UNIVERZA V LJUBLJANI

Fakulteta za računalništvo in informatiko

Dušan Kozic

VARNOST V SISTEMU DNS IN DNSSEC

DIPLOMSKO DELO VISOKOŠOLSKEGA
STROKOVNEGA ŠTUDIJA

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00071/2011
Datum: 01.03.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DUŠAN KOZIC**

Naslov: **VARNOST V SISTEMU DNS IN DNSSEC**
SECURITY IN DNS AND DNSSEC

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Opišite delovanje sistema in protokola DNS. Osredotočite se na analizo njegovih ranljivosti, znanih napadov in možnih protiukrepov. Navedite, kakšne so možnosti doseganja višjega nivoja varnosti brez sprememb protokola ter motivacijo za zasnovo varnejšega protokola DNSSEC. Nato opišite protokol in načela delovanja DNSSEC, varnostne mehanizme, ki jih vsebuje, nove zapise virov, način poizvedovanja in validacije zapisov, ter način vzdrževanja cone. Izpostavite prednosti in slabosti, ki jih prinaša nadomeščanje obstoječega DNS z novim DNSSEC. V praktičnem delu diplome predstavite trenutno stanje glede uveljavljanja DNSSEC v Sloveniji in globalno. V testnem okolju testirajte dosegljive implementacije DNS strežnikov in odjemalcev, ki podpirajo DNSSEC. Primerjajte jih med seboj ter komentirajte njihovo uporabnost v resničnem (poslovnem) okolju.

Mentor:

M. Ciglarič

doc. dr. Mojca Ciglarič

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Somentor:

prof. dr. Andrej Kos

UNIVERZA V LJUBLJANI

Fakulteta za računalništvo in informatiko

Dušan Kozic

VARNOST V SISTEMU DNS IN DNSSEC

DIPLOMSKO DELO VISOKOŠOLSKEGA
STROKOVNEGA ŠTUDIJA

MENTOR: doc. dr. Mojca Ciglarič

SOMENTOR: prof. dr. Andrej Kos

Ljubljana, 2011

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani **Dušan Kozic**,

z vpisno številko **63050190**,

sem avtor diplomskega dela z naslovom:

Varnost v sistemu DNS in DNSSEC

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom (naziv, ime in priimek)

  doc. dr. Mojce Ciglarič

  in somentorstvom (naziv, ime in priimek)

  prof. dr. Andreja Kosa

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«

V Ljubljani, dne 13. 6. 2011 Podpis avtorja: _____

ZAHVALA

»On je podaril ljudem znanje,
da bi ga mogli proslavljati v njegovih čudovitih delih.« (Sir 38,6)

Za pomoč pri izdelavi diplomske naloge se zahvaljujem mentorici doc. dr. Mojci Ciglarič in somentorju prof. dr. Andreju Kosu. Za koristne napotke in pomoč pri postavitvi testnega okolja, se zahvaljujem mag. Luki Koršiču in Janezu Sterletu iz Laboratorija za telekomunikacije Fakultete za elektrotehniko. Hvala tudi Benjaminu Zwittnigu iz Arnesa za veliko posredovanih informacij o sistemu DNS in DNSSEC.

Hvaležen sem tudi prijateljem za spodbudo pri pisanju diplomske naloge in družini za finančno pomoč pri študiju.

KAZALO

KAZALO SLIK

KAZALO TABEL

SEZNAM KRATIC

Kratica	Angleški izraz	Slovenski izraz
3DES	Triple Data Encryption Standard	trojni standard za šifriranje podatkov
A	DNS Address Record	zapis DNS naslova IPv4
AAAA	DNS IPv6 Addres Record	zapis DNS naslova IPv6
AD	Authenticated Data	potrjeni podatki (zastavica pri DNSSEC)
AES	Advanced Encryption Standard	napredni standard za šifriranje podatkov
Arnes	Academic and Research Network of Slovenia	Akademska in raziskovalna mreža Slovenije
CENTR	Council of European National Top Level Domain Registries	Združenje evropskih registrov vrhnjih domen
CNAME	Canonical Name Record	zapis DNS-vzdevka
CO	Crypto Officer	kriptografski uslužbenec
CRL	Certificate Revocation List	seznam preklicanih certifikatov
DLV	Domain Lookaside Validation	alternativno preverjanje domen
DNS	Domain Name System	sistem domenskih imen
DNSSEC	Domain Name System Security Extensions	varnost sistema domenskih imen
DOC	Department of Commerce	ministrstvo za trgovino
DOS	Denial of Service	odpoved storitve
DS	Delegation Signer	delegacija podpisnika
DURZ	Delibarately Unvalidatable Root Zone	neveljavno podpisana korenska cona
EDNS0	Extensions Mechanism for DNS version 0	razširitev protokola DNS različice 0
FIPS	Federal Information Processing Standard	zvezni standard za procesiranje podatkov
GCHQ	Government Communications Headquarters	sedež vlade za komunikacije
HINFO	Host Information	zapis DNS informacij o računalniku
HSM	Hardware Security Module	strojni varnostni modul
IANA	Internet Assigned Numbers Authority	organ za dodeljevanje številk v internetu
ICANN	Internet Corporation for Assigned Names and Numbers	internetna ustanova za dodeljevanje imen in številk
IDEA	International Data Encryption Algorithm	mednarodni algoritem za šifriranje podatkov
IETF	Internet Engineering Task Force	delovna skupina za internetno inženirstvo
IP	Internet Protocol	internetni protokol
IPSEC	Internet Protocol Security	varnostni protokol IP
ISC	Internet System Consortium	konzorcij internetnih sistemov
KASP	Key and Signature Policy	politika ključev in podpisov
KSK	Key Signing Key	ključ za podpisovanje ključa

LAN	Local Area Network	lokalno omrežje
MD5	Message Digest Algorithm 5	zgoščevalni algoritem 5
MITM	Man-in-the-middle	vrinjeni človek
MX	Mail Exchange Record	zapis e-poštnega strežnika DNS
NS	Name Server Record	zapis strežnika DNS
NSEC	Next Secure Record	naslednji varen zapis
NSEC3	NSEC Record version 3	zapis NSEC verzije 3
NTIA	National Telecommunications and Information Administration	nacionalna telekomunikacijska in informacijska administracija
NTP	Network Time Protocol	protokol omrežnega časa
PKI	Public Key Infrastructure	infrastruktura javnih ključev
PTR	Pointer Record	zapis DNS kazalca
RC4	Rivest Cipher 4	šifrant Rivest 4
RKSH	Recovery Key Share Holders	nosilci za obnavljanje ključev
RR	Resource Record	zapis DNS
RRSET	Resource Record Set	nabor zapisov DNS
RSA	Rivest, Shamir, Adelman	asimetrični šifrirni algoritem
SEP	Secure Entry Point	varna vstopna točka
SHA	Secure Hash Algorithm	varni zgoščevalni algoritem
SLD	Second Level Domain	domena na drugem nivoju
SOA	Start of Authority Record	zapis DNS začetka avtoritete
TCP	Transport Control Protocol	protokol za nadzor prenosa
TLD	Top Level Domain	domena vrhnjega sloja
TXT	Text Record	tekstovni zapis DNS
TTL	Time To Live	življenjska doba
UTC	Coordinated Universal Time	usklajeni univerzalni čas
WKS	Well Known Service Record	zapis poznanih storitev DNS
ZSK	Zone Signing Key	ključ za podpisovanje cone

POVZETEK

Danes brez protokola DNS svet ne bi bil tak, kakršen je. V internetu nam namesto zapletenih računalniških naslovov omogoča uporabo nam prijaznih imen. Pravilno delovanje DNS je ključno za delovanje interneta, zato je tudi pogosto tarča različnih napadov. Že dolgo je znano, da protokol DNS ne ustreza minimalnim varnostnim standardom, zato je potrebna rešitev, ki bo protokolu DNS dodala varnost.

Diplomska naloga obravnava razširitev obstoječega protokola DNS – DNSSEC. DNSSEC protokolu DNS zagotavlja varnost. Cilj naloge je predstaviti DNSSEC in ovrednotiti trenutno podprtost DNSSEC na različnih rešitvah za strežnike in odjemalce.

V prvem poglavju so opisane osnove protokola DNS in njegove ranljivosti, katerih razumevanje je potrebno za razumevanje delovanja DNSSEC.

V drugem poglavju sledi opis protokola DNSSEC. Opisani so varnostni mehanizmi, ki jih uporablja, novi zapisi, ki jih prinaša, kako deluje, kaj nam prinaša pa tudi katere so njegove slabosti in problemi, ki jih ne odpravlja. V tretjem poglavju sledi pregled uporabe DNSSEC v Sloveniji in svetu.

Zadnje poglavje predstavlja praktični del. Vzpostavil sem testno okolje, kjer so preizkušene različne strežniške rešitve DNSSEC, preizkušeno pa je tudi delovanje DNSSEC na operacijskih sistemih in aplikacijah, ki jih uporabljajo končni uporabniki.

Ugotovil sem, da je DNSSEC dobro zasnovan protokol, ki odpravlja večino varnostnih težav protokola DNS. Z uporabo sicer ni za hiteti, saj se je treba zavedati problemov, ki bodo povezani z bistveno večjo kompleksnostjo sistema DNS, ki jo prinaša DNSSEC. Tudi strežniške rešitve so že dokaj dobro implementirane, pogrešam pa boljšo podprtost DNSSEC s strani odjemalcev.

Diplomski nalogi je priložen še dodatek, ki vsebuje primer podpisane cone ter primere in postopke konfiguracije DNSSEC na različnih strežniških platformah.

KLJUČNE BESEDE

DNS, DNSSEC, varnost, kriptografija z javnim ključem, digitalni podpis, validiranje, strežnik

ABSTRACT

Today's world without the DNS protocol would not be as we know it. DNS protocol enables us the use of user-friendly names instead of complicated computer addresses. It is essential that the DNS protocol works correctly which is why it frequently becomes the target of all sorts of attacks. It has been known for a while that the DNS protocol does not suffice the minimal security standards, that is why we need a solution that will add security to the DNS protocol.

My diploma thesis deals with expansion of existing protocol DNS – DNSSEC. DNSSEC protocol enables security to the the DNS. The goal of my thesis is to represent the DNSSEC and evaluate the current support of the DNSSEC on different solutions for servers and clients.

The first chapter contains description of the basics of the DNSSEC protocol and it's weak points that must be understood for the understanding of how the DNSSEC actually works.

In the second chapter, there is a description of the DNSSEC. Safety mechanism, that DNSSEC uses, new records, that it introduces, how it works and what it brings, are also described. The chapter does not leave out the informations about the weak points of the DNSSEC and problems it does not solve. The third chapter deals with the use of the DNSSEC in Slovenia and across the world.

The last chapter represents a more practical part of my diploma thesis. I have set up a test environment where different server solutions for the DNSSEC are tried out. I have also tested the workings of the DNSSEC on the operating systems and applications that are used by the end costumers.

I have discovered that the DNSSEC is a well designed protocol that eliminates most of security issues of the DNS protocol. But there should not be any rush with it's use for we should be aware of the problems that will be connected with a higher complexity of the DNS system that the DNSSEC brings along. The server solutions are fairly well implanted but I miss a better support for the DNSSEC concerning the clients side.

Annex to my diploma thesis contains an example of signed zone and examples and procedures of configuration of the DNSSEC on different server platforms.

KEYWORDS

DNS, DNSSEC, security, public-key cryptography, digital signature, validation, server

UVOD

Internet brez protokola DNS ne bi bil internet, kakršnega si predstavljamo. DNS nam omogoča, da lahko namesto naslovov IP uporabljamo nam prijazna imena. Protokol DNS je ključen tudi za delovanje elektronske pošte. Več o protokolu DNS je opisano v poglavju 1 ter v [20] in [21].

Pisci protokola DNS niso kaj veliko razmišljali o varnosti. Komunikacija med strežniki DNS in strežniki ter odjemalci ni kriptirana, zato lahko napadalec, ki posluša na povezavi, ugotovi, s katerimi strežniki komuniciramo. Še huje, za integriteto podatkov ni poskrbljeno, zato lahko napadalec, ki se vrine v komunikacijo med nami in strežnikom DNS, sporočila DNS po želji spreminja. Tudi za avtentikacijo ni poskrbljeno, zato nikoli ne vemo, če smo sporočilo, ki smo ga dobili, prejeli zares od pravega strežnika.

Kljub vsem varnostnim luknjam je bila uporaba DNS nekaj običajnega in se uporabniki nismo spraševali, ali je to, kar na internetu vidimo, zares tisto, kar bi si želeli videti. Mit o varnosti DNS pa je leta 2008 dokončno razbil Dan Kaminsky z napadom aktivnega zastrupljanja medpomnilnika DNS [49]. Takrat je postalo jasno, da je protokol DNS treba prilagoditi sodobnim varnostnim standardom. Čas je, da rešitev DNSSEC, ki se je razvijala že od leta 1995, implementiramo in dejansko začnemo uporabljati. Več o napadih in slabostih protokola DNS je v diplomi opisano podpoglavju 1.5 ter v [8] in [14].

DNSSEC je razširitev obstoječega protokola DNS. Sistemu DNS dodaja mehanizme za zagotavljanje integritete in za izvajanje avtentikacije, ne pa tudi mehanizmov za zagotavljanje zaupnosti. DNSSEC torej poskrbi, da nas napadalec preko sistema DNS ne more preusmeriti na lažen računalnik. Ker zaupnosti ni, je prisluškovanje komunikaciji še vedno mogoče. DNSSEC uporablja varnostne mehanizme digitalnega podpisa, ki je kombinacija kriptografije z javnim ključem in zgoščevalnih funkcij. Več o DNSSEC je opisano v poglavju 2 in v [4], [5] in [6].

DNSSEC je na korenski coni in coni nekaterih vrhnjih domen že implementiran. Vprašanje, ki se zastavlja, pa je, kdaj in kateri končni uporabniki ga bodo začeli uporabljati. Zainteresiranost za DNSSEC ni ravno velika tudi zaradi njegove dokaj zapletene implementacije in nadalje zlasti zaradi zapletenega vzdrževanja. Na svetu še ni velikega števila strokovnjakov, ki bi razumeli in obvladali tehnologijo DNSSEC. Več o trenutnem stanju uporabe DNSSEC je opisano v poglavju 3 in v [19], [22], [23] in [24].

V diplomski nalogi sem najprej predstavil osnove sistema DNS, ki so ključne za razumevanje DNSSEC. Potem sem podrobno predstavil protokol DNSSEC vključno z njegovimi slabostmi, problemi, ki jih ne odpravlja, ter problemi, ki jih dodatno prinaša. Predstavil sem tudi osnove delovanja varnostnih mehanizmov, na katere se zanaša DNSSEC. Po opisu DNSSEC sledi pregled stanja uporabe DNSSEC v svetu in v Sloveniji. V praktičnem delu naloge sem postavil testno okolje DNSSEC in preizkusil delovanje DNSSEC na različnih strežniških tehnologijah. V testno okolje sem vključil tudi odjemalce, kjer je narejena primerjava podpore DNSSEC s strani treh najbolj priljubljenih operacijskih sistemov in aplikacij, ki tečejo v njih.

1 PROTOKOL DNS

1.1 STANDARDIZACIJA

Vsak računalnik v internetu ima svoj unikaten naslov IPv4 in/ali IPv6. Vendar si ljudje težko zapomnimo številke. Zato je bilo treba narediti sistem, ki bo pretvarjal imena, ki si jih ljudje lažje zapomnimo, v naslove IP. DNS počne ravno to: imena preslikuje v naslove IP in tudi obratno: naslove IP preslikuje v imena.

DNS je standardiziran z RFC 1034 (DOMAIN NAMES – CONCEPTS AND FACILITIES) in RFC 1035 (DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION). Dokumenta nosita letnico 1987, zato sta v tem času bila večkrat dopolnjena z mnogimi drugimi standardi, ki opisujejo potencialne varnostne težave v DNS, težave implementacije, mehanizme za dinamično posodabljanje imenskih strežnikov, za zavarovanje podatkov o coni in še mnoge druge.

1.2 ZAPISI DNS

Vsakemu vnosu v sistemu DNS pravimo zapis (RR, angl. Resource Record). Format zapisa DNS je sledeč:

NAME TTL CLASS TYPE (R DATA)

»Name« predstavlja ime zapisa DNS. »TTL« je vrednost, koliko časa naj se zapis DNS hrani v medpomnilniku strežnika DNS. »Class« predstavlja razred. »IN«, ki se običajno uporablja na tem mestu, predstavlja internetni razred. Ta zapis imamo, ker se lahko poleg internetnega protokola DNS uporabljajo tudi drugi protokoli. Tip zapisa nam pove, katera informacija je predstavljena v zapisu. Podatki, ki sledijo, so odvisni od tipa zapisa. Mogočih je več zapisov DNS istega tipa. Unikaten zapis DNS predstavljajo ime, razred in tip.

Različni tipi zapisov so definirani v RFC 1035. Sprva je obstajalo 9 tipov zapisov DNS, ki so predstavljeni v Tab. 1. [21]

Tab. 1: Tipi zapisov DNS

Tip zapisa	Pomen	Primer
A (angl. Address)	naslov IP	kozic.net. IN A 93.103.130.109
CNAME (angl. Canonical Name)	kaže na drugo ime DNS (vzdevek)	ftp.kozic.net. IN CNAME www.kozic.net.
HINFO (angl. Host Information)	informacije o strojni opremi strežnika (CPU, OS)	server.kozic.net. IN HINFO VAX-11/780 UNIX
MX (angl. Mail Exchanger)	poštni strežniki domene	kozic.net. IN MX 10 mx1.kozic.net. kozic.net. IN MX 20 mx2.kozic.net.
NS (angl. Name Server)	imena DNS avtoritativnih	kozic.net. IN NS

	strežnikov domene	ns1.kozic.net. kozic.net. IN NS ns2.kozic.net.
PTR (angl. Pointer)	naslov IP imena DNS (nasprotno od zapisa A)	109.130.103.93.in-addr.arpa. IN PTR mail.kozic.net.
SOA (angl. Start Of Authority)	strežnik DNS je avtoritativen strežnik za domeno	kozic.net. IN SOA ns1.kozic.net. hostmaster.kozic.net. (2010090801 ; Serijska številka 10800 ; Osveži po treh urah 1800 ; Ponovno poskusi čez pol ure 604800 ; Poteče čez en teden 1800) ; Najmanjši TTL pol dneva
TXT (angl. Text)	opis imena DNS	server.kozic.net. IN TXT »neki strežnik«
WKS (angl. Well-known services)	opis servisov, ki tečejo na ciljnem računalniku	server.kozic.net. IN WKS 93.103.130.109 TCP (ssh smtp domain www)

Nadaljnji dokumenti RFC so razširjali in še vedno razširjajo tipe zapisov DNS. RFC 3596 tako npr. uvaja zapis AAAA, ki kaže na naslov IPv6.

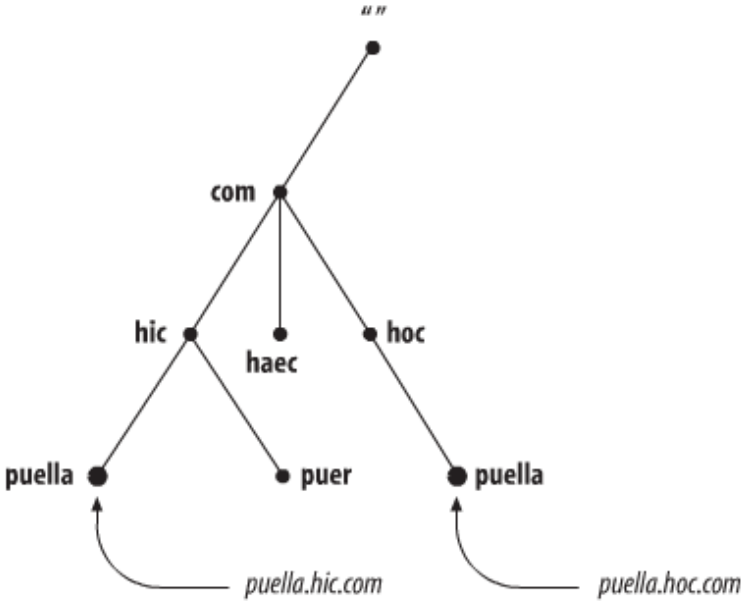
Zapis SOA nam pove, kateri strežnik DNS je avtoritativen za domeno. Dodatno vsebuje e-poštni naslov administratorja domene in serijsko številko domene. Poleg tega vsebuje čase TTL (angl. Time To Live). Časi TTL nam povejo, kako dolgo naj drugi strežniki DNS v svojem medpomnilniku hranijo zapise o domeni, preden ponovno povprašajo avtoritativni strežnik DNS po teh imenih. Sistem DNS je zasnovan tako, da strežniki ne poizvedujejo nenehno za istimi imeni, temveč pridobljene odgovore za določen čas hranijo v svojem medpomnilniku. TTL določi administrator domene. Premajhni časi TTL obremenjujejo strežnike DNS z dodatnimi zahtevki, če imamo velike TTL, pa je potrebnega veliko časa, preden se spremembe zapisov naše cone uveljavijo na drugih strežnikih. [2]

1.3 HIERARHIJA DNS

DNS je strukturiran hierarhično. Hierarhijo lahko predstavimo z drevesom. Vrhu drevesa pravimo korenska (angl. root) domena, ki nima svoje oznake.
Sledijo vrhnje (angl. Top Level Domain – TLD) domene. Le-te so razdeljene na:

- generične vrhnje domene (angl. generic top-level domains – gTLD): com, edu, gov, mil, net, org, int,
- vrhnje domene, ki pripadajo različnim državam (angl. country-code top-level domains – ccTLD): npr. si, gb, rs, au, at,
- nove generične vrhnje domene (angl. new top-level domains): aero, biz, coop, info, museum, name, pro (od leta 2000), jobs, travel (od leta 2005) in xxx (od leta 2011),
- internacionalizirane vrhnje domene (angl. internationalized top-level domains – IDN), ki omogočajo uporabo znakov, ki so specifični za določen jezik (npr. šumnikov, cirilice …),
- domena arpa, ki se uporablja za preslikavo naslovov IP v imena DNS.

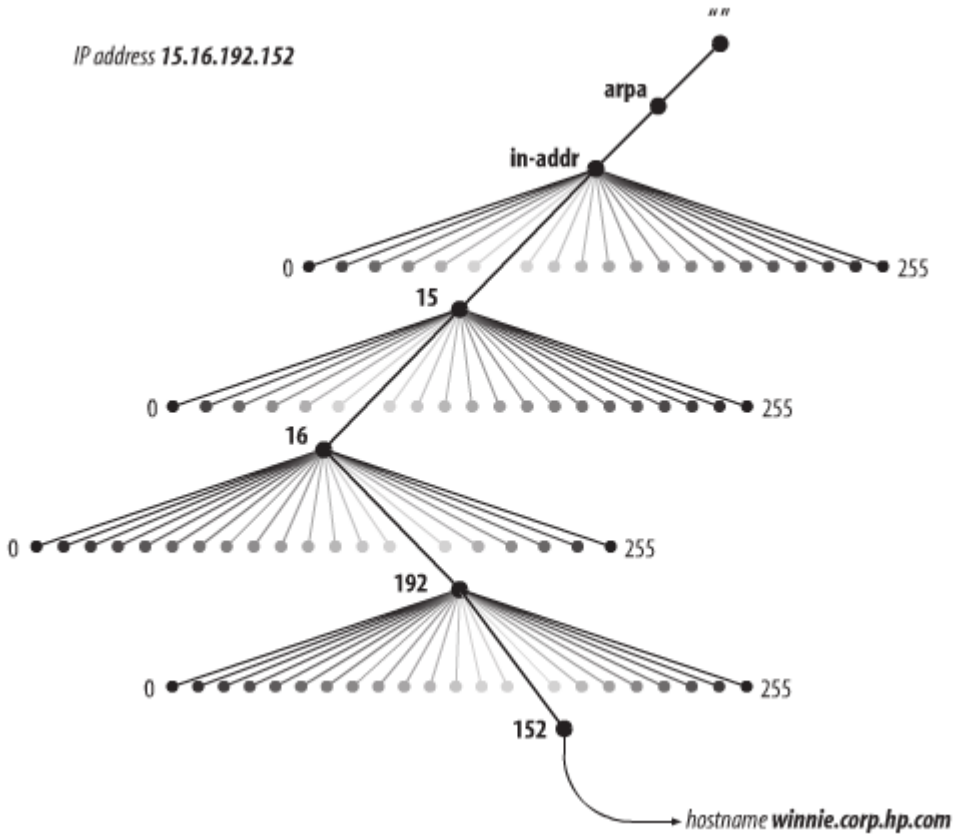
Sledijo domene, katerim pravimo SLD (angl. second-level domain). Primer cisco.com, kozic.net, ltfe.org, uni-lj.si. Drevo lahko gradimo naprej v globino.



Sl. 1: Drevo DNS [2]

Če bi v strukturi datotečnega sistema pot do vozlišča puer zapisali od vrhnjega vozlišča navzdol (/com/hic/puer – lat. com ta deček), bomo v sistemu DNS to pot zapisali od končnega vozlišča navzgor (puer.hic.com – lat. deček ta com). Ločilo med nivoji v sistemu DNS je pika. Vsakemu imenu na nižjem nivoju lahko pravimo poddomena. [2]

Preslikava naslovov IP v imena DNS, ki se ji pravi obratni DNS (angl. reverse DNS), je realizirana preko vrhnje domene arpa. Naslov IP, ki ga beremo od leve proti desni, v drevesu DNS gradimo od zgoraj navzdol, kar bo pomenilo, da bomo naslov IP, ko ga bomo brali v sistemu DNS, brali od spodaj navzgor, torej od desne proti levi. Primer: 212.101.143.177 v DNS zapišemo kot 177.143.101.212.in-addr.arpa. Ker sistem DNS omogoča uporabo več različnih razredov, tudi naslavljanje IPv6 v primeru obratnega DNS domena SLD določa, za kateri razred oz. za katero vrsto naslova IP gre. Za naslove IPv4 imamo domeno in-addr.arpa., za naslove IPv6 pa ip6.arpa.



