
Generating key pair.
Kexample.com.+013+32462
# chmod g+r K*.private
```

V adresáři /etc/bind/keys vzniknou soubory s veřejným i privátním klíčem. Druhý jmenovaný má nastavena práva na 0600, takže je čitelný a zapisovatelný jen pro vlastníka, což je uživatel root. Aby jej mohl použít BIND k podepisování, je třeba práva upravit tak, aby daný soubor mohl číst. Tuto úpravu je třeba opakovat po každém generování klíče.

Krok 2: aktivace inline signingu

Nyní je již vše připraveno k aktivaci podepisování. Přitom se BIND pokusí vedle zónového souboru vytvořit soubory se žurnálem a s podepsanou verzí zóny. Je-li zónový soubor v /etc, toto se nepovede kvůli nedostatečnému oprávnění. Navíc z praktických důvodů není vhodné, aby se v konfiguračním adresáři objevovaly soubory automaticky generované a spravované. Vhodným řešením je přesunutí cesty k zónovému souboru z /etc do provozního adresáře /var/cache/bind, kam má BIND právo zapisovat. Vytvoříme tedy symbolický odkaz na zónový soubor:

```
# ln -s /etc/bind/example.com /var/cache/bind
```

Nyní již můžeme aktivovat podepisování úpravou konfigurace:

```
zone "example.com" {
type master;
file "example.com";
inline-signing yes;
auto-dnssec maintain;
key-directory "/etc/bind/keys";
};
```

Po reloadu příkazem rndc reload by mělo dojít během několika okamžiků k podepsání zóny. Průběh můžeme sledovat pomocí:

```
# rndc signing -list example.com
Done signing with key 32462/ECDSAP256SHA256
```

Standardně je podepisováno v režimu NSEC, kdy jsou všechna jména v zónovém souboru provázána a je tedy možné postupným dotazováním zjistit všechna existující jména. Pro situace, kde je toto problém, je možné přejít na hashované záznamy NSEC3:

```
# rndc signing -nsec3param 1 0 10 deadbeef example.com
```

Číselné parametry určují postupně hashovací algoritmus NSEC3 (jediný definovaný je 1 – SHA256), flagy (žádné), počet iterací hashování (10 je rozumná hodnota) a konečně sůl, která zabraňuje použití tzv. *rainbow tabulek*. Jde o libovolné šestnáctkové

číslo dlouhé až 255 bytů. Důrazně zde doporučuji odolat pokušení k použití zprofanovaných slov jako cafebabe, deadbeef nebo faceb00c.

Podepsaná zóna je uložena v provozním adresáři v souboru s příponou .signed. Kromě toho se v daném adresáři ještě nachází žurnálový soubor sledující změny v původním nepodepsaném souboru a žurnálový soubor podepsaného souboru. Podepsaný soubor je v surovém formátu, pokud jej chceme prozkoumat, je třeba použít utilitu named-compilezone:

```
# named-compilezone -f raw -j -o - example.com /var/cache/bind/example.com.signed
zone example.com/IN: loaded serial 9 (DNSSEC signed)
example.com. 60 IN SOA ns.example.com. hostmaster.example.com. 9 120 10 3600 60
…
OK
```

Změny DNS dat provádíme stejně jako před zavedením DNSSECu – editací nepodepsaného zónového souboru a reloadem serveru. Důležité je při každé změně zvýšit sériové číslo zóny v SOA záznamu, jinak ji BIND odmítne načíst. Sériové číslo podepsané zóny sleduje číslo nepodepsané zóny, automaticky je ale zvyšováno při obnovování podpisů nebo manipulaci s klíči.

Krok 3: sdělení nadřazené zóně

Posledním krokem k vytvoření řetězu důvěry je umístění otisku klíče do nadřazené zóny. K tomu slouží utilitka dnssec-dsfromkey:

```
# dnssec-dsfromkey /etc/bind/keys/Kexample.com.+013+32462
example.com. IN DS 32462 13 1 5E6C8…9D20C8
example.com. IN DS 32462 13 2 AB779…8A9001875886A0FF9170A2579AC9E1AB
```

Vygenerované otisky se liší pouze algoritmem hashovací funkce, která je u kratšího SHA1, u delšího SHA256. Není třeba umisťovat do nadřazené zóny oba, zcela postačuje ten druhý, bezpečnější.

Speciálním případem jsou registry domén .cz a .eu. Tyto registry nepřijímají DS záznamy, ale rovnou celé veřejné klíče. Důvod je ten, že DS záznam je otiskem jak veřejného klíče, tak i doménového jména. Registry .cz a .eu pracují s tzv. *keysety*, které mohou být přiřazeny k většímu množství doménových jmen. Příslušné DS záznamy si pak registr vyrobí sám. Pro nás je důležité, že do keysetu vkládáme přímo obsah souboru s veřejným klíčem.

Krok 4: výměna klíče

Ačkoli je klíč daného algoritmu dostatečně silný, může nastat časem potřeba klíč vyměnit. Ani to není příliš obtížné, byť jistá ruční práce je vyžadována. Nejprve vygenerujeme nový klíč stejně jako v kroku 1 a sdělíme BINDu, že jej má používat:

```
# cd /etc/bind/keys/
# dnssec-keygen -a ECDSAP256SHA256 -fK example.com
Generating key pair.
Kexample.com.+013+11957
# chmod g+r *.private
# rndc sign example.com
```

V tuto chvíli jsou aktivní oba klíče a můžeme tedy změnit DS záznam v nadřazené zóně tak, aby mířil na nový klíč. Po určité době, kdy se změna projeví a starý klíč se stane nepotřebným, můžeme naplánovat jeho deaktivaci (přestane být používán k vytváření podpisů) a poté jeho odstranění ze zóny:

```
# cd /etc/bind/keys
# dnssec-settime -I now -D +35d Kexample.com.+013+32462
dnssec-settime: warning: Permissions on the file
./Kexample.com.+013+32462.private have changed from 0640 to 0600 as a result of this operation.
./Kexample.com.+013+32462.key
./Kexample.com.+013+32462.private
# chmod g+r *.private
# rndc sign example.com
```

Tímto je starý klíč okamžitě deaktivován a zcela odstraněn po 35 dnech – v této době by už zaručeně měly být všechny podpisy přegenerovány a klíč tedy bude bezpečné odebrat. To proběhne automaticky; BIND totiž adresář s klíči pravidelně kontroluje a s klíči nakládá podle časovacích metadat.

Závěrem

Podpora automatického podepisování s *inline signingem* umožňuje nasadit DNSSEC všude tam, kde je používán BIND. Nedochází zde k žádnému zásahu ani na straně vstupu, kde je obsluha stejná jako dříve, stejně jako není potřeba nijak upravovat případné *slave* servery, tedy za předpokladu, že používají přiměřeně aktuální software. Použití algoritmu ECDSA a jediného klíče celý postup zjednodušuje tak, že prakticky není potřeba věnovat žádnou péči navíc.

Doména .CZ spouští jako první automatickou správu DNSSEC klíčů

JAROMÍR TALÍŘ zahájil svou přednášku na IT 17 rekapitulací aktuálního stavu: 51,5 % domén v zóně .CZ už dnes DNSSEC používá. Rádi bychom, aby se procento přiblížilo co nejvíce ke stu. Zjistili jsme například, že 21 156 domén v .CZ je podepsaných, ale nemají otisk klíče v registru. Takové domény se tedy tváří jako nepodepsané. Jedná se o potenciál ke zvýšení počtu podepsaných zón až o další dvě procenta.

Překážkami dalšího rozšiřování DNSSECu je příliš mnoho entit:

- držitel doménového jména
- provozovatel DNS
- registrátor
- registr

V jednoduchém případě, kdy registrátor zároveň provozuje DNS, je to snadné, neboť komunikaci otisku DNSSEC klíče provede s registrem v případě potřeby sám. Problém je s provozovateli DNS, kteří sami registrátory nejsou. Ti pak mohou komunikovat s registrem jen prostřednictvím držitele domény, který obvykle problematice nerozumí a jen velmi těžko se mu vysvětluje, co má u svého registrátora nastavit. To samo o sobě je velmi komplikované, nehledě na to, že klíče by se měly po čase měnit, takže celý proces je nutné pravidelně opakovat.

V rámci IETF probíhají už několik let pokusy tento proces usnadnit. Prvním je RFC 7344 ze září 2014, které definuje nové typy DNS záznamů: CDS a CDNSKEY. Jsou určeny k tomu, aby je umístil do apexu zóny její provozovatel a signalizoval tak nadřazené

zóně, jaký otisk klíče má být v DS záznamu, zajišťujícím bezpečnou delegaci. Vzhledem k tomu, že registrační systém CZ.NIC používá tzv. *keysety*, tedy sady klíčů, ze kterých si sám generuje DS záznamy po přiřazení *keysetu* ke konkrétní doméně, je pro CZ domény relevantní jen záznam typu CDNSKEY.

Výše uvedené RFC řeší pouze změnu klíčů u zóny, která již DNSSEC má. Dokument RFC 8078 z března 2017 tento způsob rozšiřuje o možnost zavádění DNSSECu na zóně, která jej dosud nemá a také o možnost zrušení bezpečné delegace, pokud se zóna rozhodne DNSSEC vypnout.

V současné době je v procesu IETF návrh, který obrací model *pull* na model *push*, kdy místo aby nadřazená zóna pravidelně zkoumala CDS/ CDNSKEY záznamy z podřízené zóny, může provozovatel podřízené zóny potřebná data do registru poslat. Návrh sledujeme, ale protože jsme nechtěli čekat, prozatím jsme implementovali aktuální standardy, komentuje JAROMÍR TALÍŘ.

Autor: Jiří Průša, CZ.NIC

Jaromír Talíř

Podpora u klientů

Prvním provozovatelem DNS služby, který uvedené standardy podporuje, je Cloudflare. Jde o globální CDN síť, která nabízí DNSSEC na jediné kliknutí.DNSSEC je tam implementován průkopnicky, používají on-line podepisování a algoritmy s eliptickými křivkami. Nicméně nejsou doménovým registrátorem – veškeré změny týkající se DNSSECu musí dělat prostřednictvím zákazníka a jeho registrátora.

Od 20. června Cloudflare publikuje CDNSKEY záznamy. Už zhruba dva měsíce publikují CDS záznam, s tím ale v registru CZ.NIC nedokážeme pracovat.U Cloudflare jsou provozovány asi čtyři tisíce českých domén, které jsou registrovány u 34 různých registrátorů. Někteří registrátoři dokonce ani DNSSEC nepodporují, takže není možné standardním způsobem *keyset* do registru vložit.