Sl. 2: Hierarhija pri obratnem DNS [2]

Če preslikani naslovi IPv4 v imena DNS niso najlepši za oči, je zadeva še manj pregledna pri naslovih IPv6. Naslov IPv6 2a00:1368:1000:20::177 je v DNS predstavljen kot 7.7.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa (prazna mesta zapolnimo z ničlami). [2]

1.3.1 Korenska domena

V korenski domeni ima ICANN (angl. Internet Corporation for Assigned Names and Numbers) vlogo operaterja (angl. IANA Functions Operator). ICANN od strežnikov vrhnjih domen sprejema zahtevke po spremembah in te zahtevke tudi validira. Ko je validacija opravljena, ICANN vpraša administratorja korenske domene za avtorizacijo. Administrator korenske domene je NTIA (angl. National Telecommunications and Information Administration), ki je agencija znotraj ameriškega ministrstva za trgovino (angl. Department of Commerce – DoC). Ko administrator korenske domene avtorizira zahtevek, ICANN njegovo kopijo pošlje vzdrževalcu korenske domene, ki spremembo uveljavi. Vzdrževalec korenske domene je podjetje Verisign. [19]

Imamo 13 korenskih strežnikov, ki se nahajajo na rezervirani domeni root-servers.net. Strežniki so poimenovani po abecedi od A.root-servers.net do M.root-servers.net. Vsak zapis DNS kaže na en naslov IP, vendar se pod tem enim naslovom IP nahaja gruča fizičnih

strežnikov. Korenski strežniki so razporejeni po celem svetu in so v lasti različnih organizacij in podjetij.[1] [1]

Večina korenskih strežnikov ima sedaj tudi IPv6-naslove. Pri IPv6 je podan anycast naslov. To je naslov skupine računalnikov, ki nas pripelje do najbližjega med njimi. Večina korenskih strežnikov uporablja tehnologijo anycast tudi pri IPv4, čeprav le-ta ne pozna posebnih anycast naslovov.

1.3.2 Operaterji registra in registrarji

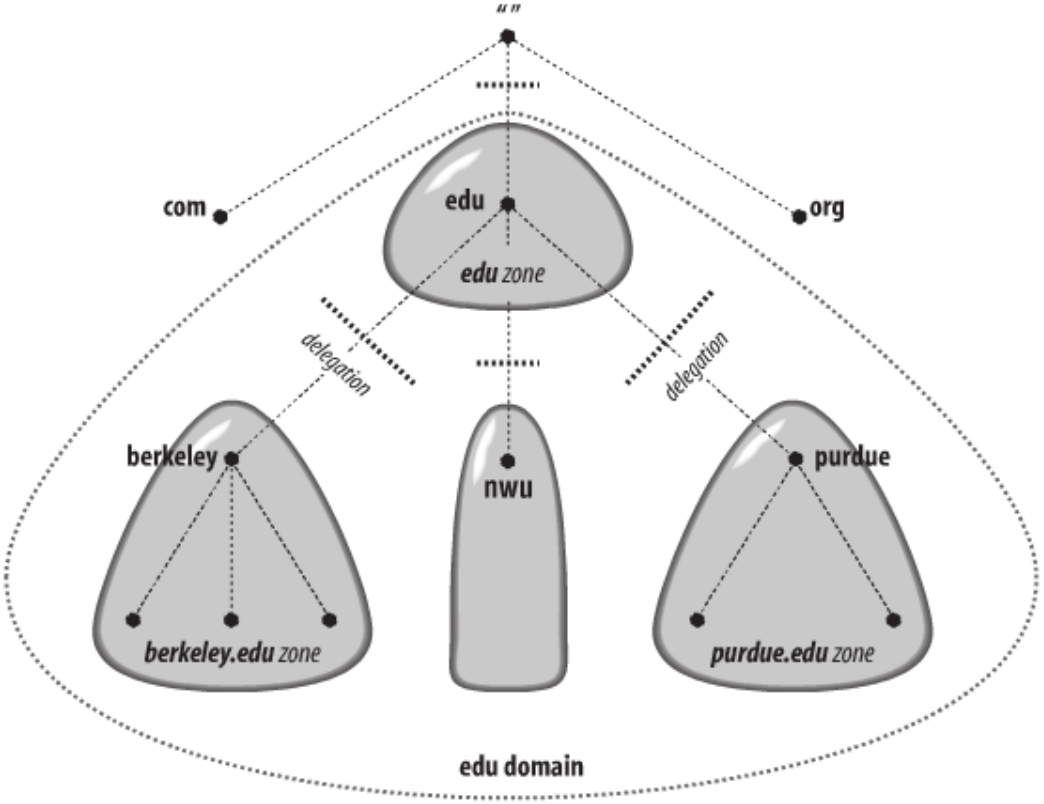
Korenski strežniki hranijo informacije, kje se nahajajo strežniki vrhnjih (TLD) domen. Strežnike vrhnjih domen upravljajo različni registri (angl. Registry Operator). Vsaka vrhnja domena ima enega registra. Register ima pogodbo z ICANN in skrbi za strežnike DNS za svojo vrhnjo domeno.

Registri imajo načine zakupa oz. pridobitve domen SLD za končne uporabnike rešene različno. Pri nekaterih registrih lahko uporabnik dobi domene neposredno pri njih, pri nekaterih preko registrarjev, pri nekaterih pa pri obojih. Registrarji so organizacije in podjetja, pri katerih je mogoče zakupiti ali dobiti domene SLD. So posredniki med uporabnikom in registrom. Registrarje za domene gTLD prav tako mora odobriti ICANN. [1] Je zatorej internet zares brez lastnika?

1.3.3 Strežniki DNS

Zapise domen strežniki DNS hranijo v t. i. conah (angl. zone file). Cone hranimo na različnih strežnikih DNS. Delegiranje pomeni, da bomo cono od določene poddomene naprej hranili na drugem strežniku. Poddomene že delegirane cone lahko nadalje delegiramo na drug strežnik. Korenski strežniki delegirajo vrhnje domene strežnikom vrhnjih domen, slednji delegirajo svoje poddomene naprej strežnikom, zadolženim za posamezno domeno SLD. Tako je sistem DNS razbremenjen. Cona in poddomena sta si nekaj podobnega, vendar poddomena ni nujno vedno cona. [2]

[1] Korenskih strežnikov je 13 zaradi originalne omejitve velikosti paketa DNS. Ta je 512 B in če želimo odgovor spraviti v en datagram UDP, je največje število »povezav« do strežnikov, ki so lahko v odgovor vključene, 13.



Sl. 3: Cone in delegacija [2]

1.3.3.1 Vrste strežnikov DNS

Obstajajo naslednje vrste strežnikov DNS:

- Avtoritativni strežnik DNS: strežnik, ki ima naloženo določeno cono:
  - primarni strežnik DNS (angl. primary master): strežnik, na katerem vnašamo in spreminjamo zapise določene cone;
  - sekundarni strežnik DNS (angl. secondary master ali slave): strežnik, ki si pridobi in posodablja podatke o določeni coni s primarnega strežnika DNS. Sekundarnih strežnikov je lahko več. Tako primarni kakor sekundarni strežnik DNS sta avtoritativna strežnika (angl. authoritative DNS server). Za vsako cono je priporočeno imeti vsaj dva avtoritativna strežnika, ki naj ne bi bila v istem podomrežju.
- Rekurzivni strežnik (angl. recursive resolver) – strežnik, ki poizveduje po hierarhiji DNS:
  - Caching Name Server – strežnik, ki poizveduje po hierarhiji DNS in določen čas hrani informacije o razrešenih imenih DNS;
  - Forwarding (Proxy) Name Server – strežnik, ki za razreševanje imena DNS ne poizveduje po hierarhiji DNS, ampak poizveduje preko drugih rekurzivnih strežnikov DNS. Po poizvedovanju prav tako v svojem medpomnilniku hrani informacije o razrešenih imenih DNS.

Mogoče so tudi mešane postavitve, kjer je strežnik za nekatere domene avtoritativen, poleg tega pa opravlja funkcijo rekurzivnega strežnika. Stub resolver je knjižnica na operacijskem

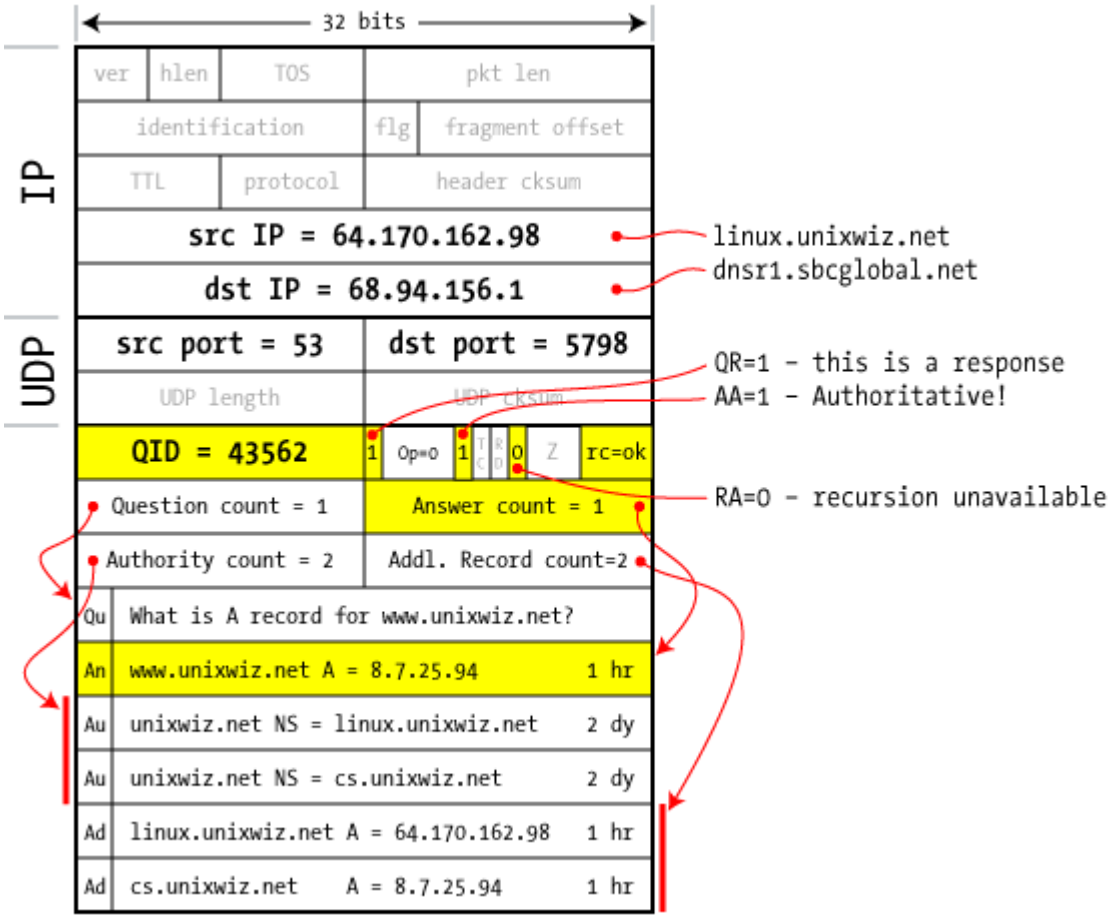
sistemu, ki skrbi za razreševanje imen DNS. Za razreševanje imen uporablja rekurzivni strežnik.

DNS uporablja transportna protokola UDP in TCP. Za poizvedovanje se večinoma uporablja protokol UDP. Ker pri njem ni treba vzpostaviti povezave, je komunikacija hitrejša, strežniki so manj obremenjeni, poleg tega pa je UDP praktičen, saj so sporočila DNS po velikosti majhna. TCP se za poizvedovanje uporabi v primeru, če UDP-komunikacija ni mogoča. Največja možna velikost sporočila DNS v primeru uporabe UDP je 512 B. Če je sporočilo večje od 512 B, bomo uporabili TCP (ali pa razširitev EDNS, ki je opisana v 2. poglavju). Za prenos con iz primarnega na sekundarne strežnike DNS se uporablja protokol TCP. Vrata, na katerih posluša strežnik DNS, so TCP in UDP vrata 53. [1]

1.4 POIZVEDOVANJE DNS

1.4.1 Format sporočila DNS

Na Sl. 4 je prikaz izgleda sporočila DNS. Prikazano je sporočilo odgovora na zahtevek DNS. Identifikacija (angl. Identification) oz. ID-transakcije (angl. Transaction ID; na sliki označen kot QID) poveže zahtevek z odgovorom. Da lahko povežemo zahtevek z odgovorom, se morajo točneje ujemati ime zahtevka, razred zahtevka, tip zahtevka in identifikacija. Polje zastavic, ki sledi polju identifikacije, je zelo pomembno, saj le-te določajo, ali gre za zahtevek ali odgovor, če je zahtevek rekurziven. Naslednja polja predstavljajo število vprašanj in odgovorov, ki so podani nadalje v samem sporočilu DNS. Na tem mestu se glava DNS konča. Sledijo vprašanje po imenu DNS (Question section; Qu), odgovor na vprašanje (Answer Section; An). Sledijo informacije o avtoritativnih strežnikih DNS za domeno, po kateri smo poizvedovali (Authority Section; Au). Za njimi sledijo dodatne informacije (Additional section; Ad), ki ponavadi vsebujejo razrešene naslove DNS avtoritativnih strežnikov DNS-domene. Sekciji dodatnih informacij pravimo tudi lepljivi zapisi (angl. glue records).



Sl. 4: Format sporočila DNS [49]

1.4.2 Potek poizvedovanja

Oglejmo si potek poizvedovanja po imenu DNS *ltfe.kozic.net*. Omenjeno domensko ime ima tako naslov IPv4 kakor naslov IPv6.

Računalnik, na katerem poizvedujemo, uporablja strežnik DNS, ki bo zanj opravil vso poizvedovanje čez hierarhijo DNS in mu bo vrnil rezultat njegove poizvedbe DNS. Takemu strežniku DNS pravimo **rekurzivni strežnik** (angl. recursive resolver) [2]. Rezultat bo vseboval naslov IP razrešene domenske poizvedbe, če pa ta ne obstaja, bo vrnil napako. Pravimo, da je računalnik svojemu strežniku DNS poslal **rekurzivni** zahtevek. Rekurzivni zahtevek določa, da mora strežnik DNS opraviti celotno poizvedovanje čez hierarhijo DNS, dokler naslova ne razreši.

Strežnik DNS bo najprej pogledal v svoj medpomnilnik. Če je že kdo prej poizvedoval po tem imenu, ima vnos shranjen in ga lahko odjemalcu takoj vrne. Odgovori DNS se za določen čas TTL shranijo v medpomnilniku. Čas TTL nastavi administrator domene pri posameznih vnosih.

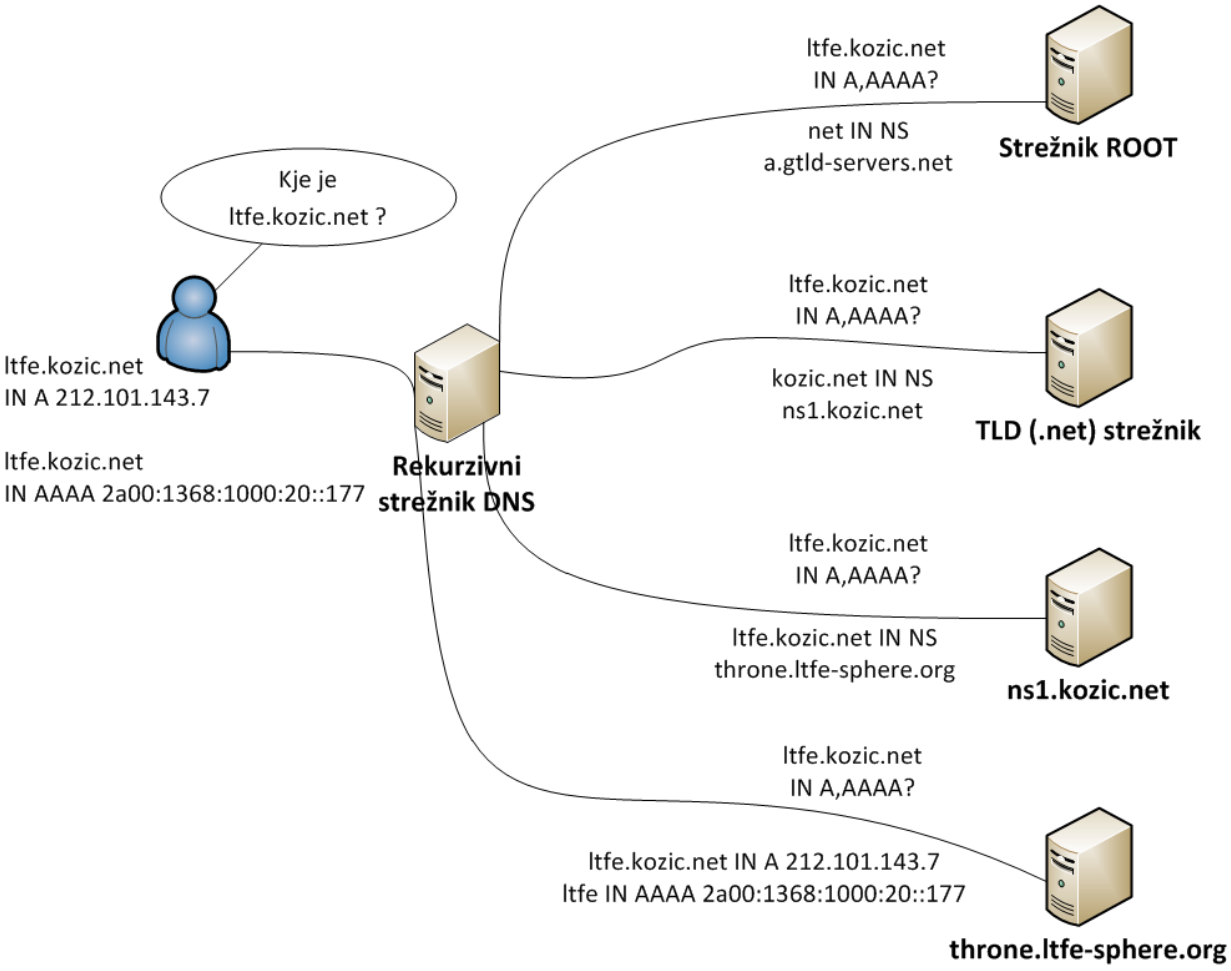
Če strežnik DNS nima nobenih podatkov o povpraševanem domenskem imenu, bo šel v **iterativno** poizvedovanje [2]. Najprej bo vzpostavil kontakt z enim korenskih (angl. root)

strežnikov. Vprašal ga bo, kakšen je naslov domene *ltfe.kozic.net* (konkretneje bo vprašal po zapisu A za naslov IPv4 in po zapisu AAAA za naslov IPv6). Korenski strežnik rekurzivnemu strežniku ne bo podal dokončnega odgovora, odgovoril pa bo, kateri strežniki DNS so avtoritativni za domeno net. Nato bo rekurzivni strežnik ponovil celotno poizvedbo po imenu *ltfe.kozic.net*, le da bo vprašanje naslovil enemu od strežnikov avtoritativnih za domeno net. Avtoritativni strežnik za domeno net bo rekurzivni strežnik napotil korak bližje. Odgovoril mu bo, kateri strežniki so avtoritativni za domeno *kozic.net*.

Avtoritativni strežnik za domeno kozic.net (ns1.kozic.net) poizvedovalcu ponovno ne bo dal končnega odgovora, ampak ga bo poslal na strežnik throne.ltfe-sphere.org, ki je avtorirativen za delegirano domeno *ltfe.kozic.net*. Če domena *ltfe.kozic.net* ne bi bila delegirana, bi strežnik throne.ltfe-sphere.org poslal končen odgovor na poizvedbo. Iz samega domenskega imena ni mogoče razbrati, katere domene so delegirane.

Strežnik throne.ltfe-sphere.org bo odgovoril na vprašanje po naslovu imena DNS *ltfe.kozic.net*. Odgovoril bo, da se ta nahaja na naslovu IP 212.101.143.177, njegov naslov IPv6 pa je 2a00:1368:1000:20::177.

Ko bo rekurzivni strežnik DNS, na katerem je odjemalec poizvedoval, dobil odgovor, ga bo posredoval odjemalcu.



Sl. 5: Poizvedovanje po hierarhiji DNS za domensko ime ltfe.kozic.net

1.5 RANLJIVOSTI PROTOKOLA DNS

Napadalec kompromitira vsak strežniški sistem vključno s sistemi strežnikov DNS, in sicer na način, da vanje vdre. To mu uspe zaradi slabe implementacije strežniških programov. Obstajajo t. i. buffer overflow napadi, ko z nepravilno formatiranimi podatki napadalec prekorači medpomnilnik strežnika. Na ta način lahko zruši strežniški program ali sam sistem, kar je ena vrsta DOS (angl. Denial of Service) napada, v hujšem primeru pa napadalec dobi dostop do ciljnega strežnika. Sistem DNS je tako kot vse storitve v internetu občutljiv na (D)DOS (angl. (Distributed) Denial of Service) napade. To so napadi, ko napadalec s skupino okuženih računalnikov, ki jih kontrolira, sproži ogromno nelegalnih zahtevkov na ciljni sistem in tako preobremeni sistem ali internetno povezavo. [1]

Slabost, na katero se bom osredotočil, pa je teoretična slabost samega protokola. DNS je nastal leta 1987 in ob njegovi implementaciji niso razmišljali o varnosti, tako je protokol DNS praktično brez kakršnihkoli varnostnih mehanizmov. Komercialne storitve preko interneta pač še niso bile razvite.

1.5.1 Zloraba zaupanja

Napadalec želi promet uporabnika, namenjen ciljnemu strežniku, namesto na ta strežnik preusmeriti nase. Ko ga preusmeri, lahko spremlja promet uporabnika, navadno pa to naredi z namenom, da se dokoplje do občutljivih podatkov, kakor so gesla uporabnika. Primer takih najobčutljivejših podatkov so podatki za dostop do elektronskega bančništva. Obstaja več načinov, kako napadalec izvede tak napad, eden izmed najbolj priročnih je preko sistema DNS. Napadalec prepriča uporabnika, da se naslov domene, npr. *klik.nlb.si*, nahaja na naslovu IP napadalčevega računalnika. Če napadalec kompromitira strežnik DNS, so na napačen naslov usmerjeni vsi uporabniki, ki za razreševanje naslovov DNS uporabljajo ta strežnik. Zelo velika škoda nastane, če napadalcu uspe kompromitirati strežnik DNS, ki ga uporablja veliko uporabnikov, npr. strežnik ponudnika internetnih storitev. Napadalec, ki vdre v sistem, lahko delovanje strežnika DNS prilagaja po lastni želji. Na računalniku imajo uporabniki običajno vnesena naslova dveh strežnikov DNS, ki jima morajo zaupati. Ko so strežniki DNS, ki jim zaupamo, kompromitirani, temu pravimo zloraba zaupanja (angl. Betrayal By Trusted Server). Zloraba zaupanja je mogoča, ker je DNS zasnovan tako, da poizvedovanje za uporabnika opravlja strežnik. Poleg tega v protokolu DNS nima uporabnik na voljo nobenih mehanizmov, s katerimi bi lahko verificiral odgovor strežnika. [1]

1.5.2 Man-in-the-middle napadi

MITM (angl. Man-in-the-middle) predstavlja napade, ko se napadalec vrine med komunikacijo dveh ali več različnih računalnikov. Ko napadalcu to uspe, lahko zgolj pasivno prisluškuje komunikaciji, lahko pa tudi aktivno spreminja njeno vsebino. Protokol DNS je občutljiv na MITM-napade in nima nobene zaščite pred njimi. Pri DNS uporabnika načeloma ne skrbi, da bi napadalec videl katere naslove DNS razrešuje. Problematično pa je, da lahko vrinjen napadalec poljubno spreminja odgovore, uporabnik pa nima mehanizmov, da bi lahko preveril, če so odgovori DNS pristni. MITM-napade je najlažje narediti v lokalnem omrežju

(LAN; angl. Local Area Network), kjer je implementacija zaščite pred njimi zapletena. [8][14]

1.5.3 Zastrupljanje medpomnilnika DNS

Že omenjena načina, kako napadalec manipulira z DNS, sta, da v strežnik preprosto vdre ali pa se mu uspe vriniti med komunikacijo strežnikov DNS. Drug način pa je, da napadalec zastrupi medpomnilnik DNS (angl. DNS Cache Poisoning). Kot vemo, ima strežnik DNS za čas vrednosti TTL vse razrešene naslove shranjene v medpomnilniku. Če je ta medpomnilnik zastrupljen, so vsi odgovori strežnika DNS za povpraševani naslov domene v tem času TTL napačni.

1.5.3.1 Sleparjenje DNS

Strežnik DNS dobi zahtevo uporabnika po razrešitvi določenega imena DNS. Če ima to ime v lastnem medpomnilniku, mu odgovori kar iz njega. Če pa tega nima, začne spraševati naprej po hierarhiji DNS. Ko strežnik DNS začne s spraševanjem, mu lahko napadalec podtakne lažen odgovor. S svojim odgovorom mora prehiteti legalen strežnik DNS, poleg tega se mora predstaviti z njegovim naslovom IP. Temu, da napadalec odgovarja namesto legalnega strežnika DNS na način, da se predstavlja kot on, pravimo sleparjenje DNS (angl. DNS spoofing).

Kakor sem omenil, ima DNS v glavi svojega sporočila 16-bitni identifikator transakcije. Na ta način strežnik DNS poveže poizvedbo z odgovorom. Za povezavo poizvedbe z odgovorom se morajo poleg identifikatorja transakcije dodatno ujemati vprašanje poizvedbe, razred poizvedbe in vrsta poizvedbe. Napadalec sam ve, kateri odgovor hoče podtakniti strežniku DNS, zato pozna ostale tri parametre vprašanja. Ponarejanje identifikatorja transakcije sprva ni bilo problematično, saj je bil identifikator preprosto realiziran kot števec. Ob vsakem zahtevku DNS je strežnik identifikator povečal za 1. Proizvajalci strežnikov DNS so potem identifikator realizirali kot poljubno (angl. random) število. S tem je bil velik del problema odpravljen, napad praktično nemogoč (napadalec bi moral strežnik nenehno zasuvati z lažnimi odgovori in upati na srečo). Problem je sicer še vedno, saj je identifikator transakcije 16-bitno število, kar predstavlja 65535 možnosti, ki dandanes ne predstavlja tako velikega prostora. [8][14]

1.5.3.2 Izvorna vrata UDP

Za uspešno podtikanje odgovora na zahtevek se morajo ujemati tudi informacije nižjeležečih protokolov. Pri protokolu IP sta to izvorni (angl. source) in ponorni (angl. destionation) naslov IP. Naslova IP ni težko ponarediti, zlasti zaradi tega, ker DNS na transportni plasti uporablja protokol UDP, ki ni povezavno orientiran. Pri povezavno orientiranem protokolu TCP je za uspešno vzpostavitev komunikacije treba vzpostaviti promet v obe strani. Če napadalec ponaredi izvorni naslov paketa IP, odgovora najverjetneje ne bo dobil on. Tako se komunikacija med napadalcem in strežnikom ne bo nikoli vzpostavila, kar bi pa bil ob uporabi protokola TCP predpogoj, da bi si napadalec in strežnik DNS lahko začela izmenjevati sporočila DNS. Na transportni plasti smo torej pri protokolu UDP, kjer se za

uspešno komunikacijo med napadalcem in strežnikom DNS morajo ujemati izvorna in ponorna vrata, ki so predstavljena s 16-bitnim številom. Ker je cilj poizvedbe nek drug strežnik, morajo ponorna vrata biti vrata UDP 53, izvorna pa so lahko poljubna, običajno med 1024 in 65535 (prvih 1023 vrat je na sistemu Unix rezerviranih za strežniške programe). Prvoten problem realizacije strežnikov DNS je bil, da so izvorna vrata prav tako bila vrata UDP 53. Potvarjanje glave UDP tako sploh ni bil problem, kar je bilo treba uganiti, je le identifikator transakcije.

Danes so strežniki DNS implementirani tako, da uporabljajo naključna izvorna vrata UDP, kakor tudi naključni identifikator transakcije. [14]

1.5.3.3 Naključnost

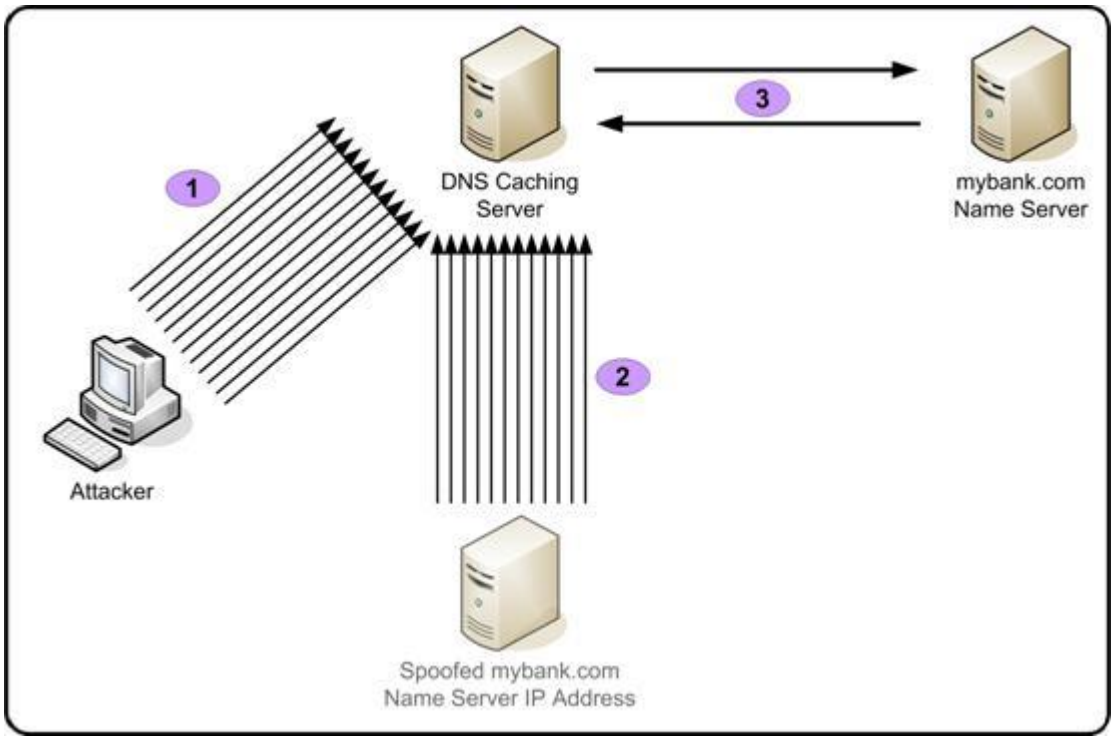
Naključnost je pri računalništvu vedno znova problem. Računalnik mora naključna števila izračunati in ker jih računa, niso več naključna. Prave naključnosti pri računalniku torej ni. Govorimo o psevdonaključnih številih.[2]

To niti ne bi bil tolikšen problem, ko spet ne bi različni strežniki DNS imeli zelo slabo implementirane naključnosti. Tako so generirali precej predvidljive identifikatorje transakcije DNS, imeli pa so tudi pomanjkljivosti pri določanju naključnih izvornih vrat UDP ali pa so uporabljali celo fiksna vrata za vse izhodne zahtevke DNS. Ena takih varnostnih lukenj (US CERT VU#800113 [50]), ki je zadevala oba najbolj razširjena strežnika DNS – Bind in Windows Server DNS, se je pojavila leta 2008, kar niti ni toliko nazaj.

1.5.3.4 Rojstnodnevni napad

Strežnik DNS pričakuje odgovor šele, ko drugemu strežniku pošlje zahtevo po poizvedovanju. Če bo uporabnik spraševal po zapisu DNS in strežnik ne bo poznal odgovora, bo strežnik moral sprožiti zahtevo. Pri rojstnodnevnem napadu (angl. birthday attack) napadalec pošilja strežniku nešteto zahtev in na ta način posredno sproži poizvedovanje s strani strežnika. Napadalec hkrati odgovarja na te poizvedbe. Prej ali slej (že pri povprečno 700 takih poskusih) bo prišlo do kolizije (torej, da se bodo parametri zahteve in odgovora ujemali) in tako bo v medpomnilnik strežnika DNS podtaknjen lažen vnos. Zaradi medpomnilnika DNS si sicer strežnik DNS zapomni poprejšnje odgovore, za katere obstaja velika verjetnost, da so legalni, zato strežnik določenega imena DNS ne bo šel razreševat vedno znova in znova. Pomembno je torej, da napadalec prehiti odgovor legitimnega strežnika DNS. [1]

[2] Če v formulo za računanje naključnih števil dodamo še kako zares naključno vrednost, kar lahko predstavljajo premiki miške, čas med pritiski tipk na tipkovnici, nihanja napetosti na matični plošči ipd., dobimo zelo dober približek dejanski naključnosti. Taki naključni generatorji se uporabljajo za kriptografijo.



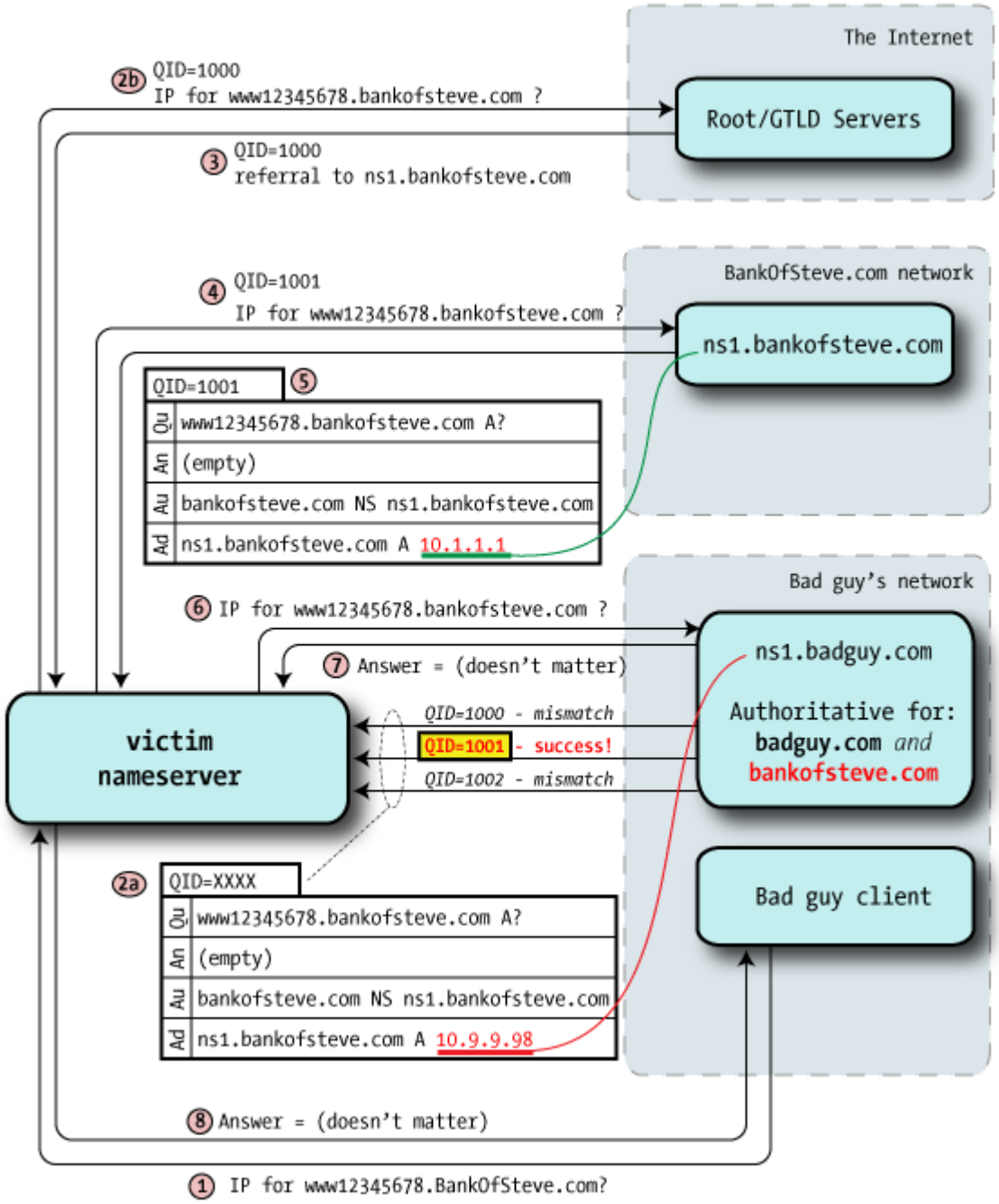
Sl. 6: Rojstnodnevni napad [52]

1.5.4 Aktivno zastrupljanje medpomnilnika DNS

Do leta 1995 je bil možen napad pasivnega zastrupljanja medpomnilnika DNS. Napadalec je imel postavljen svoj legitimen avtoritativni strežnik za eno izmed njegov domen, npr. napad.si. Ko je rekurzivni strežnik poizvedoval po domenskem imenu www.napad.si, je napadalčev strežnik vrnil odgovor za povpraševano ime, pod dodatnimi informacijami (angl. Additional section) pa je ponudil razrešen naslov domene, za katero ni bil avtoritativen, npr. www.yahoo.com, ki je seveda kazal na naslov IP lažnega strežnika. Rekurzivni strežnik DNS si je dodatno informacijo nekritično shranil v svoj medpomnilnik. Čas TTL podtaknjenega imena je seveda bil čim večji, lahko tudi 30 dni. Vsi strežniki DNS so po odkritju tega napada leta 1995 bili pokrpani, tako napad ni več mogoč. [14]

Leta 2008 pa je Dan Kaminsky odkril napad na DNS, ki je neke vrste kombinacija rojstnodnevnega napada in napada pasivnega zastrupljanja medpomnilnika DNS. Izvesti rojstnodnevni napad je precej težko opravilo, saj mora napadalec v zelo kratkem času prehiteti odgovor legitimnega strežnika DNS. Če napadalec to poskuša za veliko število različnih naslovov, mu bo prej ali slej uspelo. Napadalec lahko podtakne lažni naslov tudi za domensko ime, ki ne obstaja. Strežnik po povpraševanju po naslovu neobstaja.yahoo.com odgovori z negativnim odgovorom (domensko ime ne obstaja), po uspešnem napadu pa bo odgovoril s pozitivnim odgovorom, z naslovom IP, ki obstaja. Napad je dokaj nesmiseln, saj se uporabnik ne bo odpravil na naslov, ki ne obstaja. Ampak Dan Kaminsky se je tistega dne malo poigraval, se lotil rekurzivnega strežnika DNS in mu poskušal zastrupiti medpomnilnik za vsaj en vnos neke domene. Loteval se je naslovov, ki ne obstajajo, najprej 001.domena.si, potem 002. domena.si in tako naprej. Pri enem poskusu mu je uspelo. Mimogrede pa je odkril eno stvar. Odgovor DNS poleg razrešenega naslova vsebuje tudi informacije o avtoritativnih strežnikih DNS in razrešene naslove avtoritativnih strežnikov v polju dodatnih informacij.

Odgovoru je podtaknil še narobe razrešen naslov avtoritativnega strežnika za to domeno. Od tistega trenutka naprej je strežnik DNS za domeno domena.si poizvedoval na napadalčevem strežniku DNS. Napadalcu se ponovno splača, da TTL odgovora zapisa NS nastavi na 30 dni. Prednost tega napada je očitna slabost protokola DNS, saj so v odgovoru za vsako ime DNS ponujene tudi informacije o avtoritativnih strežnikih za domeno, kateri ta zapis pripada. [49]



Sl. 7: Primer napada aktivnega zastrupljanja medpomnilnika DNS [49]

Prišlo je do konferenc, kjer so se zbrali največji strokovnjaki s področja varnosti in sistema DNS. Ugotovili so eno. Zaščite pred napadom aktivnega zastrupljanja medpomnilnika DNS,

ki ga je odkril in predstavil javnosti Dan Kaminsky, ni. Kot rešitev za zagotavljanje lažnega občutka varnosti so uporabnikom ponudili popravke za različne strežnike DNS, ki izboljšujejo generiranje naključnih števil v polju za identifikacijo transakcije v glavi DNS in pri izbiri izvornih vrat UDP. Dodali so še nekaj dodatnih priporočil:

- za zapise NS nastaviti čim večji TTL,
- nastaviti rekurzivne strežnike DNS, da preden zaupajo odgovoru DNS, ta odgovor preverijo na vsaj dveh avtoritativnih strežnikih DNS.

Domene, ki potrebujejo visoko dinamičnost, si ne morejo privoščiti visokih časov TTL (npr. domena 24ur.com uporablja za zapis NS TTL 1 minuto).

Od leta 2008 je sistem DNS z varnostnega vidika ogrožen v tolikšni meri, da mu ne moremo več zaupati. Zanj obstaja rešitev – DNSSEC, ki je podrobneje opisana v nadaljevanju.

2 DNSSEC

2.1 ZAKAJ DNSSEC

Kakor sem že omenil, je DNS v osnovi dokaj star protokol brez kakršnihkoli varnostnih mehanizmov. Na DNS obstaja nekaj napadov, ki so mogoči zaradi slabe varnostne zasnove samega protokola. Ko preko strežnika DNS razrešujemo posamezno domensko ime, ne moremo biti nikoli prepričani, ali je bilo ime razrešeno pravilno oziroma ali smo res prišli na naslov, na katerega smo bili namenjeni. Pri kritičnih aplikacijah si pomagamo z ostalimi varnostnimi rešitvami, kakor sta npr. protokola SSL in TLS, ki skrbita za šifriranje na transportni in aplikacijski plasti ter ki s pomočjo infrastrukture javnih ključev PKI (angl. Public Key Infrastructure) preverita tudi istovetnost strežnika. Vendar večina internetnih aplikacij, zlasti spletni brskalniki, še vedno večinoma uporablja nekriptirane povezave, pri njihovi uporabi tako nikoli ne moremo z gotovostjo trditi, da smo zares povezani s strežnikom, s katerim mislimo, da smo. V ta namen se je pojavila potreba po varnostni razširitvi protokola DNS na način, da se bo dalo preveriti istovetnost njegovih odgovorov.

DNSSEC (angl. Domain Name System Security Extensions) je modifikacija obstoječega protokola DNS, ki zagotavlja dokazovanje avtentičnosti podatkov domene z digitalnim podpisovanjem le-teh. DNSSEC zagotavlja zgolj avtentičnost zapisov DNS, zaupnosti komunikacije nam ne nudi. Prisluškovanje komunikaciji je mogoče, ni pa mogoče spreminjanje in potvarjanje odgovorov DNS. [1][2]

DNSSEC nam zagotavlja zaščito pred različnimi napadi na DNS, katerih namen je zastrupiti medpomnilnik DNS. Varnostno preverjanje avtentičnosti zapisov DNS bo poskrbelo, da bodo nelegalni odgovori ustrezno prepoznani, strežnik DNS takih podatkov ne bo upošteval. Zagotavlja nam tudi potrebno zaščito pred MITM-napadi. Načeloma nas ne skrbi toliko, ali vrinjen napadalec vidi, katere naslove razrešujemo, in po tem sklepa, npr. po katerih straneh brskamo, bolj nas skrbi, da napadalcu ne bi uspelo, da bi nas zavedel na napačno stran. Pred tem nas pa DNSSEC uspešno varuje.

2.2 STANDARDIZACIJA

Diskusija po potrebi DNSSEC se je v IETF (angl. Internet Engineering Task Force) začela leta 1995, ko je bil javno objavljen članek Stevena Bellovina, ki je odkril nekaj velikih lukenj v protokolu DNS [10]. Če omenim nekaj zgodovine [33] je bil prvi standard DNSSEC RFC 2065 (1997). Prva verzija, mišljena za postavitev, je bila zapisana v RFC 2535 (1999). Od leta 1999 do 2001 so zadeve okrog implementacije DNSSEC mirovale. Leta 2001 so ugotovili, da ima RFC 2535 hude, zlasti zmogljivostne pomanjkljivosti pri hierarhični gradnji zaupanja (strežnik DNS za višjo domeno bi moral sam podpisovati celo domeno na nižjem nivoju).

Od leta 2001 do 2005 so RFC 2535 izboljševali. Leta 2005 je nastal standard, ki se uporablja in implementira tudi danes, imenovan *DNSSEC-bis* (beseda *bis* za imenom standarda pomeni drugo verzijo standarda [1]). Standardiziran je v RFC 4033 do RFC 4035:

- **RFC 4033** DNS Security Introduction and Requirements

- **RFC 4034** Resource Records for the DNS Security Extensions
- **RFC 4035** Protocol modifications for the DNS Security Extensions

DNSSEC-bis se še razvija in dopolnjuje. Dopolnjen je bil z naslednjimi standardi RFC:

- RFC 4509 (leta 2006), ki omogoča uporabo SHA-256 v zapisu DS,
- RFC 5011 (2007), ki določa samodejno posodabljanje t. i. puščic zaupanja,
- RFC 5155 (2008), ki uvaja nov zapis NSEC3,
- RFC 5702 (2009), ki poleg osnovnih SHA-1 in MD5 omogoča uporabo zgoščevalnih algoritmov SHA-2.

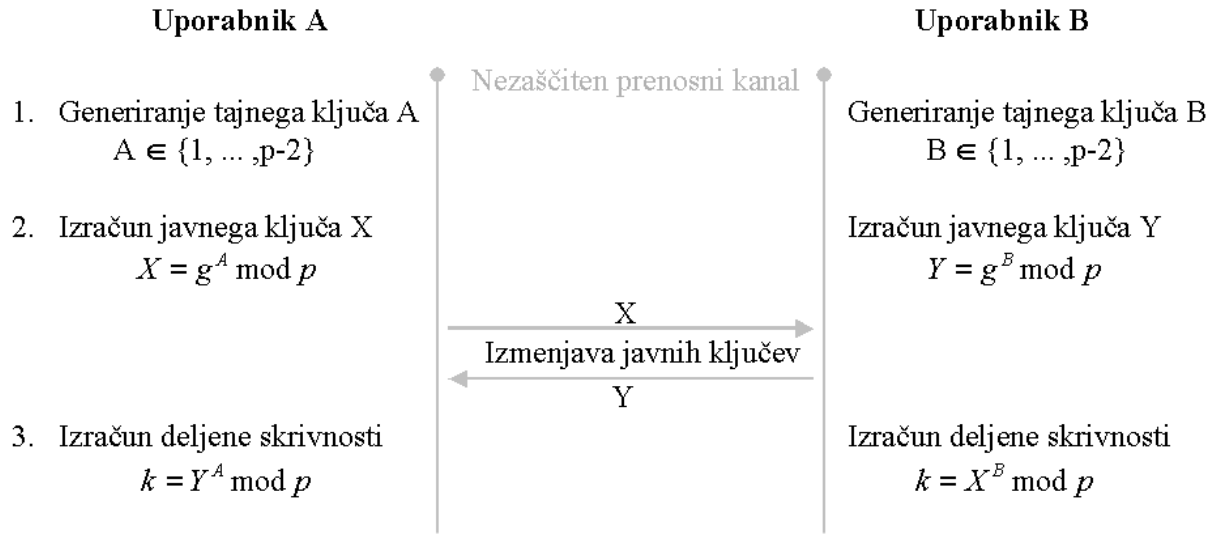
2.3 VARNOSTNI MEHANIZMI

2.3.1 Enkripcija

DNSSEC temelji na uporabi kriptografije z javnim ključem. Ljudje so svoja sporočila šifrirali skozi vso zgodovino. Zanimivo pa je, da so vse do sedemdesetih let prejšnjega stoletja za šifriranje uporabljali t. i. simetrično kriptografijo. Pri njej sta ključa za šifriranje in dešifriranje enaka, lahko rečemo, da se uporablja zgolj en ključ. Problem takega šifriranja je, da si morata osebi pred začetkom medsebojne komunikacije na varen način izmenjati ključ. Najboljši način je, da se dobita v živo, seveda v okolju, kjer prisluškovanje ni mogoče.
Ko so se začeli pojavljati računalniki in prva omrežja, je naenkrat med seboj hotelo komunicirati veliko računalnikov z vseh koncev sveta. Na dolgih mednarodnih povezavah je potreba po kriptiranju še toliko večja. Že omenjeni problem varne izmenjave ključev ostaja aktualen. Administratorja vsakega para računalnikov, ki bi hotela med seboj komunicirati, bi si morala najprej varno izmenjati ključ. Drug problem pa je v številu potencialnih parov, ki bi želeli komunicirati med seboj. Kot primer samo vzemimo e-poštne strežnike. Vsak e-poštni strežnik bi moral imeti vnaprej vnesene ključe za simetrično enkripcijo vseh ostalih e-poštnih strežnikov, na katere bo pošiljal oz. iz katerih bo prejemal elektronsko pošto.

Čas je torej bil, da se razvije neka povsem druga vrsta šifriranja, asimetrično kriptiranje, kjer bi se ključa za šifriranje in dešifriranje razlikovala.

Ena takih idej se je v sedemdesetih letih prejšnjega stoletja porodila kriptografoma z imeni Whitfield Diffie in Martin Edward Hellman. Razvila sta svoj algoritem za varno izmenjavo ključev, ki se mu pravi izmenjava ključev Diffie-Hellman (angl. Diffie-Hellman Key Exchange). Entiteta A in entiteta B, ki želita med seboj varno komunicirati, se najprej morata zmeniti glede dveh števil – g in p. Število g je običajno majhno naravno število, število p pa je veliko praštevilo. Entiteta A generira skrito poljubno število Xa, entiteta B pa skrito poljubno število Xb. Entiteta A na podlagi svojega skritega števila izračuna število Ya po formuli $g^{Xa}\ mod\ p = Ya$. Podobno izračuna število Yb entiteta B: $g^{Xb}\ mod\ p = Yb$. Ya in Yb sta javni vrednosti in si jih entiteti izmenjata. V zadnjem koraku entiteta A izračuna skriti ključ Z po formuli $Z = Yb^{Xa}\ mod\ p$, entiteta B pa dobi isti ključ Z po formuli $Z = Ya^{Xb}\ mod\ p$. Entiteti A in B sta prišli do istega skritega števila, ne da bi si to število prej izmenjali. Napadalec, ki prisluškuje, pa števila Z ne more izračunati, saj mu manjkata skriti števili Xa in Xb. Za rekonstrukcijo števila Z potrebuje vsaj enega izmed njih [12].