Podpora CDNSKEY záznamů při hostování DNS na vlastním serveru je zatím slabá. Populární nástroj OpenDNSSEC je zatím nepodporuje, nicméně máme informace, že podpora je na cestovní mapě a měla by se letos objevit. Nejpoužívanější DNS server BIND ve verzi 9.11 záznamy podporuje částečně: Umí automaticky generovat klíče, ale publikaci záznamů je třeba řídit ručně. Jediným nástrojem, který v tuto chvíli podporuje CDNSKEY záznamy je Knot DNS od verze 2.5, která vyšla 6. června 2017. Stačí jednou nastavit a zapomenout. Tak jednoduché by to mělo být.

Rotace KSK klíče v Knotu funguje na principu dvojího podpisu. Běží v něm kontrola, zda byl DS záznam v nadřazené zóně publikován. V konfiguraci je možné nastavit buď kontrolu proti autoritativním serverům nadřazené zóny, nebo použít nějaké rekurzivní resolvery. K rotaci KSK klíče dojde až v okamžiku, kdy všechny sledované DNS servery mají aktualizovaný DS záznam. Informace se ale nevaliduje, proto je dobré použít DNSSEC-validující resolver.

Ve vývoji je návrh *push* mechanizmu pomocí REST rozhraní. Také se v Knotu připravuje podpora pro rotaci CSK klíče, tedy jediného klíče, který podepisuje celou zónu. Takové použití je oblíbené u algoritmů používajících eliptické křivky.

Nasazení v registru

Nasazení na straně registru mělo tři možné scénáře.

1. automatické změny budou prováděny registrátory,
2. CZ.NIC bude automatickou správu provádět pro speciálně označené *keysety*,
3. CZ.NIC bude vše zpracovat místo registrátorů.

Protože podle filozofie distribuovaného registračního modelu jsou *keysety* ve správě určeného registrátora, zahájil CZ.NIC diskuzi s registrátory, zda jsou ochotní automatickou správu implementovat a pokud ne, zda by jim vadilo, pokud CZ.NIC automatickou správu převezme. Negativní odpověď na obě otázky vedla k realizaci třetího jmenovaného scénáře.

U domén, které dosud nemají žádný *keyset*, se začne registr ptát na CDNSKEYzáznam. K dotazování se použije TCP protokol, aby byla menší šance na podvržení odpovědi. Když bude záznam nalezen, stane se doména kandidátem na automatické zavedení bezpečné delegace. O tom informujeme e-mailem technického správce *nssetu*. Dále bude registr 7 dnů sledovat, zda se CDNSKEYzáznam nebude měnit. Po uplynutí sedmi dnů CZ.NIC zaregistruje nový *keyset* ve tvaru AUTO-<náhodný řetězec>, přiřadí jej k doméně a informuje e-mailem držitele domény a prostřednictvím zprávy EPP protokolu také registrátora domény.

Pro domény, které už mají přiřazený automatický *keyset*, je situace jednodušší, neboť informace je autentizována stávajícím DNSSEC podpisem. Odpadá tedy sedmidenní ochranná lhůta a obsah automatického *keysetu* se upraví podle CDNSKEY záznamu. V případě, že klient publikuje speciální formát záznamu určený ke zrušení delegace, dojde k odstranění *keysetu*. Pro domény, které mají přiřazený jiný než automatický *keyset*, dojde při nalezení CDNSKEY záznamu k založení nového *keysetu*, který ten ručně přiřazený nahradí.

Automatické *keysety* budou mít jako registrátora i technický kontakt CZ.NIC. Nebude blokována možnost ručního přiřazení automatického *keysetu* k jiné doméně, ale takové nastavení nejspíše povede dříve či později k nefunkčnosti,varuje JAROMÍR TALÍŘ.

Pokud uživatel nechce DNSSEC na své doméně používat, měl by požádat svého provozovatele DNS, aby CDNSKEY záznam nepublikoval. Také je možné přiřazení *keysetu* zablokovat použitím zámku na úrovni registru, například prostřednictvím aplikace doménový prohlížeč. Pokud už ke změně došlo, je možné *keyset* z domény vymazat prostřednictvím registrátora. Také je možné změnit *nsset* na nějaký jiný, při této operaci se z domény automaticky odebere *keyset*. Takové řešení se hodí zejména pro registrátory, kteří správu *keysetů* vůbec neimplementují.

Už je spuštěno

Systém automatické správy bezpečné delegace spouští CZ.NIC 20. června 2017. Prohledávání domén na CDNSKEY záznamy probíhá jednou denně, zatím jsou prohledávány pouze domény bez DNSSECu. Za týden začnou být prohledávány i domény s automaticky přiřazenými *keysety.* Někdy zhruba za měsíc začneme prohledávat i domény s ručně spravovanými *keysety*.

Pokud používáte Knot DNS, aktualizujte na nejnovější verzi a automatickou správu klíčů zapněte. Pokud používáte Cloudflare, prosím klikněte v administraci na to jedno tlačítko, o vše ostatní se postaráme my, vyzval na závěr své přednášky JAROMÍR TALÍŘ.

Jediný krok nutný k zavedení DNSSEC na doméně u Cloudflare.