Sl. 8: Izmenjava ključev Diffie-Hellman [11]

Slabost te izmenjave je, da morata pri njej obe entiteti aktivno sodelovati. Priročneje bi bilo, da bi vsaka entiteta imela par ključev: javni ključ, ki bi bil vedno na voljo vsakomur, ki bi želel varno komunicirati z njo, in privatni ključ, ki bi bil tajnost posamezne entitete. S privatnim ključem bi entiteta lahko vedno dešifrirala sporočilo, ki je bilo šifrirano z njenim javnim ključem. Govorimo o enosmerni funkciji e(x), ki šifrira sporočilo, vendar ima le-ta »loputo« [15]. S skrito informacijo (v našem primeru privatnim ključem) se da to enosmerno funkcijo preprosto invertirati. Diffie in Hellman sta si tak način kriptografije zamislila, opisala in poimenovala kriptografija z javnim ključem (angl. Public-Key Criptography) [13], nista pa ga uspela realizirati. Predpostavljala sta, da v matematiki obstajajo enosmerne funkcije, ki bi tega bile zmožne, nista pa jih uspela najti.
Je pa to leta 1977 uspelo trem kriptografom, ki so algoritem poimenovali po kratici svojih priimkov kot RSA (Rivest, Shamir, Adleman).[3] Algoritem temelji na modularni aritmetiki z velikimi praštevili. Iz javnega ključa entitete A je nemogoče dobiti njen privatni ključ in obratno [12]. Algoritem RSA temelji na predpostavki, da računalniki zelo slabo faktorirajo velika praštevila. Ko se bo odkrila boljša metoda faktoriranja velikih praštevil, bo algoritem RSA padel. Kljub temu algoritem RSA trenutno velja za zelo dobrega in je posledično tudi zelo priljubljen.

Splošna ideja asimetrične kriptografije oz. kriptografije z javnim ključem je torej, da ima vsaka entiteta svoj javni in privatni ključ. Javni ključ je na voljo vsem ostalim entitetam, privatni ključ pa ima dotična entiteta varno shranjen pri sebi. Entiteta A, ki želi komunicirati z entiteto B, bo to storila na način, da bo sporočilo šifrirala z javnim ključem entitete B, entiteta B bo sporočilo dešifrirala s svojim privatnim ključem.

Poleg RSA, ki je najbolj priljubljen in najširše uporabljen algoritem za asimetrično šifriranje, obstaja še nekaj asimetričnih algoritmov. Kot eno boljših je ocenjeno šifriranje z uporabo

[3] Leta 1997 je britanska vlada razkrila, da so bili postopki asimetrične kriptografije dejansko izumljeni s strani uslužbencev GCHQ (angl. Government Communications Headquarters) leta 1973. Raziskovalci so neodvisno izumili sistem za izmenjavo ključev Diffie-Hellman in posebno obliko RSA. [35]

eliptičnih krivulj. Na splošno velja, da je asimetrična kriptografija precej bolj procesorsko zahtevna kakor simetrična kriptografija. Že same dolžine ključev so mnogo daljše. Pri RSA in Diffie-Hellman so tipične dolžine ključev 1024 bitov, lahko pa gredo vse do 4096 bitov. Pri algoritmih za simetrično šifriranje, kot so npr. 3DES, AES, RC4, IDEA, je tipična dolžina ključev od 128 do 192 bitov. Ker je asimetrična kriptografija procesorsko tako zahtevna, ponavadi uporabljamo t. i. hibridne kriptosisteme, kjer asimetrični algoritmi poskrbijo, da se entiteti ena drugi avtenticirata in da si varno izmenjata ključ za simetrično kriptiranje, ki ga izvajata v nadaljevanju komunikacije. [11]

2.3.2 Digitalni podpis

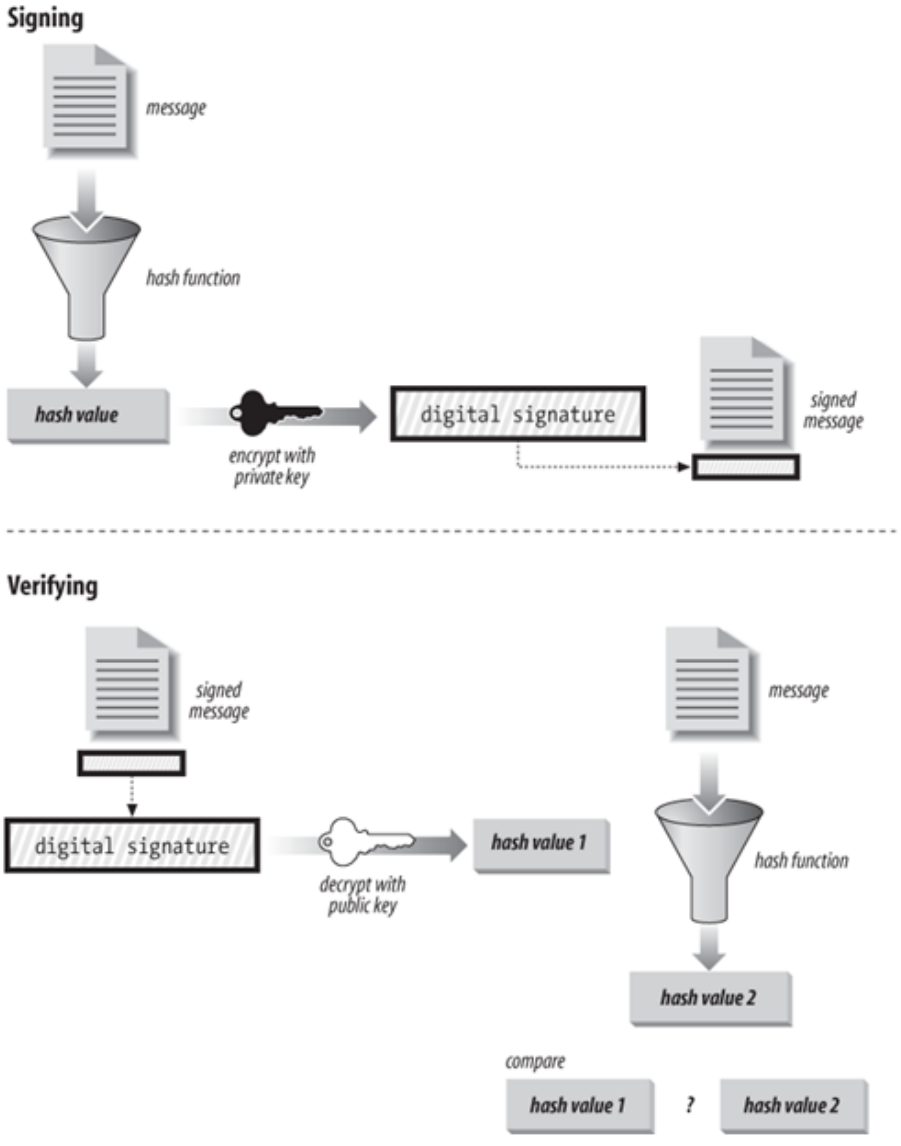
Kriptografija z javnim ključem hkrati generira javni in privatni ključ, ki sta si enakovredna. Mimogrede bi lahko ob procesu kreiranja ključev namesto privatnega ključa kot privatni ključ vzeli javnega. Skratka, kar se zaklene z enim ključem, se vedno da odkleniti z drugim. Če entiteta A šifrira sporočilo s svojim javnim ključem, ga lahko vse ostale entitete, ki imajo javni ključ entitete A, dešifrirajo. Samo entiteta A je tista, ki lahko šifrira sporočilo s svojim privatnim ključem. Sedaj vsakdo, ki sporočilo bere, ve, da je to sporočilo tako, kakršno je napisala entiteta A (spremenjena ni niti ena vejica).

Šifriranju s privatnim ključem se pravi digitalno podpisovanje. V zvezi z njim se uporablja nerodno preveden slovenski izraz ne-tajenje (angl. non-repudiation), kar pomeni, da je digitalni podpis s pravnega vidika enakovreden lastnoročnemu podpisu. Že prej je bilo omenjeno, da je šifriranje z asimetričnimi algoritmi procesorsko zelo zahtevno dejanje. Zato tudi ne šifriramo celotnega sporočila, ampak zgolj njegov izvleček (angl. hash). Le-ta je rezultat enosmerne matematične zgoščevalne funkcije, ki je izvedena nad sporočilom. Z izvlečki si pomagamo pri zagotavljanju integritete sporočila (z njo zagotovimo, da sporočilo ni bilo spremenjeno). Rezultat je fiksne dolžine, to je običajno od 128 do 192 bitov. Algoritmi, ki se uporabljajo, so MD5, ki se zaradi matematičnih slabosti opušča, SHA-1 in SHA-2, ki je neuradni standard za SHA256, SHA384, SHA512 (cifre za imenom pomenijo dolžino izpisa v bitih).

Zgoščevalne funkcije so enosmerne funkcije, za katere je značilno, da nimajo inverzne funkcije. Le-to pomeni, da iz rezultata zgoščevalne funkcije ne moremo rekonstruirati originalnega sporočila, kar pa je nemogoče tudi zaradi dejstva, da se je med izvajanjem zgoščevalne funkcije del informacije izgubil. Rezultat je namreč bistveno krajši od sporočila v originalu. Zgoščevalne funkcije so injektivne, kar pomeni, da lahko zgoščevalne funkcije, izvedene nad različnimi sporočili, vrnejo enak rezultat. Injektivnost je seveda nezaželena, vendar se ji ne moremo izogniti, saj zgoščevalne funkcije vrnejo rezultat omejene dolžine, vhodna sporočila pa so lahko poljubno dolga. Tako imamo npr. pri zgoščevalni funkciji, ki vrne 128-bitni rezultat, na voljo poljubno število različnih vhodnih sporočil, vseh možnih rezultatov pa je končno število (v našem primeru 2^{128}). Temu, da funkcija pri dvema različnima sporočiloma vrne enak rezultat, pravimo kolizija. Vse zgoščevalne funkcije imajo možnih neskončno število kolizij, razlikujejo pa se v tem, kako lahko je kolizijo najti (torej poiskati sporočila, za katere funkcije vrnejo enake rezultate). Poleg tega naj bi zgoščevalne funkcije pri malenkost različnih sporočilih vrnile rezultate, ki so si povsem različni.

V primeru uporabe MD5 je kolizije preveč preprosto najti, zato ni več priporočena za uporabo. V naslednjih letih lahko pričakujemo novo zgoščevalno funkcijo, ki bo odpravila slabosti obstoječih glede preprostega iskanja kolizij in možnosti kolizij pri minimalnih spremembah originalnega sporočila [12][25].

Če povzamem je postopek digitalnega podpisovanja sledeč. Entiteta A najprej z uporabo zgoščevalne funkcije naredi izvleček sporočila, nato ga s svojim privatnim ključem šifrira in doda originalnemu sporočilu. Entiteta B bo ob preverjanju digitalnega podpisa z javnim ključem entitete B dešifrirala izvleček podpisanega sporočila, ki ga je priložila entiteta A. Nato bo še sama izračunala izvleček originalnega sporočila. Če se bosta izvlečka ujemala, je sporočilo identično in avtentično.



Sl. 9: Postopek digitalnega podpisovanja in preverjanja digitalnega podpisa [2]

2.4 ZAPISI DNSSEC

DNSSEC nam prinaša samo 4 oz. 6 novih zapisov. To so DNSKEY, RRSIG, DS, NSEC in NSEC3 ter NSEC3PARAM, ki sta uvedena naknadno v RFC 5155 [5][7].

2.4.1 DNSKEY

Zapis DNSKEY vsebuje javni ključ, s katerim lahko preverimo podpise RRSIG vseh zapisov DNS v določeni coni. Obstajata dve vrsti ključev – KSK (angl. Key Signing Key) in ZSK (angl. Zone Signing Key).

2.4.1.1 KSK IN ZSK

KSK je ključ, s katerim naj bi podpisali zgolj ZSK, s ključem ZSK pa naj bi podpisali vse ostale zapise v coni. Dovolj je uporabljati samo en ključ. [4]

Zakaj bi uporabljali dva ključa? Več kriptiranih podatkov imamo zajetih, lažje nam je razbiti ključ. Pri velikih conah, npr. coni kake multinacionalke, se mora cona podpisati večkrat dnevno. Zapisi se nenehno dodajajo, brišejo in spreminjajo. Zato moramo na določen čas zamenjati naš ključ (ZSK). Priporočila za DNSSEC narekujejo, da bi se ZSK velikosti 1024 bitov moral menjati vsaj na vsake 4 mesece. DNSSEC v svoji hierarhiji deluje na način, da mora izvleček ključa posredovati domeni en nivo višje. Torej bi morali vsake 4 mesece uveljaviti spremembe v domeni en nivo višje, ki ponavadi ni v našem upravljanju. Zato se je uvedel KSK, ki lahko podpiše ZSK. Če uporabljamo KSK, domeni na višjem nivoju sporočimo zgolj tega, ZSK podpišemo sami s KSK. KSK je običajno daljši, priporočena dolžina ob predpostavki, da ga bomo menjali enkrat na leto, je 1024 bitov za manj pomembne domene, 1300 bitov za srednje pomembne in 2048 bitov za zelo pomembne domene. Glede na to, da z njim na vsake 4 mesece podpišemo zgolj en zapis (ZSK), kriptoanalitiki nimajo ravno veliko podatkov za kriptoanalizo. [2][18]

2.4.1.2 Sintaksa DNSKEY

Zapis DNSKEY zgleda takole:

```
kozic.net.              604800  IN      DNSKEY  257 3 5
AwEAAdv8To75CqCTc7CrLdf2/Kr6LsGHpxtRuC62EjYxsGM54JW3s60R
QufxhS7UEJEMLUqzDVI4vZ9BTCuKt+Xbad2An6Rt8l2+iSjK82c+c0eX
9j8Oo41d8IYOQoiOsefTahFYKm4ugLfPPy6l0vLaL9q+iFcbXLopqX0U
rRz5an1HriJeLNYFtILj48qYpGXHbIZ+hVkyHaD3AcJQj7fPHWj2akhv
AFA57e6ffEz91Lf/eDV0z29Dbf8+Lwxgzw/nsmgqJcYekKm4C6ZviAoU
LGsWYAxyFZxZld7ppbyCN5Uby8zL+MvvbmPl7O3EBEK98xfHcxeTGBso
0/KCxfRP5cKD6kB4S5DCBWTRx7omiHJOthhoXOgHesVBE7AwqyZ0AnNn
rOyagkOMDua9v8lDfovSnxkNaeSGln6XdnMka2ddDQ8zbBePnwZo5laK
r9tGyds9d4FlPcG2ze58JBN1NqoCTqvOcTOsN174K9wz+oJRv67nP8X7
crvxBfMs95LOUtoWNZxCvpTHvb4pByJkz6zhbRHHV2pvCGXwJA9zXIDJ
dkfeI1PDl4wRY2j0NtTzacpeZlZSX/wxELRMatRZM/7bogT9RUMxmTyd
```

```
mScsJQ18h2MqfnfNk4i6iiK7wi85FXGRo4lsRVUlOHH6qkey/uS+b+ZT
q9T43rYH8FspMf63
```

Prvo polje za tipom (angl. Flags; 257) predstavlja zastavico. Veliko je 2 bajta, večina bitov mora biti postavljenih na 0. Pomembna sta bit 7 in bit 15. Če je bit 7 nastavljen na 1, pomeni, da je v tem zapisu vsebovan javni ključ od cone. V tem primeru se ta ključ uporabi za preverjanje ostalih zapisov v coni. Če je nastavljen na 0, pomeni, da gre za nek drug javni ključ DNS. Bit 15 predstavlja SEP (angl. Secure Entry Point). Če je 0, pomeni, da gre za ZSK, če je 1, pomeni, da gre za KSK. V našem primeru imamo v tem polju številko 257, kar pomeni, da sta oba bita nastavljena na 1. Torej gre za javni ključ, s katerim je podpisana cona, in sicer KSK. Če bi imeli ZSK, bi bila vrednost 256, če pa bi imeli nek drug ključ DNS, pa bi bila vrednost 0. RFC 5011 naknadno dodaja uporabo bita 8, ki predstavlja bit za preklic ključa DNSKEY. Če je nastavljen na 1, pomeni, da je bil ključ KSK kompromitiran, ostalim strežnikom pa sporoča, naj več ne validirajo s tem ključem in naj ta ključ umaknejo, če mu zaupajo. Z bitom 8 na 1 ima ključ KSK vrednost 385. Naslednje polje (angl. Protocol) je polje protokola. Polje je obdržano iz prejšnjih različic DNSSEC, trenutno se v njem vedno zahteva vrednost 3. Tretje polje (angl. Algorithm) predstavlja algoritem za podpisovanje, ki ga uporablja cona. Po RFC 4034 so tukaj mogoče naslednje vrednosti:

- 0: rezervirano
- 1: RSA/MD5 (ni priporočeno)
- 2: Diffie-Hellman
- 3: DSA/SHA-1
- 4: Eliptične krivulje (ECC)
- **5: RSA/SHA-1 (obvezno)**
- 253–254: za privatno uporabo
- 255: rezervirano

Naše zapise DNSSEC moramo po RFC 4034 vedno podpisati vsaj z algoritmom RSA/SHA-1, lahko pa uporabo algoritma RSA/SHA-1 zamenjamo z uporabo enega izmed naknadno uvedenih (poznejši RFC več ne predpisujejo nujno uporabo algoritma 5). Diffie-Hellman se sicer ne more uporabljati za podpisovanje con, se ga pa lahko uporablja za ostala opravila v zvezi z DNSSEC. V primeru imamo v tretjem polju vrednost 5, kar pomeni, da gre za ključ, ki ga bomo uporabljali za preverjanje podpisov RSA/SHA-1. V zadnjem polju (angl. Public Key) je javni ključ, zakodiran v formatu BASE-64. [2][5]

Kot vidimo imamo za digitalno podpisovanje na voljo zgolj dve zgoščevalni funkciji, MD5 in SHA-1. RFC 5702 omogoča tudi uporabo drugih zgoščevalnih algoritmov SHA-2, točneje RSA/SHA-256 in RSA/SHA-512. SHA-384 ni predviden za uporabo v DNSSEC.

2.4.2 RRSIG

RRSIG je digitalni podpis posameznega RRSET (angl. Resource Record Set). RRSET predstavlja vse zapise DNS, ki imajo isto ime in so istega tipa (če bi mia.kozic.net v spodnjem primeru imela več zapisov A, bi se za vse skupaj naredil zgolj en podpis RRSIG). Z njim so podpisani tudi vsi ostali zapisi, ki jih prinaša DNSSEC. Izgleda takole:

```
mia.kozic.net.          604800  IN      A       93.103.130.109
```

```
mia.kozic.net.          604800  IN      RRSIG   A 5 3 604800
20101008125456 20100908125456 17580 kozic.net.
jY9p+Y/R0DagMEMP5y1HOGqg0uMPbMsRowrqDg6BD1a07BFB1GU6wpx3
V+33YTqf9bMTp4EmaWg7A98N1SVHBUgNVpBc65ZZy24xrKx953jDm0hQ
WKWWQs6hiO/w9LRQTIVCB5BAyw5mCUNA8p9Qoqzgltgdrdupg4K1DPh5 iGY=
```

Prva vrstica kaže zapis DNS, druga vrstica pa zapis RRSIG, ki predstavlja digitalni podpis originalnega zapisa DNS. Prvo polje za RRSIG (angl. Type Covered) je tip vsebovanega sporočila (v našem primeru A). Če bi mia.kozic.net pokrivala več zapisov (npr. še zapis MX), bi bil za vsak tak podpis oblikovan svoj RRSIG. Naslednje polje (angl. Algorithm) predstavlja algoritem, s katerim je narejen digitalni podpis. Vemo, da 5 predstavlja algoritem RSA/SHA-1. Posamezen zapis DNS lahko podpišemo z več različnimi algoritmi, v tem primeru imamo več zapisov RRSIG za posamezen zapis DNS. Naslednje (tretje) polje (angl. Labels) definira, koliko oznak imamo v imenu zapisa DNS. Za ločilo med oznakami je mišljena pika, torej ima mia.kozic.net 3 oznake. Polje oznak je potrebno pri uporabi zapisov z zvezdico (npr. *.kozic.net). Četrto polje (angl. Original TTL; 604800) kaže originalen TTL zapisa DNS. Originalni zapis TTL moramo shranjevati, ker strežnik DNS vrednost TTL vsako sekundo zmanjša za 1. Za preverjanje digitalnega podpisa pa seveda moramo podpisati vrednost, ki se vsaj določen čas ne spremeni. Naslednji dve polji sta potek (angl. Signature Expiration) in začetek podpisa (angl. Signature Inception). Začetek podpisa je čas, ko smo cono podpisali, potek pa je čas, ko podpis poteče. Po poteku podpisa se zapis RRSIG več ne more uporabljati za validiranje podatkov o zapisu DNS, ki ga pokriva. Informaciji o poteku in začetku podpisa sta predstavljeni v obliki absolutnega časa UTC (angl. Coordinated Universal Time) LLLLMMDDUUmmSS, kjer LLLL predstavlja leto, MM mesec, DD dan, UU uro, mm minuto in SS sekundo. Naslednje polje (angl. Key Tag; 17580) predstavlja oznako ključa, to je identifikator ključa, s katerim je podpisan zapis. Identifikator ključa ni enoličen. V posamezni coni lahko imamo več ključev, s katerimi podpisujemo podatke, zato je treba povedati, kaj smo podpisali s katerim. Predzadnje polje (angl. Signer's Name; kozic.net) predstavlja lastnika podpisa. Gre za ime domene, v kateri najdemo ključ, s katerim lahko preverimo podpisane zapise. Zadnje polje (angl. Signature) predstavlja sam digitalni podpis. Podpis pokriva vse, kar je desno od zapisa RRSIG brez zadnjega polja (samega sebe). Zakodiran je v formatu BASE-64. [2][5]

2.4.3 DS

Zapis DS (angl. Delegation Signer) se uporablja za preverjanje zapisa DNSKEY. Nahaja se v domeni en nivo višje. Tako lahko zgradimo hierarhično verigo zaupanja (angl. chain of trust). Zapis DS izgleda takole:

```
kozic.net.          86400 IN DS 6046 5 1
890C6C8DCDEE056D6667C112D2029B3B57A298F3
```

Prvo polje za tipom je oznaka ključa (angl. Key Tag; 6046), ki nam pove, kateri izmed zapisov DS pripada kateremu ključu DNSKEY-domene, ki jo validiramo. Ključ je podan v obliki zapisa DNSKEY. Naslednje polje je algoritem (angl. Algorithm; 5), ki ga uporablja DNSKEY za podpisovanje zapisov v coni. Sledi tip zapisa DS (angl. Digest Type; 1). DNSSEC-bis za zapis DS predvideva samo uporabo zgoščevalne funkcije SHA-1, RFC 4509

to razširja na SHA-256 (v tem primeru je Digest Type 2). Sledi izračunan izvleček (angl. Digest), ki je zapisan v heksadecimalni obliki. [2][5][16]

2.4.4 NSEC

Podpisati moramo tudi odgovor na poizvedbo, ki sprašuje po neobstoječem imenu. Takšno povpraševanje po DNS vrne odgovor NXDOMAIN (angl. Non-Existent Domain). Poleg tega, da vrnemo tako kodo, je treba v DNSSEC odgovor tudi ustrezno podpisati. Če negativnih odgovorov ne bi podpisovali, bi lahko napadalec na vse poizvedbe DNSSEC vračal negativne odgovore, uporabnik pa ne bi imel nobene informacije o njihovi pristnosti.

V DNSSEC cono najprej podpišemo, potem pa naš strežnik DNS odgovarja z vnaprej pripravljenimi in podpisanimi odgovori. Ko enkrat cono podpišemo, privatnega ključa več ne potrebujemo. Lahko ga umaknemo s strežnika, na ta način se zavarujemo, da bi kdo, ki nam vdre v strežnik, lahko modificiral našo cono. Negativnega odgovora za posamezno povpraševano ime tako ne moremo sproti oblikovati in podpisati. Za poizvedbo po neobstoječem imenu DNS bi lahko imeli vnaprej pripravljen podpisan odgovor. Z njim bi lahko vrinjeni napadalec odgovoril na vsako legitimno vprašanje. Če legitimnega uporabnika napadalec ne bi mogel preusmeriti na napačno stran, bi pa vsaj lahko naredil neke vrste napad DOS. Naša domena bi bila nedosegljiva, hkrati pa bi uporabnik mislil, da povpraševani zapisi naše domene zares ne obstajajo. Potrebujemo torej tudi zanesljiv dokaz, da povpraševano ime ne obstaja (angl. Authenticated Denial of Existence).

Pri podpisovanju negativnih odgovorov si pomagamo z zapisom NSEC (angl. Next SECure), ki izgleda takole:

```
dusan.kozic.net.        1800    IN      NSEC    hrosci.kozic.net. A RRSIG
NSEC

dusan.kozic.net.        1800    IN      RRSIG   NSEC 5 3 1800
20101008125456 20100908125456 17580 kozic.net.
lgAbmZJcrBJTdLg89FiHjpmOWkEkqs0YDURWpc0xyOd5k7iXn+8W8VB4
GHN16pKZZrWTpJhh3bQ/ZMa3nmc2xh8fRtVZ+E3jnZ3W4nCkfX9pf9RI
zWEjqDTWPvk6V+ebQfzEb/t0htaWnbtlCs0dtIYajbjWP7sQ6WvhACIY 9g0=
```

Kot je razvidno, je zapis NSEC dodatno tudi podpisan, kar dokazuje njegovo avtentičnost. V našem primeru smo vprašali po zapisu ftp.kozic.net, ki ne obstaja, zato je strežnik odgovoril z zapisom NSEC. Sam zapis NSEC nam pove, da domenskemu imenu dusan.kozic.net sledi ime hrosci.kozic.net. Med dusan.kozic.net in hrosci.kozic.net, kjer se leksikografsko tudi nahaja domensko ime ftp.kozic.net, pa ni ničesar. Odgovor je ustrezno podpisan, zato smo lahko prepričani, da med tema dvema zapisoma dejansko ni ničesar. A, RRSIG in NSEC pa nam povejo, da ima domensko ime dusan.kozic.net te tri zapise DNS. Podpisani negativni odgovori v coni predstavljajo polovico njene velikosti. Ogromno resursov se vrti okrog ene na videz preproste stvari. [2][5]

2.4.5 NSEC3

Iskanje načina, kako podpisati negativne odgovore DNS, je bil za razvijalce DNSSEC kar velik izziv. Mojo tezo potrjuje tudi dejstvo, da jim prvič ni uspelo. Problem je v tem, da se lahko napadalec s poizvedovanjem po neobstoječih zapisih dokoplje do vseh podatkov posamezne cone.[4] Ko vpraša po ftp.kozic.net, prejme namig, da obstajata tako zapis dusan.kozic.net kakor tudi zapis hrosci.kozic.net. Naslednje, kar lahko napadalec naredi, je, da vpraša po neobstoječem imenu, ki po abecedi sledi imenu hrosci.kozic.net. Napadalec npr. vpraša po imenu hrosci2.kozic.net. Malo verjetno je, da bi tako ime obstajalo, sicer pa ni težko poiskati takega imena, ki ne obstaja:

```
hrosci.kozic.net.       1800    IN      NSEC    j4.kozic.net. CNAME
RRSIG NSEC
```

Napadalec je pravkar dobil namig, da za imenom hrosci.kozic.net obstaja ime j4.kozic.net. Sedaj vidimo način, po katerem se lahko napadalec iterativno dokoplje do vseh zapisov neke cone. Večina administratorjev strežnikov DNS ne želi, da bi lahko uporabnik dobil sliko celotne domene, ki jo administrirajo. Zato imamo navadno omejen prenos vseh zapisov cone zgolj na sekundarne strežnike DNS. DNSSEC nam je prinesel veliko varnosti, hkrati pa nam je je nekaj odvzel. [2][8]

Iznajti je bilo treba boljši način; tako je leta 2008, tri leta za standardom DNSSEC-bis, nastal RFC 5155, ki odpravlja ta problem. Zapis NSEC3 izgleda takole:

```
2la218nnivrkseqoqg3vqgiqva0pgdut.org. 874 IN NSEC3 1 1 1 D399EAAB
2LJ88IKK5A1886SR8IM8RC2OE3IM2QF8 A RRSIG
```

```
h9p7u7tr2u91d0v0ljs9l1gidnp90u3h.org. 874 IN RRSIG NSEC3 7 2 86400
20110206165056 20110123155056 1743 org.
fq2zqKdocDkULM7qEHBcAA4jW4CswILQaX9lPwtEUWAab50O8MHwXpKd
jEYwFJnWdCp73x9yjE+SUixadoDcZ/l/Os21ZCUiqlM9Hp/GZXV8wnrP
EfrbnTku8E2afpuEXpM1bihD97xyNXEs+6kpJUUH1BvUT5SI2P13IGrm BrI=
```

Cono z zapisi NSEC3 naredimo podobno kakor cono z zapisi NSEC. Razlika je, da v primeru, ko delamo cono z zapisi NSEC3, najprej izračunamo izvlečke vseh zapisov v coni, potem pa izračunane izvlečke razvrstimo po abecedi.

Poizvedovali smo po kozic.org. Dobili smo odgovor, da to domensko ime ne obstaja. Nad imenom kozic je avtoritativni strežnik za domeno .org izvedel zgoščevalno funkcijo. Potem je poiskal izvleček, ki je leksikografsko pred izvlečkom od imena kozic (2la218nnivrkseqoqg3vqgiqva0pgdut) in ga vrnil kot odgovor. Prvo polje za zapisom NSEC3 predstavlja zgoščevalno funkcijo (angl. Hash Algorithm; 1 pomeni SHA-1, ki je edina dovoljena za uporabo po RFC 5155). Naslednje polje (angl. Flags) predstavlja zastavice. Trenutno je uporabljena samo ena zastavica, ki določa, ali domena vsebuje delegacije opt-out. Ob uporabi opt-out ne ustvarjamo zapisov NSEC3 za delegacije, ki nimajo zapisa DS (torej za

[4] Obstaja celo lažji način sprehajanja po coni. Napadalec lahko spraševanje direktno po zapisih NSEC. V Windows Server tako sprehajanje po coni na ta način sicer ni možno.

delegacije poddomen, ki niso podpisane). Tretje polje po vrsti (angl. Iterations) predstavlja število iteracij, ki jih je naredila naša zgoščevalna funkcija. Več iteracij naredimo, bolj so zakriti zapisi DNS znotraj domene. Četrto polje (angl. Salt) predstavlja parameter sol, to je dodana beseda, ki se uporabi pri kalkulaciji izvlečka. Z uporabo parametra sol in večjega števila iteracij, ki jih naredi zgoščevalna funkcija, otežimo iskanje originalnih zapisov s pomočjo rainbow tabel. Naslednje polje (angl. Next Hashed Owner Name) predstavlja izvleček naslednjega zapisa v domeni. V našem primeru se izvleček imena kozic nahaja med zapisoma 2la218nnivrkseqoqg3vqgiqva0pgdut in 2lj88ikk5a1886sr8im8rc2oe3im2qf8**.** Zadnje polje (angl. Type Bit Maps) pa predstavlja podatek, katere zapise vsebuje izvleček imena, ki nam je podan kot odgovor NSEC3 (A, RRSIG). [7]

Za verifikacijo zapisa NSEC3 imamo dovolj podatkov. Imamo zgoščevalni algoritem, parameter sol in število iteracij. Iz tega lahko izračunamo izvleček imena, po katerem smo povpraševali. Potem preverimo, ali se naše ime nahaja med ponujenima odgovoroma. Na koncu je odgovor NSEC3, podpisan z zapisom RRSIG; tako lahko preverimo njegovo pristnost. Smo pa dobili dodaten zmogljivostni problem – tako validator kakor avtoritativni strežnik DNSSEC imata sedaj še eno dodatno stvar za preračunavanje.

Uporaba zapisa NSEC3 dodaja še en zapis DNSSEC – zapis NSEC3PARAM. NSEC3PARAM vsebuje parametre NSEC3. To so zgoščevalna funkcija, zastavice, število iteracij in parameter sol. Potrebujejo jih avtoritativni strežniki za računanje izvlečkov imen. Strežniki, ki validirajo in razrešujejo naslove DNS, zapisa NSEC3PARAM ne uporabljajo. [7]

2.5 POIZVEDOVANJE DNSSEC

DNSSEC uporablja preprosto razširitev protokola DNS, imenovano EDNS0 (angl. Extension mechanisms for DNS version 0). EDNS0 omogoča uporabo več zastavic, tipov oznak in povratnih kod, njegova najpomembnejši element pa je ta, da omogoča daljša sporočila DNS. Originalno sporočilo DNS je lahko veliko največ 512 B, kar se je ob uporabi IPv6 in zlasti DNSSEC izkazalo za premalo. Največja velikost sporočila DNS, ki jo omogoča EDNS0, je 4096 B, ki se tudi običajno uporablja. DNSSEC dodatno uporablja EDNS-zastavico DO (angl. DNSSEC OK flag). Z njo strežnik pove, da podpira DNSSEC in da v odgovoru želi prejeti tudi zapise DNSSEC.

DNSSEC uporablja še dve zastavici v standardni glavi DNS, in sicer zastavici AD in CD. Zastavici predstavljata dva bita izmed treh do sedaj neuporabljenih bitov v glavi DNS. Zastavica AD (angl. Authenticated Data) pomeni, da je rekurzivni strežnik uspešno validiral odgovor DNS. Ko je odgovor DNSSEC pristen, rekurzivni strežnik to zastavico postavi na 1. Strežnik pošlje zastavico CD drugemu strežniku, ko ne želi, da drugi strežnik zanj dela validacijo DNSSEC. To ponavadi naredi strežnik, ki je zmožen sam validirati odgovor DNSSEC. [1][2]

2.5.1 Validiranje DNSSEC

Ena izmed nalog rekurzivnega strežnika DNSSEC je, da preveri, ali je odgovor DNSSEC pristen. Če nekdo ponareja odgovore DNS ali pa zastruplja medpomnilnik DNS, rekurzivni

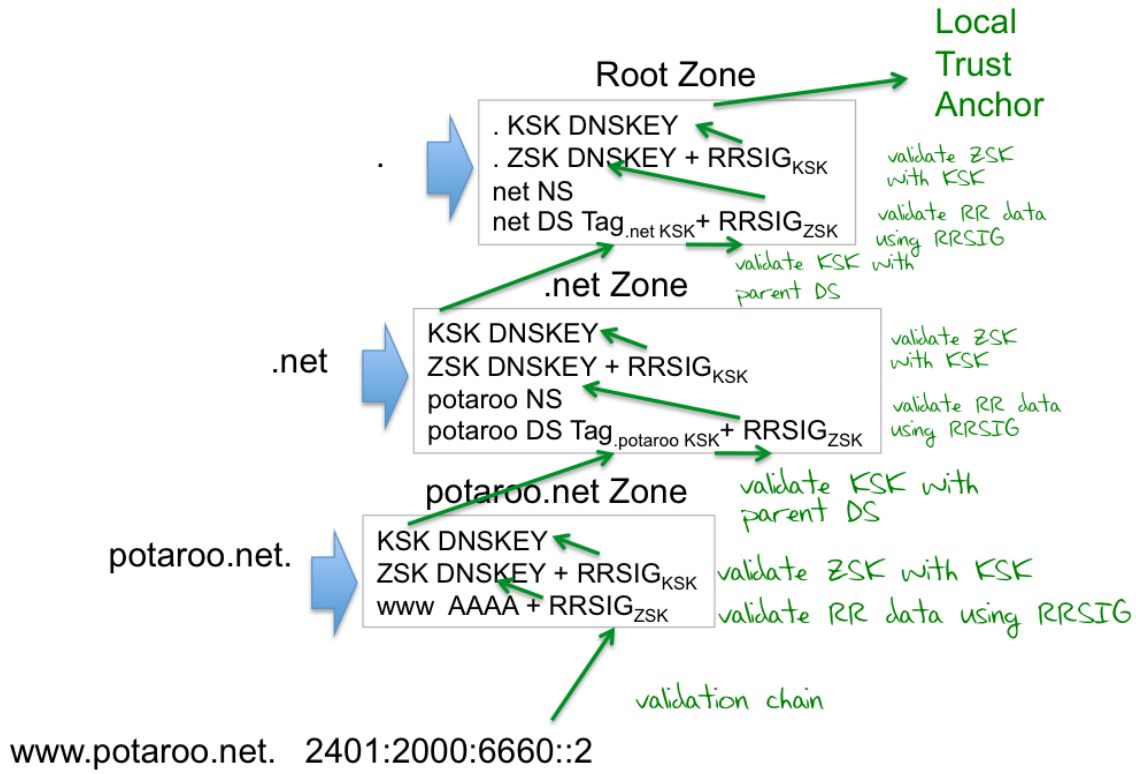
strežnik DNS ne bo uspešno validiral takega odgovora (torej ga ne bo označil za veljavnega). V takem primeru v odgovoru uporabniku zastavica AD ne bo nastavljena, kar bo za uporabnika pomenilo, da validacija domene ni bila uspešna in da domeni ne sme zaupati.

Poizvedovanje po DNSSEC si poglejmo na primeru. Razrešiti hočemo naslov www.kozic.net. Med izvajanjem poizvedb DNSSEC bomo v odgovorih poleg zapisov tipa A dobivali tudi podpise teh odgovorov – zapise tipa RRSIG. Preverjanje zapisov RRSIG pa bo potekalo ravno obratno, kakor sam proces poizvedovanja po DNS. Če po DNS poizvedujemo tako, da najprej začnemo s korensko domeno, bomo zapise RRSIG validirali ravno v obratni smeri. Ob poizvedovanju DNSSEC skupaj s standardnimi zapisi DNS dobivamo tudi zapise DNSSEC. Validiranje zapisa DNS za www.kozic.net poteka po sledečem postopku:

1) RRSIG zapisa www.kozic.net validiramo z ustreznim zapisom DNSKEY za domeno kozic.net (če uporabljamo KSK in ZSK nadalje validiramo RRSIG zapisa DNSKEY ZSK s KSK; primer kaže uporabo samo enega ključa DNSKEY).
2) Avtoritativni strežnik za domeno .net nam je vrnil zapis DS, ki se nanaša na ključ DNSKEY v avtoritativnem strežniku za domeno kozic.net. Z njim validiramo ključ avtoritativnega strežnika za domeno kozic.net.
3) Preveriti moramo veljavnost samega zapisa DS domene kozic.net. Tudi za zapis DS obstaja njegov podpis RRSIG. Podpis RRSIG zapisa DS validiramo s ključem DNSKEY domene .net.
4) Avtoritativni strežnik za vrhnjo domeno nam je vrnil zapis DS, ki se nanaša na javni ključ DNSKEY domene .net. Z njim validiramo ključ avtoritativnega strežnika za domeno .net.
5) Zapis DS za domeno .net je podpisan s ključem DNSKEY korenske domene. Validirajmo RRSIG zapisa DS s ključem DNSKEY vrhnje domene.
6) Preveriti moramo javni ključ korenske domene. Vprašajmo strežnik, ki je nad korenskim strežnikom, po zapisu DS javnega ključa korenskega strežnika.

Zadnje ne bo mogoče, saj nad korensko domeno ni ničesar. Izvleček javnega ključa korenskega strežnika ali pa njegov javni ključ moramo vedeti sami oz. mora vedeti naš rekurzivni strežnik DNS, ki ga bomo uporabljali za validiranje.

Vse, kar potrebujemo za uspešno validiranje, je torej en javni ključ oz. izvleček enega javnega ključa, tj. javni ključ korenske domene. Dobimo ga lahko v paketu z našim strežnikom DNS ali pa ga snameno s spletne strani IANA. Previdnost pri prenosu javnega ključa za korensko domeno pa ni odveč. Prepričani namreč moramo biti, da je to res javni ključ korenske domene. Pri tem si lahko pomagamo s SSL. [1][2]



Sl. 10: Primer postopka validiranja DNSSEC domene www.potaroo.net [53]

2.5.2 Delno validiranje

Za uspešno validiranje domene morajo DNSSEC podpirati vsi strežniki DNS v hierarhiji. Če je pa v hierarhiji kak strežnik, ki ne podpira DNSSEC, bo preverjanje neuspešno. Domena kozic.net je sicer lahko podpisana, vendar se v domeni en nivo višje (domena .net) ne nahaja zapis DS za domeno kozic.net. Validiranje DNSSEC v tem primeru ne bo uspešno. Ker se DNSSEC še uveljavlja, mora biti poskrbljeno tudi za ta scenarij. Podpisanim domenam, katerih domene na višjem nivoju niso podpisane, pravimo otoki varnosti (angl. Islands of Security).

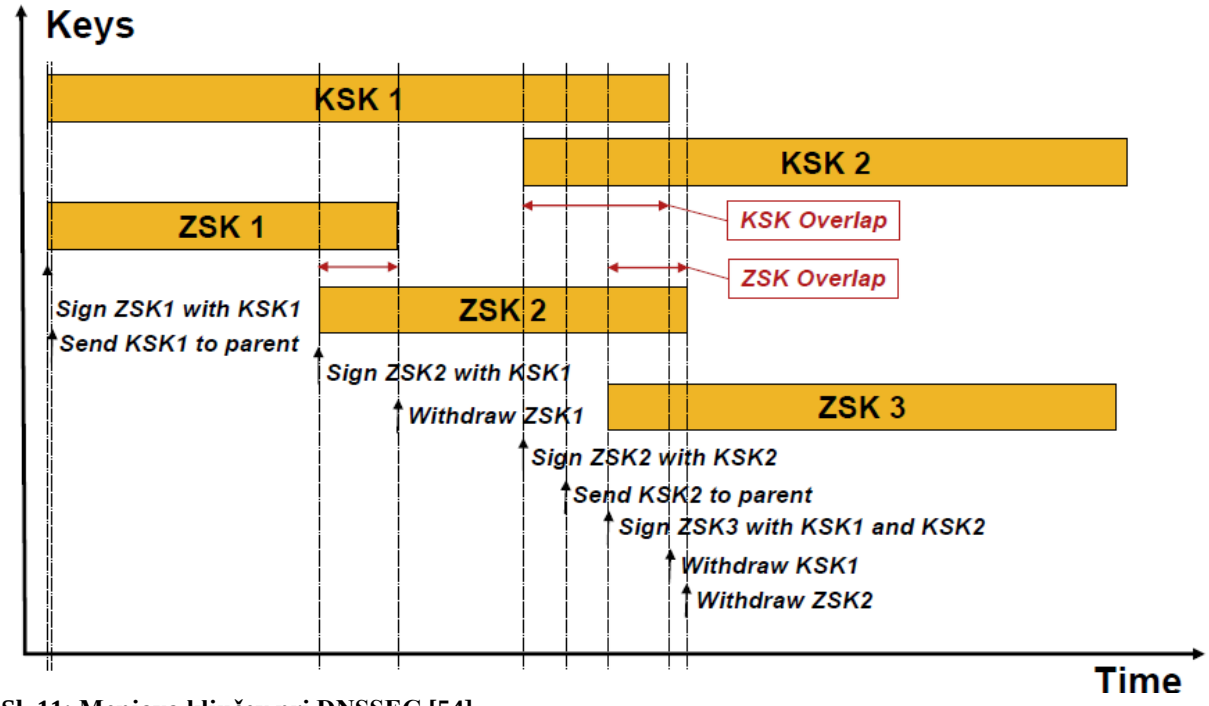
Lahko pa ima strežnik DNS našega prijatelja ali pa poslovnega partnerja vnešeno t. i. sidro zaupanja (angl. Trust Anchor) za domeno kozic.net. Strežnik, ki ima vnešeno sidro zaupanja za posamezno domeno, ima vnesen njen ključ KSK in ga posledično lahko validira. Na ta način prijateljev strežnik DNS ne bo šel iskat zapisa DS za domeno kozic.net na domeno en nivo višje, ampak bo imel sam dovolj informacij, da bo lahko izvedel validiranje te domene. Če bi domena kozic.net imela poddomene, pa bi se od najnižjega nivoja v hierarhiji do puščice zaupanja uporabljal postopek, opisan v prejšnjem podpoglavju (2.5.1). [1]

2.6 VZDRŽEVANJE CONE

Že prej je bilo omenjeno, da ima vsak zapis RRSIG svoj rok trajanja. Ko le-ta poteče, strežnik, ki validira domeno, ne bo vrnil zastavice AD. Doseženo bo podobno stanje kakor v

primeru, ko bi napadalec ponarejal naše DNSSEC-odgovore, kar pomeni, da uporabnik domeni ne bo zaupal. Iz tega sledi, da bi naša domena za legitimne uporabnike postala nedosegljiva. Zato moramo poskrbeti za pravilno vzdrževanje cone. Administrator si ne sme dovoliti, da bi preprosto pozabil ponovno podpisati cone, in sicer jo lahko ponovno podpišemo z istim ključem, v tem primeru je to opravilo precej preprosto, le pozabiti ne smemo nanj.
Poleg ponovnega podpisovanja cone je ob določenih časovnih intervalih treba tudi zamenjati ključe, s katerimi je cona podpisana. Ključ DNSKEY sicer nima roka trajanja, ki ga imajo samo podpisi RRSIG. Kot sem že omenil, je varnost ključa pogojena tudi s količino zajetih podatkov. V manjših statičnih conah napadalec ne bo mogel zajeti velikega števila različnih podatkov, zato tako pogosta menjava ključev ni potrebna. Na voljo so orodja, ki proces vzdrževanja con in ključev popolnoma avtomatizirajo (dejansko je menjava ključa KSK polavtomatska, saj je treba sporočiti zapis DS coni en nivo višje, vse ostalo pa je lahko avtomatizirano).

Zamenjati certifikat na spletnem strežniku je lažji proces, kakor zamenjati ključ DNSKEY. Kakor vemo, DNS predpomni podatke za določen čas v medpomnilniku. Zapis DNSKEY se torej lahko hrani tudi po zamenjavi v medpomnilniku katerega od strežnikov DNS, kateri drugi zapis in njegov podpis pa ne, če noben uporabnik še ni vprašal po tistem zapisu. V tem primeru bi bila validacija neuspešna, saj bi bil zapis podpisan z novim ključem, rekurzivni strežnik pa bi ga validiral s starim. Zato potrebujemo učinkovito strategijo menjave ključev. Uporabljata se dve različni. [1][2][18]



Sl. 11: Menjava ključev pri DNSSEC [54]

2.6.1 Metoda »pre-publish«

Novi ključ DNSKEY objavimo, preden domeno z njim podpišemo. Tako imamo hkrati objavljena dva ključa DNSKEY. Stari se bo še naprej uporabljal za validacijo zapisov, novi pa bo objavljen z namenom, da si strežniki DNS tudi njega sproti zapomnijo. Če je TTL za našo cono 86400 (= 24 ur), moramo novi ključ DNSKEY objaviti vsaj dva dni pred potekom naše cone. Novi ključ DNSKEY objavimo skupaj s starim ključem, s katerim imamo še vedno podpisane zapise. Lahko smo prepričani, da bodo vsi strežniki DNS v roku enega dneva pridobili tudi novi ključ. Nato vse zapise podpišemo samo z novim ključem, vendar starega obdržimo v coni, in sicer vsaj 24 ur. Po izteku 24 ur bodo stari zapisi RRSIG izginili iz vseh strežnikov DNS. Potem lahko stari ključ v miru umaknemo iz cone, priporoča pa se tudi, da stari privatni ključ popolnoma uničimo. Vidimo, da je čas menjave ključev odvisen od TTL.

2.6.2 Metoda dvojnega podpisovanja

Pri metodi dvojnega podpisovanja (angl. Double-Signing Method) cono podpišemo z obema ključema hkrati. Pri uporabi te metode lahko stari ključ skupaj s starimi zapisi odstranimo po času TTL. Menjava ključev je torej dvakrat hitrejša kakor po metodi »pre-publish«.

Kljub temu je priporočeno, da pri menjavi ključa ZSK uporabimo metodo »pre-publish«. Dvojno podpisana cona je dvakrat večja od cone, ki je podpisana z metodo »pre-publish«, saj je vsak zapis DNS podpisan z obema ključema ZSK, zato za domene TLD metoda dvojnega podpisovanja sploh ne pride v poštev. Ob poizvedovanju DNS se prenese tudi dvakratna količina prometa. Je pa metoda dvojnega podpisovanja potrebna pri zamenjavi algoritma za podpisovanje cone. Ob menjavi KSK pa se uporablja metoda dvojnega podpisovanja. Stari ključ KSK mora ostati v coni, dokler ne poteče stari zapis DS na višjem nivoju (nikar ne pozabimo objaviti zapisa DS novega ključa v domeni en nivo višje). Če imamo svoj ključ ročno vnesen v določenih strežnikih DNS (angl. Trusted Anchors), ga moramo zamenjati tudi tam. [1]

2.7 PREDNOSTI IN SLABOSTI DNSSEC

2.7.1 Zmogljivost

Zapisi DNSSEC so od zapisov DNS večji 5- do 7-krat [8], velikost podpisane cone pa je večja 3- do 4-krat [2]. Na avtoritativnih strežnikih domene to pomeni, da strežnik DNS potrebuje bistveno več spomina, da naloži in servira cono. Odgovori DNS bodo tako večji, DNSSEC bo bistveno povečal količino prenesenega prometa DNS po internetu. Datagrami ne bodo več tako majhni, kakor so bili doslej.

Najbolj obremenjen bo rekurzivni strežnik DNS, ki bo validiral zapise DNSSEC. Pri njem bo zaradi računanja kriptografije zahtevana bistveno večja procesorska moč kakor doslej. Internetni ponudniki bodo potrebovali strežnike z večjo procesorsko močjo, tudi sami strežniki DNS pa bodo morali biti napisani tako, da bodo znali to procesorsko moč izkoriščati čim bolje. Na validiranje bo zelo vplivala tudi dolžina verige zaupanja. [2]

Če bi se prehod na DNSSEC zgodil čez noč, bi se celoten sistem DNS zaradi bistveno večje porabe procesorske moči in pomnilnika na strežnikih DNS ter zaradi bistveno več omrežnega prometa med njimi sesul. Vendar je prehod postopen in počasen, vprašanje če bodo vse domene sploh kdaj uporabljale DNSSEC. Tako kakega črnega scenarija ni za pričakovati. Je pa treba upoštevati, da DNSSEC prinaša veliko dodatno obremenitev doslej dokaj nezahtevnega in hitrega sistema DNS.

2.7.2 Časovna sinhronizacija

Začetek in veljavnost podpisa DNSSEC sta podana v obliki absolutnega časa. Sedaj je pomembno, da strežniki DNS sinhronizirajo čas preko strežnikov NTP (angl. Network Time Protocol). Strežniki morajo imeti pravilno nastavljen absolutni čas, v nasprotnem primeru se lahko zgodi, da bodo kot veljavnega validirali zapis, ki je potekel, ali kot neveljavnega validirali zapis, ki je še veljaven. Pred uvedbo DNSSEC so strežniki DNS lahko imeli napačno nastavljeno uro. [8][9]

DNSSEC je ranljiv za t. i. freshness oz. replay napad [9]. Dokler je nek odgovor DNS veljaven (vključno s podpisom RRSIG), ga je mogoče ponavljati, kar v praksi izgleda tako, da si napadalec nekam shrani stari odgovor DNS. Zapis shranjenega odgovora kaže na nek naslov IP in se med časom spremeni (zlasti se to pogosto dogaja pri dinamičnem DNS). Napadalec še vedno hrani stari zapis DNS, ki ga po potrebi lahko vrine rekurzivnemu strežniku DNS. Ker je zapis RRSIG še vedno veljaven, bo vrinjeni odgovor uspešno validiran. Zaščita pred tem napadom je nastavljen kratek čas trajanja zapisov RRSIG, kar zahteva pogosta nova podpisovanja cone, ki so lahko opravljena s starimi ključi. Velike cone, kjer se zapisi pogosto spreminjajo, so replay napadu precej izpostavljene.

Zamislimo si situacijo, da je ključ cone kompromitiran. Od RFC 5011 imamo na voljo možnost preklica ključa DNSKEY. Kljub temu pa sistem DNS uporablja predpomnenje starih odgovorov. Na ta način preklic na vseh strežnikih ni takoj viden. Dodatno je zaradi TTL in delegacij mogoč prehod v začasno nekonsistentno stanje [9]. Standard DNSSEC [4] predpisuje nastavitev časov TTL znotraj posamezne cone, zaplete pa se lahko v domeni, ki je en ali več nivojev višje. Napadalec lahko po kraji ključa po želji spreminja, prireja in podpisuje vse zapise DNS znotraj cone. V tem primeru administrator čimprej objavi preklican ključ, kreira novega in nov zapis DS sporoči strežniku en nivo višje. Vendar ima v našem primeru zapis DS en nivo višje večji TTL kakor zapis DNSKEY v naši coni. Dokler bo zapis DNSKEY shranjen v medpomnilniku katerega od rekurzivnih strežnikov DNS, bo tam shranjen tudi zapis DS. Če bi imel zapis DS manjši ali vsaj enak TTL od zapisa DNSKEY cone, bi kompromitiran zapis DNSKEY najpozneje po poteku svojega TTL postal neveljaven [9]. Idealno bi seveda bilo, da bi bil TTL zapisa DS najmanjši, kot je le mogoče, vendar bi to bistveno bolj obremenjevalo sistem in strežnike DNS. Treba je najti kompromis med veljavnostjo zapisov DNSSEC v medpomnilniku in med nevarnostjo kompromitacije naših ključev.

2.7.3 Sprehajanje po conah in NSEC3

Omenil sem že pomanjkljivost zapisa NSEC, ki je namenjen avtentikaciji negativnih odgovorov DNS. Tale zapis je napadalcu omogočil sprehod po coni. Mnogi administratorji pa ne želijo, da bi si lahko napadalec ali vedoželjnež ogledal vse zapise DNS njihove cone. Avtorji DNSSEC so uvedli zapis NSEC3 [7], ki naj bi s pomočjo zgoščevalnih funkcij cono naredil ponovno nevidno.

NSEC3 je s t. i. opt-out poskrbel za velike cone, zlasti za cone strežnikov vrhnjih (TLD) domen, ki že tako ali tako zasedajo ogromno prostora in pomnilnika. Za cone, ki so delegirane in ne uporabljajo DNSSEC, ne delamo zapisov NSEC3. Na ta način se poveča velikost cone za manj kot 5 % [26]. Kljub temu naj bi še vedno bilo poskrbljeno za varno delegiranje poddomen. Vzemimo za primer delegirano domeno unsigned.kozic.net, ki ne uporablja DNSSEC, medtem ko ga kozic.net uporablja. Izvleček za unsigned.kozic.net se npr. nahaja med izvlečkoma za www.kozic.net in mail.kozic.net. Avtoritativni strežnik DNS na ta način za kozic.net v odgovoru na poizvedbo po imenu unsigned.kozic.net ponudi zapis NSEC3, ki zgleda takole: `izvleček od www.kozic.net NSEC3… izvleček od mail.kozic.net`. Če rekurzivni strežnik DNS izračuna izvleček od unsigned.kozic.net, se bo le-ta nahajal med tema dvema izvlečkoma. Slabost tega je, da lahko napadalec v tem primeru kreira svoj odgovor, izračuna njegov izvleček in poišče ustrezen zapis NSEC3, s katerimi dokaže njegovo istovetnost. Na ta način lahko napadalec preusmeri delegirano nepodpisano domeno na svoj lažni strežnik DNS. Avtorja članka A Security Evaluation of DNSSEC with NSEC3 [9] iz tega razloga uporabo opt-out zelo problematizirata in odsvetujeta. Sam v tem primeru ne vidim tolikšnega varnostnega tveganja, saj se lahko v tem scenariju napad izvrši na naslednjem nivoju, to je s potvarjanjem odgovorov strežnika delegirane nepodpisane domene, ki se jih ne da preveriti. Uporabo opt-out torej bolj vidim kot prednost kakor slabost.

Ker obstajajo tabele ogromno preračunanih izvlečkov različnih zgoščevalnih funkcij (angl. Rainbow Tables), se da iz izvlečka neke zgoščevalne funkcije v teh tabelah poiskati originalen tekst. Zapisi DNS so dokaj kratki, zato je velika verjetnost, da se bodo originalna imena naših zapisov DNS nahajala v teh tabelah. Le-to nam pove, da tudi NSEC3 ni ravno najučinkovitejša rešitev pred skrivanjem vnosov v neki coni. Da bi vseeno imena zapisov malo bolj prikrili, so zapisu NSEC3 dodali parameter sol, ki spremeni rezultat zgoščevalne funkcije in tako upočasni iskanje po tabelah izračunanih izvlečkov zgoščevalnih funkcij. Vendar uporaba parametra sol ne naredi ravno veliko, saj mora biti zaradi potreb verifikacije priložen odgovoru NSEC3 [9]. Vse, kar naredi, je, da napadalcu ukrade nekaj časa, saj mora poleg vsega preračunavati še parameter sol. Dokaj uporabna tehnika v zvezi z NSEC3 pa je število iteracij, ki jih naredi zgoščevalna funkcija. Zgoščevalno funkcijo tako lahko izvedemo nad samim njenim rezultatom in tako naprej. Več iteracij naredimo, bolj se zaščitimo, vendar potrebujemo višje procesorske zmogljivosti. Dodaten zmogljivostni problem, ki ga prinaša NSEC3, je, da mora sedaj računati tudi avtoritativni strežnik DNS – računati mora izvlečke. Kljub temu že omenjeni članek A Security Evaluation of DNSSEC with NSEC3 [9] navaja, da je po nekaj dneh preračunavanja zapisov NSEC3 mogoče priti do originalnih zapisov DNS. Le-to je dokaz, da zgoščevalnih funkcij ne moremo uporabljati za zagotavljanje zaupnosti podatkov. Če želimo zaupnost, moramo uporabiti kriptografijo.

2.7.4 Ostale pomanjkljivosti

Preverjanje veljavnosti podpisov DNSSEC se po RFC 4033 izvaja na rekurzivnem strežniku DNS. To je strežnik, ki ga uporabnikov računalnik uporablja za razreševanje naslovov DNS. Če so podpisi DNSSEC veljavni, bo ta strežnik v odgovoru vrnil zastavico AD. Če te zastavice ne bo, preverjanje DNSSEC ni uspelo in uporabnikov računalnik ne bo zaupal odgovoru DNS. Celotno zaupanje uporabnikovega računalnika glede odgovora DNSSEC temelji na eni zastavici. Komunikacija med uporabnikovim računalnikom in rekurzivnim strežnikom DNS, ki se običajno nahaja pri ponudniku ISP, v osnovi ni kriptirana, torej lahko napadalec, ki se vrine med uporabnika in rekurzivni strežnik, brez večjih težav manipulira s posredovanimi odgovori DNSSEC (dejansko mu je treba manipulirati zgolj z eno zastavico – enim bitom). Temu pravimo omejitev zadnjega skoka (angl. »Last-Hop« Limitation) [9]. Zaščita je, da se povezava med rekurzivnim strežnikom DNS in uporabnikovim računalnikom kriptira. V tem primeru moramo uporabljati mehanizem DNS TSIG ali pa zgraditi tunel IPsec (angl. Internet Protocol Security) [4], kar poleg uporabe DNSSEC prinaša še uporabo drugih varnostnih mehanizmov, vse skupaj pa dodatno poslabša odzivnost, zlasti pa je uporaba takih varnostnih mehanizmov za uporabnika zapletena. Mehanizem TSIG je v prvi vrsti namenjen varnemu prenosu cone med primarnim in sekundarnimi strežniki DNS, lahko se pa uporabi tudi za zavarovanje komunikacije med odjemalcem in strežnikom DNS. Prav tako uporaba mehanizma TSIG ali pa tunela IPsec na velikih rekurzivnih strežnikih ISPjev ne pride v poštev, saj bi med drugim zmogljivost preveč trpela. Tako v sedanjih postavitvah rekurzivnih strežnikov DNS uporabe teh rešitev ni za pričakovati.

Paziti moramo tudi, kje hranimo zasebni ključ. Priporočeno je, da privatnega ključa ne hranimo na samem strežniku DNS. Na ta način preprečimo, da se v primeru vdora v naš strežnik DNS napadalec ne more dokopati do našega privatnega ključa [8]. Če uporabljamo dinamični DNS, pa kaj takega ni mogoče, saj moramo nove vnose podpisovati sproti. Torej dinamični DNS poslabša varnost DNSSEC.

Problematična je tudi replikacija domene med primarnim in sekundarnimi strežniki DNS. [8] DNSSEC ne skrbi za samo varnost v tem primeru, uporabiti moramo druge varnostne mehanizme, kot je npr. TSIG. Povsem mogoče je, da postane domena na različnih avtoritativnih strežnikih DNS inkosistentna. Za tak scenarij je mogočih več vzrokov; domena se ne prenese pravilno ali vrinjeni napadalec med prenosom spremeni podatke ali pa sekundarni strežniki preprosto »pozabijo« prenesti spremenjeno domeno na primarnem strežniku.

Sam rekurzivni strežnik, ki validira DNSSEC, mora biti tudi pravilno napisan. DNSSEC za komunikacijo še vedno uporablja nekriptirana sporočila DNS, ki jih lahko vrinjeni napadalec preprosto modificira. Rekurzivni strežnik ne sme nikoli zaupati v odgovoru nastavljeni zastavici AD [9] in mora biti tisti, ki taka sporočila validira. Z modificiranjem odgovorov DNSSEC v procesu razreševanja DNS lahko napadalec izvede napad na domeno na način, da postane ta domena nedosegljiva. Odgovori ne bodo validirani, zato domena legitimnemu uporabniku ne bo na voljo.

2.7.5 Dobre in slabe lastnosti DNSSEC

V Tab. 2 se nahaja povzetek dobrih in slabih lastnosti DNSSEC.

Tab. 2: Dobre in slabe lastnosti DNSSEC

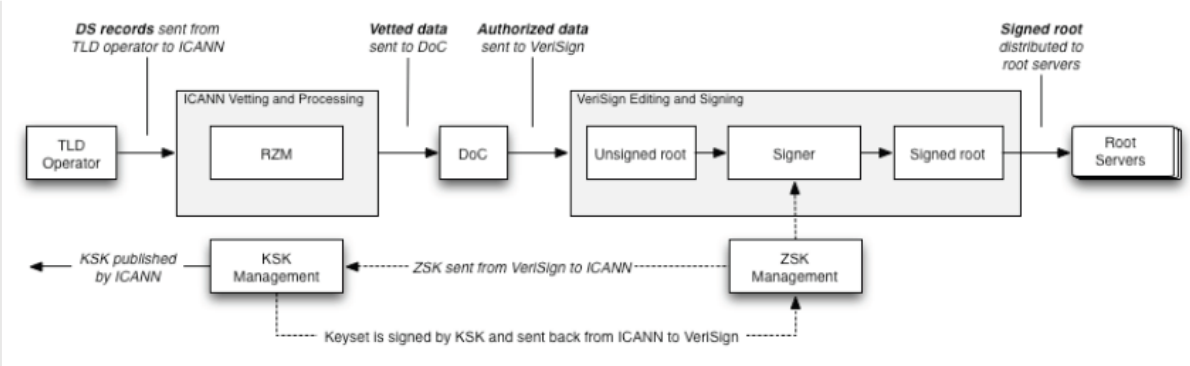
Dobre	Slabe
uporaba varnostnih mehanizmov PKI	ni zaupnosti pri poizvedovanju
zaščita pred DNS spoofing	večja poraba resursov
zaščita pred zastrupljanjem medpomnilnika DNS	sprehajanje po coni (NSEC)
NSEC3 opt-out poskrbi, da cone strežnikov TLD ne bodo preveč zrasle	bistveno večje cone
	klient ne validira DNSSEC, zanaša se na rekurzivni strežnik DNS
podpisov DNS se ne da ponarejati	DNSSEC ne preprečuje napadov, z varnostnimi mehanizmi pa omogoča, da so zaznani

3 DNSSEC V PRAKSI

3.1 PRISTOJNE ORGANIZACIJE

3.1.1 Pristojne organizacije za korensko cono

ICANN [55] ima vlogo upravljavca funkcij IANA [56] (angl. IANA functions operator). Operatorji TLD želje po spremembah zapisa DS za njihovo cono sporočijo ICANN. ICANN ima tudi vlogo generiranja ključa KSK in podpisovanja ključa ZSK za vrhnjo cono. Pri tem sodeluje z internetno skupnostjo. ICANN mora za dovoljenje za vsako akcijo vprašati NTIA (angl. National Telecommunications and Information Administration [58]), ki je urad znotraj DoC (angl. Department of Commerce [59]). Podjetje Verisign [57] je vzdrževalec korenske cone (angl. root zone mantainer). Verisign generira ključ ZSK in njegov javni del pošlje ICANN, da ga podpiše s ključem KSK. Nato podpiše ostale zapise v coni in spremembe na koncu tudi objavi. Verisign je tisti, ki na koncu uveljavi spremembe na vseh ostalih korenskih strežnikih. [3]



Sl. 12: Vloge in pristojnosti [3]

3.1.2 Organizacije, ki skrbijo za uveljavljanje DNSSEC

Pri IETF [60] obstaja delovna skupina DNS Extensions (dnsext) Working Group, ki skrbi za razvoj in vzdrževanje standardov DNSSEC in z njim povezanih dokumentov RFC.

DNSSEC Deployment Initiative je pobuda, katere cilj je spodbuditi vse sektorje, da uveljavijo in implementirajo DNSSEC. Pobudi zagotavlja podporo Direktorat za znanost in tehnologijo pri Ministrstvu za domovinsko varnost ZDA. Znotraj pobude se je formirala delovna skupina za postavitev DNSSEC (angl. DNSSEC Deployment Working Group), ki združuje strokovnjake, aktivne v razvoju in postavljanju DNSSEC. Skupina je odprta za kogarkoli, večina aktivnosti in sodelovanja je preko njihovega e-poštnega seznama (angl. mailing list), ki je dostopen vsakomur. Pobuda ima tudi svojo spletno stran, kjer objavlja različne članke, predstavitve in dokumente v zvezi z DNSSEC. [27]

CENTR (angl. Council of European National Top Level Domain Registries [61]) je združenje registrov vrhnjih domen ccTLD, ki pripadajo različnim državam znotraj Evrope. Znotraj

združenja CENTR si različne evropske države izmenjujejo znanje in izkušnje pri vpeljavi in uveljavljanju DNSSEC.

Zelo uporabna spletna stran, na kateri so dostopni različni viri glede DNSSEC, je www.dnssec.net. [34]

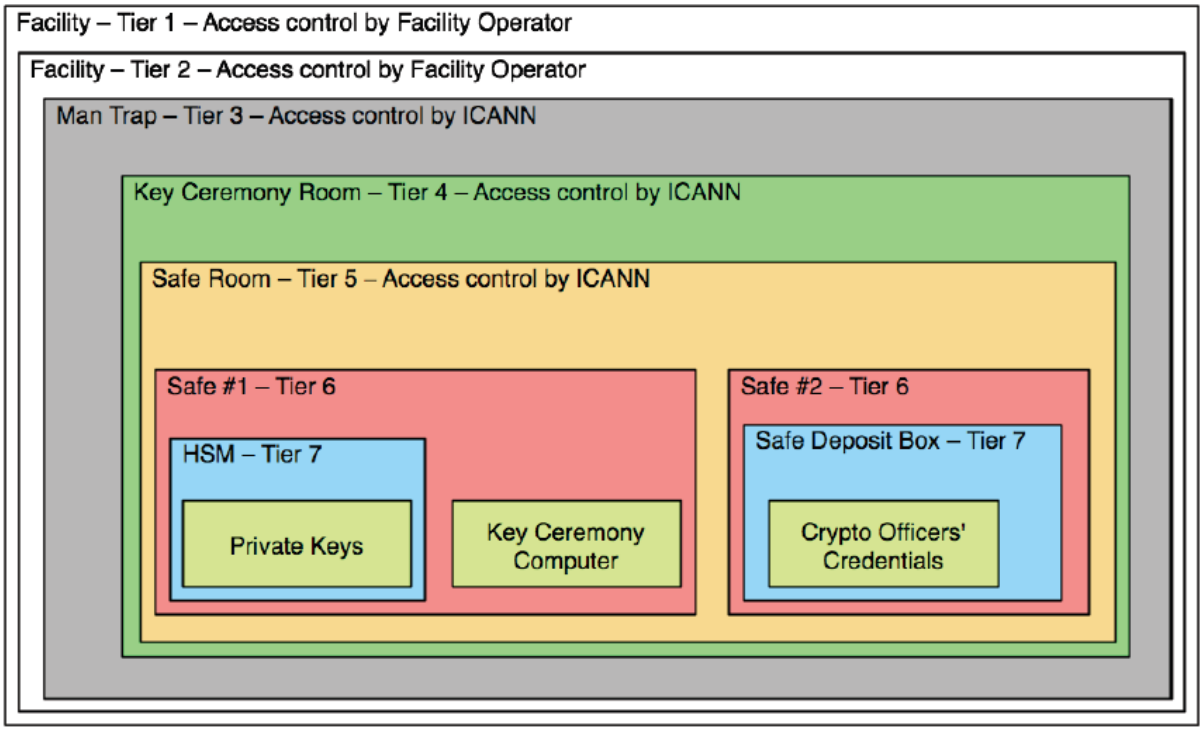
3.2 DNSSEC NA KORENSKIH STREŽNIKIH

Priprave na podpis korenske (angl. root) cone so se začele decembra 2009. Januarja 2010 je bila korenska cona v testne namene podpisana z neveljavnimi ključi. Meseca maja so vsi vrhnji strežniki servirali neveljavno podpisano korensko cono (angl. DURZ – deliberately unvalidatable root zone). Junija je bil prvi dogodek (angl. first KSK ceremony), ko je bil generiran ključ KSK. Korenska cona je podpisana od 15. julija 2010. [29]

Korenska cona ni podpisana s ključem, ki bi bil generiran mimogrede, ampak je generiranje ključev za podpis korenske cone zapleten postopek in dogodek. Kot že omenjeno, je upravljanje ključa KSK za korensko cono v pristojnosti ICANN, upravljanje ključa ZSK in podpisovanje cone pa v pristojnosti podjetja Verisign.

3.2.1 Fizična varnost

Da se v javnosti ne bi porajali dvomi, da ameriška vlada nadzoruje vso podpisano korensko cono in s tem posredno celoten sistem DNSSEC, sta v lasti ICANN zgolj fizični poslopji, kjer hranijo napravo za generiranje KSK in podpisovanje ZSK. Eno poslopje se nahaja na zahodu ZDA, točneje v Los Angelesu (nedaleč od poslopja ICANN), drugo pa na vzhodu države izven urbanega področja v Washington D.C. Za fizično varnost je odlično poskrbljeno, obe poslopji imata kar sedem varnostnih nivojev, prehod v vsakega naslednjega je zelo otežen. Vstop v območje 4 in višje ni dovoljen niti osebju, zaposlenemu v poslopju. Vstop v območje 5 lahko omogočita samo dva posebej izbrana specialista ICANN. Dostop do zadnjega nivoja pa imajo zgolj skrbno izbrane osebe s celega sveta, ki niso uslužbenci ICANN ali Verisign. Te osebe so kriptografski uslužbenci (CO, angl. Crypto Officers). Za vsako območje (vzhod in zahod) je izbranih 7 ljudi s celega sveta. Kriteriji za izbiro so, da so ljudje strokovnjaki na področju DNSSEC in kriptografje, preveri pa se tudi njihovo finančno stanje. Vsak izmed njih ima svoj sef, kjer se hrani njegova pametna kartica, in ključ, s katerim sef odklene. Za aktivacijo kriptografske naprave HSM (angl. Hardware Security Module), ki se nahaja v območju 7, morajo biti prisotni vsaj trije od sedmih kriptografskih uslužbencev. [3]



Sl. 13: Shema fizične varnosti poslopja, kjer se generira in hrani KSK [3]

Kriptografski uslužbenci so torej zadolženi za generiranje ključa KSK in podpisovanje ključa ZSK. V prostorih je seveda tudi ustrezen videonadzor, sam proces podpisovanja in generiranja ključev pa se snema in se po dogodku objavi na internetu.

Poskrbljeno je tudi za scenarij, ko bi obe poslopji izginili iz obličja zemlje. Kriptiran privatni ključ KSK, ICANN prenese izven obeh območij. Obstajajo posebne osebe, imenovane RKSH (angl. Recovery Key Shareholders), ki so določene za obnovitev ključa. Vsaka izmed njih ima pametno kartico, kjer je shranjen del ključa za dešifriranje privatnega ključa KSK, ki ga hrani ICANN. Za uspešno dešifriranje je potrebnih pet od sedmih oseb, določenih za dešifriranje ključa. Obstaja še sedem rezervnih kriptografskih uslužbencev in šest rezervnih oseb za dešifriranje privatnega ključa. [19]

Za ZSK in podpisovanje korenske cone skrbi podjetje Verisign. Ima podobno skrbno izbrane kriterije glede fizične varnosti in izbire oseb, ki generirajo ZSK in z njim podpisujejo cono. Poskrbljeno je tudi za ustrezno komunikacijo in varen prenos ključev med ICANN in Verisign. [22]

3.2.2 Vzdrževanje cone

KSK korenske cone je ključ RSA, velik 2048 bitov, z veljavnostjo petih let. ZSK korenske cone je ključ RSA z dolžino 1024 bitov in veljavnostjo 90 dni. Štirikrat na leto se torej morajo sestati tudi kriptografski uslužbenci, da ponovno podpišejo nov ZSK. Kako poteka obnavljanje in podpisovanje ZSK ter KSK, je prikazano na sliki Sl. 14.

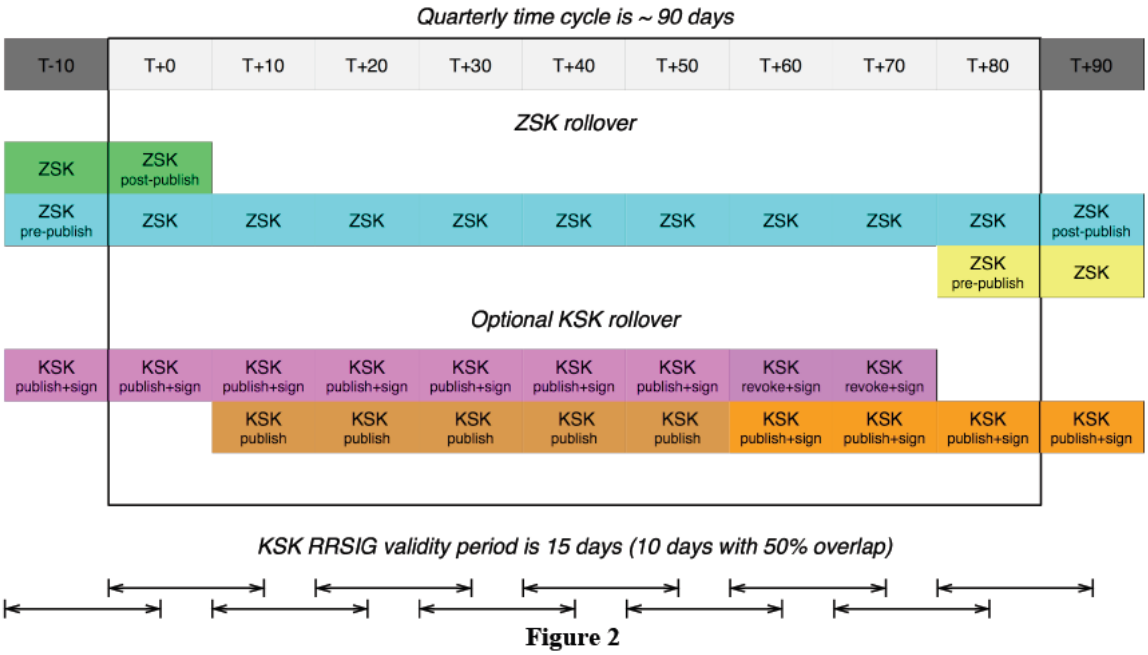


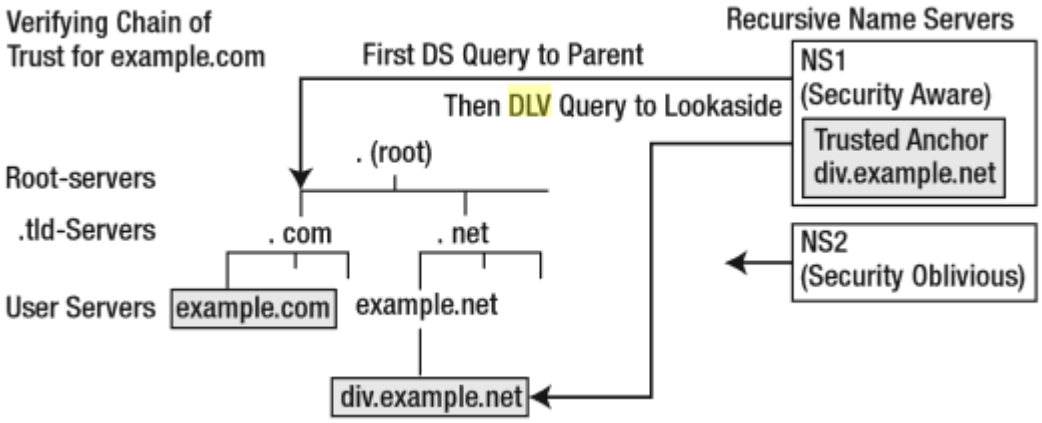
Figure 2

Sl. 14: Obnavljanje ZSK in KSK [3]

Kriptografski uslužbenci se dvakrat na leto sestanejo v poslopju na zahodu, dvakrat na leto pa v poslopju na vzhodu ZDA. Po opravljenem delu se novi ključi prenesejo na redundantno lokacijo na drugi strani ZDA. [3][19][22]

3.2.3 Projekt ISC DLV

DLV (angl. DNSSEC Lookaside Validation) predstavlja alternativo klasičnega validiranja zapisov DNSSEC. Obstajajo različni ponudniki storitev DLV, eden med njimi je ISC (angl. Internet System Consortium). Ko uporabnik hoče uporabljati DLV, poleg zapisa DS še kreira zapis DLV, ki mimogrede ni standardiziran in zavzema privaten prostor zapisov DNS. Ta zapis nato posreduje ponudniku storitev DLV. Register DLV za shranjevanje zapisov DLV uporablja svojo cono, npr. dlv.isc.org. Če je naša cona kozic.net in sporočimo svoj zapis registru DLV, se bo zapis hranil na naslovu kozic.net.dlv.isc.org. Validiranje preko registra DLV poteka tako, da validator najprej išče za zapisom DLV kozic.net.dlv.isc.org. Če ta zapis ne obstaja, se validator pomika po hierarhiji navzgor, torej bo nadalje preverjal net.dlv.isc.org. Če zapis DLV za net.dlv.isc.org obstaja, bo validator cono od tam naprej validiral na običajen način. V coni .net bo iskal zapis DS za kozic.net.
Register DLV je nastal, ko korenska cona še ni bila podpisana. Uporabnikom je ponudil možnost validiranja čim več prostora z ročnim vpisom enega samega ključa v rekurzivni strežnik DNSSEC. Obstaja več registrov DLV: ISC DLV, Verisign, IKS Jena. Danes se DLV ponuja kot alternativna možnost validiranja DNSSEC zlasti tistim, ki ne bodo zaupali podpisani korenski coni. [1][37][38]



Sl. 15: Poizvedovanje po registru DLV [1]

3.3 DNSSEC NA VRHNJIH DOMENAH

Prve podpisane domene DNSSEC so bile domene naslednjih držav: Brazilije (.br), Bolgarije (.bg), Češke (.cz), Portorika (.pr) in Švedske (.se). [34]

Med generičnimi vrhnjimi domenami je bila prva v začetku leta 2009 podpisana domena .gov. Junija 2009 je med komercialnimi generičnimi domenami bila prva podpisana domena .org. Domeni .info in .biz sta bili podpisani septembra 2010. Novembra 2010 je sledila .asia, decembra 2010 pa domena .net. Pred kratkim, v začetku aprila 2011, je bila podpisana tudi domena .com. Cona za obratni DNS za IPv6 .ip6.arpa je bila podpisana marca 2010, cona za obratni DNS za IPv4 .in-addr.arpa pa je bila podpisana marca 2010.

Slovenska domena .si je trenutno podpisana v testnem okolju, ki je kopija produkcijskega. Podpis domene .si v produkcijskem okolju se načrtuje v začetku leta 2012.

Tab. 3: DNSSEC na gTLD

DOMENA TLD	PODPIS	DOLŽINA KLJUČA KSK	NEGATIVNI ODGOVOR	PODPISANA
GOV	RSA/SHA-1	2048	NSEC3	začetek 2009
ORG	RSA/SHA-1	2048	NSEC3	junij 2009
INFO	RSA/SHA-1	2048	NSEC3	september 2010
BIZ	RSA/SHA-256	2048	NSEC	september 2010
ASIA	RSA/SHA-1	2048	NSEC3	november 2010
NET	RSA/SHA-256	2048	NSEC3	december 2010
EDU	RSA/SHA-1	2048	NSEC3	?
CAT	RSA/SHA-512	2048	NSEC3	?
MUSEUM	RSA/SHA-512	2048	NSEC3	?
COM	RSA/SHA-256	2048	NSEC3	april 2011

Aktualen seznam podpisanih vrhnjih domen DNSSEC je na [30].

Politiko podpisovanja vrhnjih domen določa posamezni upravljavec vrhnje domene.

3.4 DNSSEC PRI UPORABNIKIH

Najbolj zagreta pri implementaciji DNSSEC je bila ameriška vlada, ki je izdala ukaz, da morajo biti do decembra 2009 podpisane domene vseh zveznih agencij znotraj domene .gov. Roka se ni držalo 80 % zveznih agencij, vključujoč Ministrstvo za domovinsko varnost (angl. Department of Homeland Security), ki sicer v tem času že ima podpisano svojo domeno [31]. Trenutno je podpisanih okoli polovico domen .gov, seznam katerih se nahaja na [39].

Znotraj češke domene TLD (.cz) ta hip uporablja DNSSEC 123.015 od 795.953 domen. Češka je tako veliko število uporabnikov DNSSEC pridobila na način, da je nek registrar, ki je hkrati tudi ponudnik strežnikov DNS, podpisal vse domene, ki gostujejo na njegovih strežnikih DNS. Lastniki mnogih podpisanih domen niti ne vedo, da njihove domene uporabljajo DNSSEC. Ogromno teh domen bi naj bilo podpisanih z istim ključem, skratka gre bolj za trik, kakor pa zgledno implementacijo DNSSEC. Švedi, ki so prav tako ena prvih držav, ki so začeli z DNSSEC, poleg tega pa imajo tudi veliko znanja s tega področja, imajo podpisanih okoli 5.000 od več kot milijon domen. [51]

Dejansko se vse pomembne podpisane domene DNSSEC nahajajo znotraj domene .gov. Drugje kakega interesa namreč še ni. Bolj ali manj uporabljajo DNSSEC samo strani, ki propagirajo ali prodajajo rešitve DNSSEC. Glavni pobudnik uporabe DNSSEC bodo vlade različnih držav, ki bodo sčasoma predpisovale uporabo DNSSEC pri uporabi določenih storitev. Ena takih storitev je tudi bančništvo.

3.4.1 Registrarji s podporo DNSSEC

Prvi registrarji, ki so začeli uporabnikom omogočati, da vrhnjemu strežniku DNS sporočijo svoj zapis DS, so se pojavili junija 2010. To so bili GoDaddy, DynDNS in NamesBeyond [23]. Sedaj je registrarjev že nekaj 10, večina jih je v ZDA. Seznam registrarjev za domeno .org, ki podpirajo DNSSEC, je viden na povezavi [36].

Registrar, ki ima hkrati tudi svoje strežnike DNS, lahko uporabniku sam podpiše domeno in skrbi za njeno vzdrževanje ter vsa opravila v zvezi z DNSSEC. V tem primeru uporabnik samo reče, da želi na domeni uporabljati DNSSEC, za vse ostalo pa poskrbi registrar. Če ima uporabnik svoj strežnik DNSSEC, pa mu lahko registrar s spletnim vmesnikom omogoči, da vpiše zapis DS, ki ga bo potem sporočil naprej na strežnik TLD. Na ta način uporabnik potrebuje za spremembo zapisa DS le uporabniško ime in geslo svojega računa, potrebno za dostop do vmesnika za upravljanje z zakupljenimi domenami. Upravljanje ponavadi poteka preko varne povezave SSL.

3.5 DNSSEC NA REKURZIVNIH STREŽNIKIH

Čim bolj se približujemo uporabniku, slabše je DNSSEC realiziran. Na internetu sicer ni veliko informacij o tem, kateri ISP podpirajo validiranje DNSSEC. Bolj ali manj vsi strežniki

DNS pa bi morali posredovati zapise DNSSEC, za kar je potreben mehanizem EDNS0, ki je realiziran že zaradi uporabe IPv6.
Googlov strežnik DNS je primer tipičnega strežnika DNS, ki DNS-sporočila posreduje naprej, vendar jih ne validira. Med večjimi ponudniki interneta pa preseneča Comcast, ki se je zelo zagnano lotil projekta DNSSEC. Trenutno uporabnikom omogoča, da po želji uporabljajo njihov strežnik DNS s podporo validiranja DNSSEC, vendar kmalu načrtuje uvedbo tega kot osnovne storitve za vse uporabnike. [27][28]

3.6 DNSSEC V SLOVENIJI

Tudi v Sloveniji vsi ISP podpirajo posredovanje sporočil DNSSEC. Internetni ponudnik T-2 [64] ima na svojih rekurzivnih strežnikih polno podporo DNSSEC – torej njegovi strežniki DNS tudi validirajo DNSSEC. Arnes ima prav tako na svojih rekurzivnih strežnikih že omogočeno validiranje DNSSEC.

Glede podpisovanja domene TLD .si je Arnes (angl. Academic and Research Network of Slovenia [62]), register domene .si, v mesecu aprilu 2011 vzpostavil testno okolje, ki je glede podatkov kopija produkcijskega okolja. Na njem podpisujejo domeno TLD .si in nekaj drugih poddomen (npr. arnes.si, cert.si). V času testiranja bodo sistem za registracijo domen prilagodili tako, da bodo zainteresirani uporabniki lahko posredovali ključe DS. V testnem okolju je postavljen tudi rekurzivni strežnik, ki zna validiratati drevo .si. Le-ta je na voljo vsem zainteresiranim uporabnikom. Proti koncu letošnjega leta (2011) je načrtovana migracija testnega okolja v produkcijskega, kar pomeni, da bo Arnes v letu 2012 na domeni TLD .si že uporabljal DNSSEC. Načrtovana je uporaba ključev RSA/SHA-256 in NSEC3 z Opt-out.

Poleg tehnične podpore DNSSEC v registru Arnes pripravlja dokumentacijo in izvaja izobraževanja registrarjev in uporabnikov. [23]

Slovenski registrarji domen (npr. Domenca, Domovanje.com, SiOL …) trenutno ne omogočajo posredovanj zapisa DS domenam TLD. Za slednje moramo domeno registrirati pri enem izmed tujih ponudnikov, ki to omogočajo.

Za uporabo DNSSEC je zainteresirana slovenska vlada, Ministrstvo za visoko šolstvo, znanost in tehnologijo ter APEK (Agencija za pošto in elektronske komunikacije [64]). Tudi določene banke že kažejo interes.

T-2 ima svojo domeno t-2.net in še nekaj ostalih podpisanih (uporabljajo algoritem RSA/SHA-1 NSEC).

3.7 DNSSEC SE UVAJA POČASI

Uvedba DNSSEC za upravljanje naredi sorazmerno preprost sistem DNS bistveno bolj zapleten. Administratorji strežnikov naenkrat potrebujejo veliko več znanja, kakor so ga doslej. Možnosti, da pri uporabi DNSSEC pride do napak, so velike. Ob uporabi DNSSEC se lahko hitro zgodi, da zaradi lastne napake ali zaradi napake v programski opremi strežnika naša domena naenkrat več ne bo uspešno validirana, kar pomeni, da takrat na internetu ne

bomo dostopni. Napake so lahko precej kritične: zaradi hrošča v strežniku DNS Bind nekaj ur ni delovalo validiranje na domenah TLD .net in Francije (.fr).

V noči s 15. na 16. maj 2011 za nekaj ur ni bila dosegljiva reverzna preslikava naslovov IPv6 v imena DNS (domena ip6.arpa). ICANN to cono podpisuje z uporabo rešitve OpenDNSSEC, ki zaradi neznanega hrošča ni več ponovno podpisala cone. ICANN sicer z različnimi orodji nadzoruje pravilnosti ponovnega podpisovanja in delovanja svojih con, vendar so napako kljub vsemu prezrli.

Takim ali drugačnim hroščem v programski opremi se ne bomo mogli povsem izogniti, saj je programska oprema za DNSSEC šele v fazi razvoja. [27] Upam si trditi, da bo ob vpeljavi DNSSEC internet postal bistveno manj dosegljiv, kot je sedaj.

Za večje ponudnike storitev DNS je DNSSEC tudi dodaten strošek. Npr. registri TLD si zaradi redundance najamejo strežnike DNS drugih ponudnikov v tujini. Ponudniki takih strežnikov lahko zaračunavajo velikost cone in količino prometa, ki sta pri DNSSEC večji.

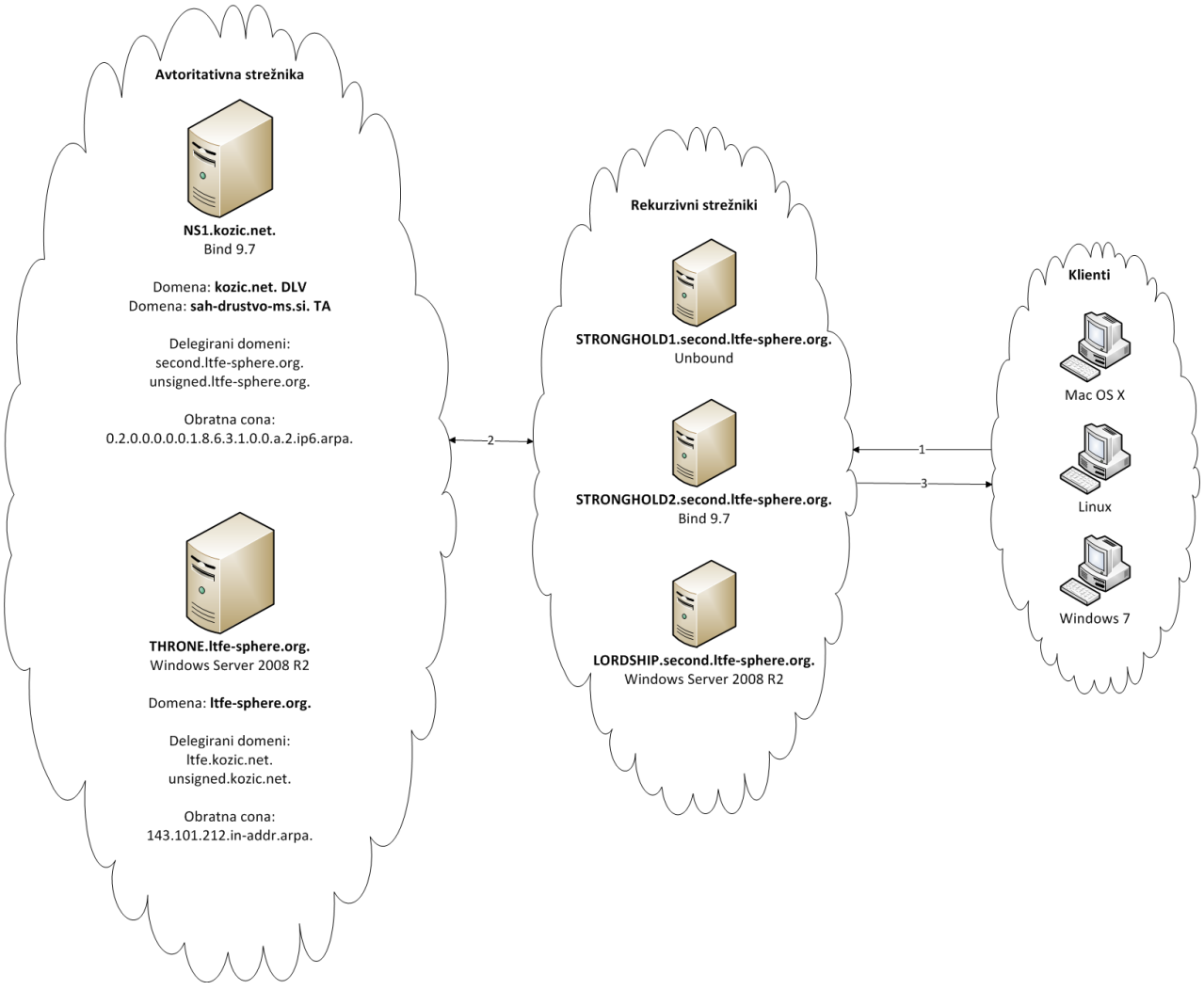
Kaj storiti v primeru, ko uporabnik DNSSEC menja registrarja ali ponudnika strežnikov DNS? Da bo njegova domena v času selitve dostopna, bo sedaj moral sodelovati tudi ponudnik, ki je stranko izgubil. Kakšen pa je interes ponudnika za stranke, ki mu odhajajo?

4 IMPLEMENTACIJA DNSSEC V TESTNEM OKOLJU

V praktičnem delu sem želel ugotoviti, kakšno je stanje glede DNSSEC na odjemalcih in strežnikih. Pri strežnikih sem preizkusil tako avtoritativne kakor tudi rekurzivne strežnike. Zanimala me je kompleksnost njihove konfiguracije in vzdrževanja ter podprtost različnih standardov RFC.

Zanimalo me je, kako lahko končni uporabniki trenutno uporabljajo DNSSEC, zato sem si še ogledal, kako je DNSSEC podprt v samih operacijskih sistemih za odjemalce in katere aplikacije trenutno podpirajo DNSSEC. Pogledal sem tudi sistemska orodja, ki poizvedujejo po hierarhiji DNS in preverjajo delovanje DNSSEC.

4.1 TESTNO OKOLJE



Sl. 16: Topologija testnega okolja

Za testiranje DNSSEC sem postavil dva avtoritativna strežnika, tri rekurzivne strežnike in tri odjemalce z različnimi priljubljenimi različicami operacijskih sistemov. Avtoritativna strežnika sta najbolj razširjen strežnik DNS Bind in Windows Server 2008 R2, ki ga uporablja mnogo podjetij. Za rekurzivne strežnike, namen katerih je bil zlasti validiranje testnih domen, sem uporabil tako Bind in Windows Server kakor tudi strežnik DNS Unbound, ki bi naj bil zlasti primeren za uporabo kot rekurzivni strežnik. Več kot polovica strežnikov je imela IPv4- in IPv6-naslov. Testiral sem na treh različnih domenah (kozic.net, sah-drustvo-ms.si, in ltfe-sphere.org) in dveh obratnih conah (212.101.143.0/24 in 2a00:1368:1000:20::/64). Domene so bile ustrezno razporejene na različnih strežnikih DNS. Vse so bile vpete v hierarhijo DNS in so imele tako zapise IPv4 kakor tudi zapise IPv6. Glede zaupanja DNSSEC sem uporabil vse tri mogoče načine: torej puščice zaupanja, DLV in zapis DS, vpet v hierarhijo DNSSEC. Za preverjanje združljivosti različnih strežnikov DNS sem svojim domenam naredil poddomene, ki sem jih delegiral na drugi avtoritativni strežnik DNS (če je bila domena na Windows strežniku, sem poddomene delegiral na strežnik Linux in obratno). Vsi strežniki so bili virtualizirani. Okolje, na katerem so tekli, je bilo Microsoft Hyper-V in VMware ESX. Natančnejši prikaz testnega okolja je razviden na Sl. 16 in v Tab. 4, Tab. 5, Tab. 6, Tab. 7.

Tab. 4: Strežniki DNS, naslovi IP, platforma

Strežnik	Naslov IPv4	Naslov IPv6	Vrsta strežnika	Platforma
NS1.kozic.net	93.103.130.109	/	avtoritativni	Linux/ Bind 9.7
THRONE.ltfe-sphere.org	212.101.143.177	2a00:1368:1000:20::177	avtoritativni	Windows Server 2008 R2
LORDSHIP.second.ltfe-sphere.org	212.101.143.6	2a00:1368:1000:20::6	rekurzivni	Windows Server 2008 R2
STRONGHOLD1.second.ltfe-sphere.org	212.101.143.7	2a00:1368:1000:20::7	rekurzivni	Linux/ Bind 9.7
STRONGHOLD2.second.ltfe-sphere.org	212.101.143.7	2a00:1368:1000:20::7	rekurzivni	Linux/ Bind 9.7

Tab. 5: Domene na avtoritativnih strežnikih DNS

Avtoritativni strežnik	Domena	Vrsta zaupanja	Delegirana domena
NS1.kozic.net	kozic.net sah-drustvo-ms.si	DLV Trust Anchor	kozic.ltfe-sphere.org unsigned.ltfe-sphere.org
THRONE.ltfe-sphere.org	ltfe-sphere.org	DS (vpeta v hierarhijo)	ltfe.kozic.net unsigned.kozic.net

Tab. 6: Obratne cone na avtoritativnih strežnikih DNS

Avtoritativni strežnik	Obratna cona
NS1.kozic.net	0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa.
THRONE.ltfe-sphere.org	143.101.212.in-addr.arpa.

Tab. 7: Algoritmi za digitalno podpisovanje po conah

Avtoritativni strežnik	Cona	Algoritem	Dolžina ključa (KSK/ZSK)	Negativni odgovor
NS1.kozic.net	kozic.net.	7 (RSA/SHA-1)	2048/1024	NSEC3
	sah-drustvo-ms.si.	5 (RSA/SHA-1)	2048/1024	NSEC
	second.ltfe-sphere.org.	5 (RSA/SHA-1)	2048/1024	NSEC
	0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa.	8 (RSA/SHA-256)	2048/1024	NSEC
THRONE.ltfe-sphere.org	ltfe-sphere.org.	5 (RSA/SHA-1)	2048/1024	NSEC
	ltfe.kozic.net.	5 (RSA/SHA-1)	2048/1024	NSEC
	143.101.212.in-addr.arpa.	5 (RSA/SHA-1)	2048/1024	NSEC

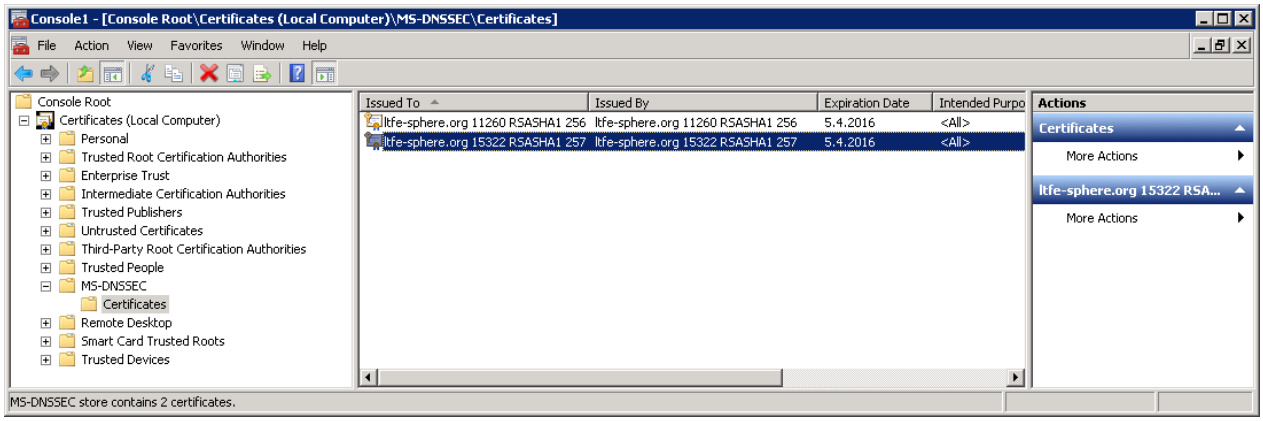
4.2 STREŽNIKI DNS S PODPORO DNSSEC

4.2.1 Avtoritativni strežniki DNS

4.2.1.1 Windows Server

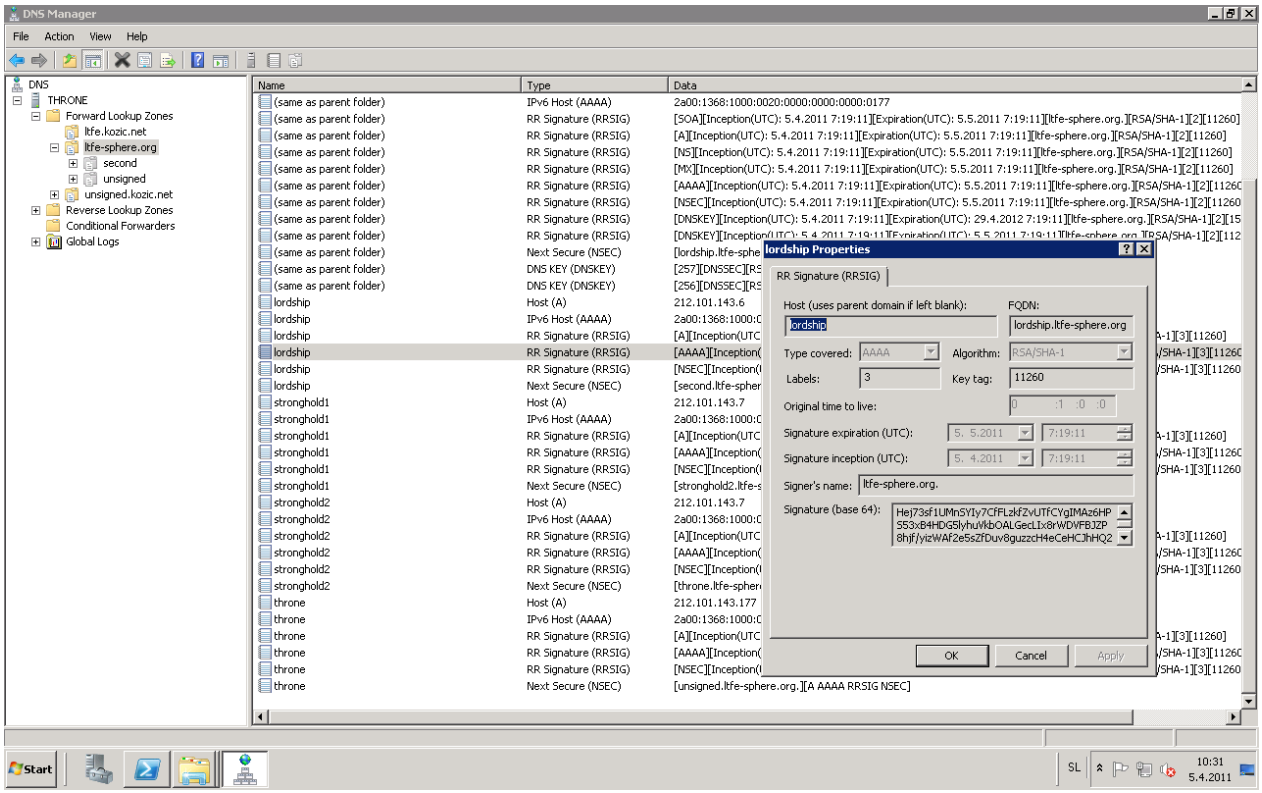
Windows Server ima podporo DNSSEC od različice Windows Server 2008 R2. Microsoft je ob implementaciji DNSSEC izdal literaturo DNSSEC Deployment Guide. Konfiguracija DNSSEC je v Windows Server 2008 R2 mogoča samo v ukazni vrstici (angl. Command Prompt), v grafičnem vmesniku pa je mogoče pregledovanje zapisov DNSSEC. Sicer se v grafičnem vmesniku izjemoma lahko vnese t. i. puščice zaupanja, kar je pa tudi vse. Windows Server ima implementirana samo osnovni standard RFC DNNSEC-bis (RFC 4033, 4034, 4035) iz leta 2005 in RFC 4509 iz leta 2006, ki omogoča uporabo SHA-256 v zapisih DS. Poznejše razširitve niso podprte, kar pomeni, da je Windows Server omejen na uporabo zapisov NSEC in uporabo podpisovanja z algoritmom RSA/SHA-1. [40]

Da je delo z DNSSEC poenostavljeno in uporabniku prijaznejše, je Microsoft izdal skripti PowerShell, ki nadalje kličeta ustrezne ukaze za delo z DNSSEC. Tako lahko z enim ukazom kreiramo ključa KSK in ZSK ter tudi podpišemo cono. Ključa se generirata kot digitalni potrdili in se tudi shranita v Certificate Storage, ki je centralna lokacija operacijskega sistema Windows za shranjevanje digitalnih potrdil. Primer podpisa domene se nahaja v prilogi 2.1.



Sl. 17: Certificate Storage, kjer sta shranjena ključa za DNSSEC

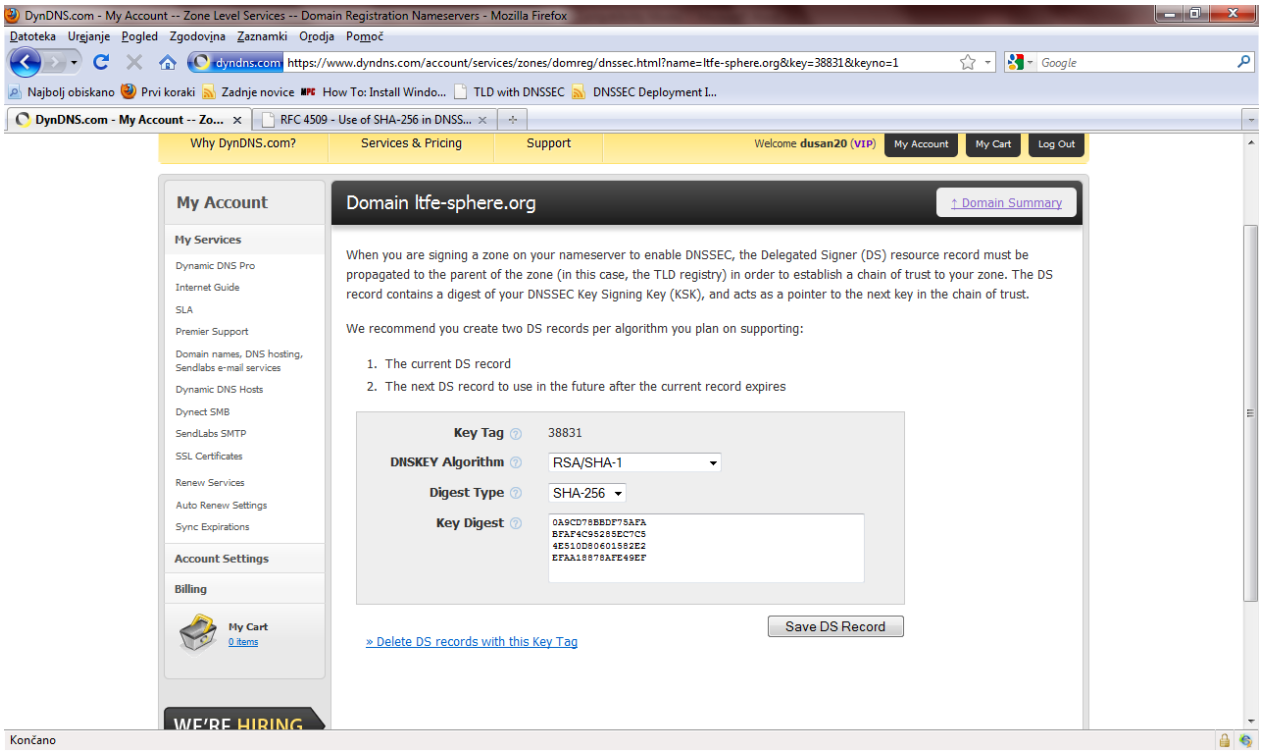
Sl. 18: Ključ KSK za domeno ltfe-sphere.org



Sl. 19: DNS Manager v Windows Server

Pregledovanje zapisov podpisane domene je na voljo v grafičnem vmesniku za upravljanje strežnika DNS. Vmesnik je lepo urejen in strukturiran. Da so zapisi DNSSEC pomešani med zapisi DNS, za administratorja ni tako problematično, saj se da zapise sortirati po njihovi vrsti. Iz grafičnega vmesnika lahko razberemo, da je KSK po privzetem veljaven eno leto, ZSK pa en mesec.

Pomanjkljivost PowerShell skripte Signzone.ps1 je, da kreira ključa KSK in ZSK, ki sta enake dolžine. Kakor sem že omenil, naj bi bil KSK daljši od ZSK in naj bi se tudi manj pogosto menjaval. PowerShell je uporabniku prijazen skriptni jezik, skripta je dobro napisana. Z dodatkom ene vrstice in spremembo ene spremenljivke sem dosegel, da se dolžini ključev po novem razlikujeta. Ker je na novo podpisana domena ltfe-sphere.org vpeta v hierarhijo DNSSEC, sem nov zapis DS preko grafičnega sporočila sporočil registrarju, ki je v mojem primeru DynDNS. Zapis DS je bil samodejno kreiran ob podpisovanju domene in se nahaja v datoteki dsset-*ime_domene*. Takoj po vnosu zapisa DS je bila domena uspešno validirana.



Sl. 20: Sporočanje zapisa DS registrarju

4.2.1.1.1 Ponovno podpisovanje cone

Podpisovanje cone v DNSSEC ni dovolj. Bistvena naloga je njeno vzdrževanje. Administrator namreč lahko naredi veliko škode, če ob poteku podpisov RRSIG »pozabi« ponovno podpisati cono. Domena v tistem času ne bo uspešno validirana, kar bo povzročilo njeno nedosegljivost.

Skripta Signzone.ps1 administratorju omogoča ponovno podpisovanje cone. S parametrom Resign preprosto ponovno podpišemo cono. Tak način je najlažji, saj nam ni treba ponovno kreirati ključev in jih sporočati registrarju. Le paziti moramo, da bomo zapise RRSIG podpisali vsaj za čas TTL prej, preden potečejo. Parameter Resign ima slabost, saj mu moramo podati imena ključev, s katerima smo cono že podpisali. Kar pomeni, da moramo odpreti Certificate Storage in izbrskati imena ključev KSK in ZSK. Primer se nahaja v prilogi 2.2.

4.2.1.1.1.1 Menjava ključa ZSK

Menjava ključa ZSK je interna zadeva, zato nam novega ključa prav tako ni treba sporočiti registrarju. Moramo pa paziti, da upoštevamo čase TTL. Skripta pri menjavi ključa ZSK deluje interaktivno, na voljo imamo oba načina menjave ključa ZSK: metodo enojnega podpisovanja in metodo dvojnega podpisovanja.

Pri menjavi ključa z enojnim podpisovanjem skripta najprej kreira nov ključ ZSK, doda nov ključ v cono, vendar cono podpiše samo s starim ključem. Po času TTL cono podpiše samo z

novim ključem, s tem da stari ključ še vedno pusti v coni. Ponovno počaka za čas TTL in po njem umakne stari ključ iz datoteke s cono. Za menjavanje ključa je torej potreben dvakratni čas TTL. Primer menjave ključa ZSK z metodo enojnega podpisovanja se nahaja v prilogi 2.3.

Če se odločimo za menjavo po metodi dvojnega podpisovanja, bo skripta najprej kreirala nov ključ ZSK, podpisala cono z novim in starim ključem (spet ji moramo podati imena starih ključev), čakala za čas TTL, po njem ponovno podpisala cono samo z novim ključem ZSK in umaknila stari ključ ZSK. Primer menjave ključa ZSK z metodo dvojnega podpisovanja se nahaja v prilogi 2.4.

4.2.1.1.1.2 Menjava ključa KSK

KSK vedno menjujemo z metodo dvojnega podpisovanja. Druge možnosti skripta Signzone niti ne podpira. Zaplete se, če je TTL zapisa DS v coni, ki je en nivo višje, višji od TTL v naši coni. Zato moramo skripti ročno sporočiti parameter TTL zapisa DS v coni en nivo višje (v nasprotnem primeru bi se KSK prehitro zamenjal, validacija naše domene pa bi bila neuspešna). V našem primeru imamo TTL cone ltfe-sphere.org nastavljen na 3600, vendar je TTL cone org bistveno višji (86400). Za menjavo KSK bomo zato porabili dva dni in ne zgolj 2 uri. Skripta najprej kreira ključ KSK, nam izpiše njegov zapis DS in nato čaka na uporabnika, da ta zapis sporoči registrarju oz. administratorju domene en nivo višje. Šele potem bo začela z menjavo ključa KSK. Primer menjave ključa KSK se nahaja v prologi 2.5.

Če povzamem, ima Windows Server omejeno podporo standardom DNSSEC, ne ponuja grafičnega vmesnika za delo z DNSSEC, samo upravljanje s ključi pa ima s pomočjo PowerShell skripte, ki jo je Microsoft sam napisal, dobro rešeno. Svoje zmožnosti DNSSEC ima tudi dokumentirane. Za avtoritativni strežnik DNSSEC je povsem primeren, če smo zadovoljni z uporabo algoritma RSA/SHA1 in manj varnega zapisa NSEC pri negativnih odgovorih. Ključe ima shranjene v obliki digitalnih potrdil v Certificate Storage, ki predstavlja centralno in varno lokacijo za hranjenje digitalnih potrdil. Sam strežnik DNS nima dostopa do digitalnih potrdil, skratka izkoriščanje morebitne luknje v strežniku DNS napadalcu ne bo dalo dostopa do privatnih ključev. Slaba lastnost pa je, da zaradi tega ni mogoč dinamičen DNS. [40]

4.2.1.2 Bind

Bind ima podporo DNSSEC od verzije 9.3 naprej. Prinaša nam tudi orodja, s katerimi lahko podpisujemo cone.

V konfiguraciji strežnika Bind najprej vklopimo podporo DNSSEC, tako da v named.conf.options dodamo »dnssec-enable yes«. Podpisovanje cone poteka v naslednjem vrstnem redu:

- kreiramo ključa KSK in ZSK,
- podpišemo cono.

Za podpisovanje cone je najprej treba kreirati ključe, kar storimo z orodjem dnssec-keygen. Ključ KSK sem kreiral z ukazom:[5]

```
dnssec-keygen -r /dev/urandom -a RSASHA512 -b 2048 -f KSK kozic.net
```

Parameter »r« določa, kateri generator naključnih števil uporabljamo. Datoteka /dev/urandom je generator psevdonaključnih števil v sistemu Linux. Če ta parameter izpustimo, je naključnost boljša, vendar je potem generiranje naključnih števil v virtualnih računalnikih prepočasno. S parametrom »a« določimo vrsto ključa (RSA/SHA-512), z »b« pa njegovo dolžino. S parametrom »f« povemo, da bomo kreirali ključ KSK, na koncu še podamo ime domene, za katero generiramo ključ. Kreirati je treba še ključ ZSK, ki sem ga generiral z ukazom:

```
dnssec-keygen -r /dev/urandom -a RSASHA512 -b 1024 kozic.net
```

Generator ključa kreira datoteki s ključem. Na začetek doda informacije, od kdaj je ključ veljaven, kdaj se začne proces menjave in kdaj ključ poteče. Bind od verzije 9.7 naprej zmore namreč sam menjavati ključe.
Ko imamo ključe generirane, moramo podpisati še cono. Le-to naredimo z ukazom:

```
dnssec-signzone -g -S -3 ab00 -H 2 -o kozic.net db.kozic.net
```

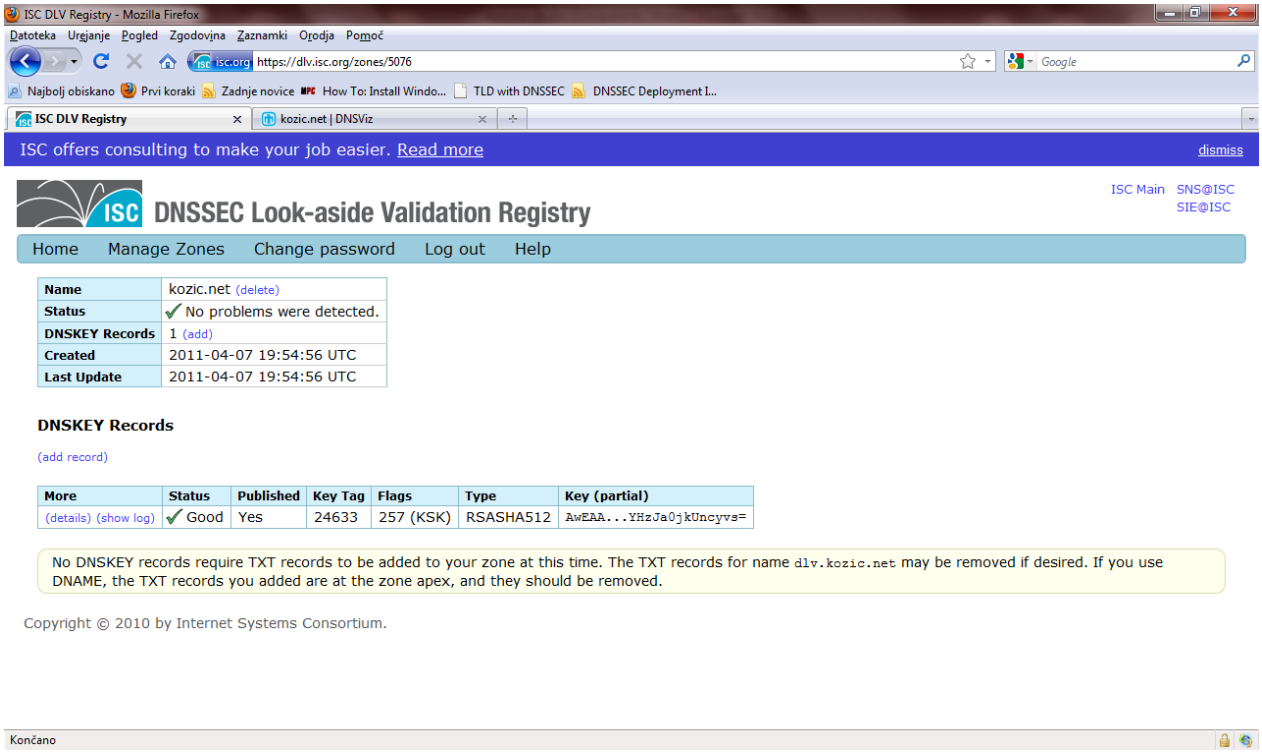
S parametrom »g« dosežemo, da se bodo podpisali tudi zapisi DS delegiranih poddomen. Domena ltfe.kozic.net je delegirana na Windows strežniku throne.ltfe-sphere.org. V direktorij, v katerem imamo cono, moramo prenesti datoteko z zapisom DS (dsset-ltfe.kozic.net). Orodje za podpisovanje cone bo potem pogledalo za datotekami dsset in v cono ustrezno dodalo ter podpisalo zapise DS. Parameter »S« določa, da Bind sam skrbi za menjavo ključev, ko le-ti potečejo. Parameter »3« pomeni, da smo domeno podpisali z NSEC3, ab00 je parameter sol, parameter »H« določa, kolikokrat bomo pognali zgoščevalno funkcijo nad zapisi NSEC3, »o« pomeni ime cone, na koncu pa še dodamo ime datoteke, v kateri so shranjeni zapisi DNS cone. Kot izhod bomo dobili datoteko db.kozic.net.signed (dodala se bo končnica signed), ki predstavlja cono z dodanimi zapisi DNSSEC (torej podpisano cono). V konfiguraciji DNS je treba popraviti pot do datoteke s cono, ker imamo avtomatiziran proces menjave ključev, je tudi treba strežniku Bind povedati, da naj sam skrbi za podaljševanje podpisov, ko le-ti potečejo. V zgornjem primeru nisem navedel datuma poteka zapisov RRSIG, po privzetem imajo vsi zapisi RRSIG življenjsko dobo enega meseca. Ker prav tako nisem navedel datuma poteka ključa, bo Bind, ko se bo začel približevati čas izteka veljavnosti zapisov, sam podaljšal zapise RRSIG (ker imam generiran samo en ključ, bo cono ponovno podpisal z istim ključem).

Za preverjanje veljavnosti domene kozic.net uporabljam servis ISC DLV. Po podpisu cone je preko spletnega vmesnika temu servisu treba sporočiti zapis DNSKEY. Nato ISC DLV od nas zahteva, da dokažemo, da je domena zares naša. Kreira zapis DLV, ki ga moramo vnesti v našo cono DNS, primer:

[5] Generiran ključ je drugačen, kakor je predvideno v topologiji testnega okolja. Upravljanje s ključi v Bind je bila vmesna faza, na koncu je za upravljanje ključev, ki jih je uporabljal Bind, skrbel OpenDNSSEC.

```
dlv.kozic.net.          0       IN      TXT     "DLV:1:htfccesknfsq"
```

Ključ DNSKEY je smiselno sporočiti strežniku DLV, še preden je cona dejansko z njim podpisana. V tem primeru se izognemo dvakratnemu podpisovanju cone.



Sl. 21: Nadzorna plošča registra ISC DLV

4.2.1.2.1 Menjava ključev

Od verzije 9.7 naprej Bind podpira delno avtomatizacijo pri menjavi ključev. Vsaka datoteka s ključi vsebuje glavo, v kateri je podanih nekaj sledečih informacij (zapisane so v obliki časa UTC kakor pri zapisu RRSIG):

- Created: kdaj je bil kreiran,
- Publish: kdaj naj se objavi,
- Activate: kdaj naj se z njim podpiše cona,
- Inactive: kdaj se z njim neha podpisovati cona (ključ je še vedno objavljen),
- Delete: kdaj bo ključ odstranjen iz cone,
- Revoke: kdaj naj bo ključ preklican (v tem stanju ima ključ nastavljeno posebno zastavico; ključ je še vedno objavljen, z njim pa je tudi cona še vedno podpisana.).

Preizkusil sem samodejno menjavo ključa ZSK za cono kozic.net, uporabil sem metodo enojnega podpisovanja. TTL cone je 3600. Najprej sem moral kreirati nov ključ, ki bo zamenjal obstoječega. Novemu ključu sem nastavil čas objave in aktivacije:

```
dnssec-settime -P 20110429163000 -A 20110429173000 Kkozic.net.+010+49457
```

Ključ bo objavljen 29. aprila 2011 ob 18:30, aktiviran pa 29. aprila 2011 ob 19:30. Stari ključ mora postati deaktiviran 29. aprila ob 19:30 (čas začetka menjave + TTL), odstranjen pa 29. aprila ob 20:30 (čas začetka menjave + 2*TTL):

```
dnssec-settime -I 20110429173000 -D 20110429183000  Kkozic.net.+010+30950
```

Menjava se je izvedla ob predpisanih časih. Problematično je, da Bind ne preverja, kako smo nastavili čase (tako lahko pridemo v stanje, ko je domena nepravilno podpisana). Bind za nas tudi ne bo kreiral novih ključev, zato proces menjave ključev v Bind ni povsem avtomatiziran, je pa administratorju lažje, saj lahko pravočasno vnaprej nastavi, kdaj se bodo izvedle menjave. Za potek zapisov RRSIG pa administratorju ni treba skrbeti, saj Bind po potrebi ponovno podpiše domeno, če le ima na voljo veljaven ključ (dobro je pustiti, da nazadnje kreirani ključ nima datuma poteka veljavnosti). [42]

Primer konfiguracije strežnika Bind in izgled datoteke s ključem sta v prilogi 3.

4.2.1.2.2 Dinamičen DNS in Bind

Ob uporabi dinamičnega DNS je ob spreminjanju zapisov v sistemu DNS treba spremenjene zapise tudi podpisovati. Za to opravilo mora imeti strežniški program ves čas dostop do privatnega ključa, kar je sicer potrebno tudi v prejšnjem primeru, ko Bind samodejno ponovno podpisuje cono in menjuje ključe. Cena dinamičnosti je nižja varnost.

Vse, kar je potrebno storiti za vklop dinamičnega DNS, je, da se v konfiguraciji cone nastavi, kateri naslovi IP lahko spreminjajo podatke cone. To naredimo tako, da v konfiguracijo posamezne cone dodamo stavek allow-update { 127.0.0.1; 192.168.1.0/24; }; S stavkom določimo, kateri naslovi IP lahko spreminjajo vsebino cone. Določil sem, da lahko vsebino cone kozic.net spreminjajo računalniki iz privatnega omrežja 192.168.1.0/24. Podrobnejša konfiguracija Bind za dinamičen DNSSEC se nahaja v prilogi 3.1.
Sistem Windows po privzetem ob zagonu sporoči svojemu strežniku DNS svoje ime (sporočilo DNS update). Vse, kar sem naredil, je, da sem pod nastavitvami DNS vpisal, kateri domeni DNS pripada moj računalnik (kozic.net), odjemalec Windows je potem strežniku sporočil svoje ime. Proces je v sodelovanju s strežnikom DHCP, ki odjemalcu sporoči ime domene, popolnoma avtomatiziran.

V strežniku DNS se je pojavil vnos NOAH.kozic.net (moji delovni postaji je ime NOAH), ki je bil tudi ustrezno podpisan. Mehanizem dinamičnega DNS, ki se zanaša samo na naslove IP, ne deluje najbolj varno. Bind prav tako omogoča uporabo mehanizma TSIG, ki omogoča avtentikacijo in poskrbi za vzpostavitev varnega kanala med odjemalcem in strežnikom.

4.2.1.3 NSD

NSD je avtoritativni strežnik DNS. Na njem tudi teče nekaj strežnikov korenske domene. NSD nam ne prinaša nobenih orodij za podpisovanje in vzdrževanje DNSSEC. V njem sploh ni treba vklopiti DNSSEC, ko vključimo cono, ki je podpisana, se NSD tega sam zave in jo začne obravnavati kot DNSSEC-podpisano cono.

NSD podpira vse standarde RFC, povezane z DNSSEC. V prvotnem testnem okolju strežnika NSD nisem nameraval testirati, vendar sem se zaradi njegove razširjenosti vseeno odločil, da ga omenim.

Čeprav NSD nima nobenih orodij za podpisovanje in vzdrževanje DNSSEC, nam NLnet Labs ponuja paket ldns, ki nam ta orodja prinaša skupaj z orodji za preverjanje delovanja DNSSEC:

- orodje ldns-keygen je ekvivalentno Bindovemu orodju dnssec-keygen,
- orodje ldns-signzone je ekvivalentno Bindovemu orodju ldns-signzone.

Tudi sintaksi sta si podobni.

Strežnik NSD sicer ne podpira samodejnega vzdrževanja cone in dinamičnega DNS nasploh. Najbolj smiselno ga je uporabljati v kombinaciji s profesionalnimi orodji za upravljanje in vzdrževanje DNSSEC, ki so opisana v 4.2.1.4.

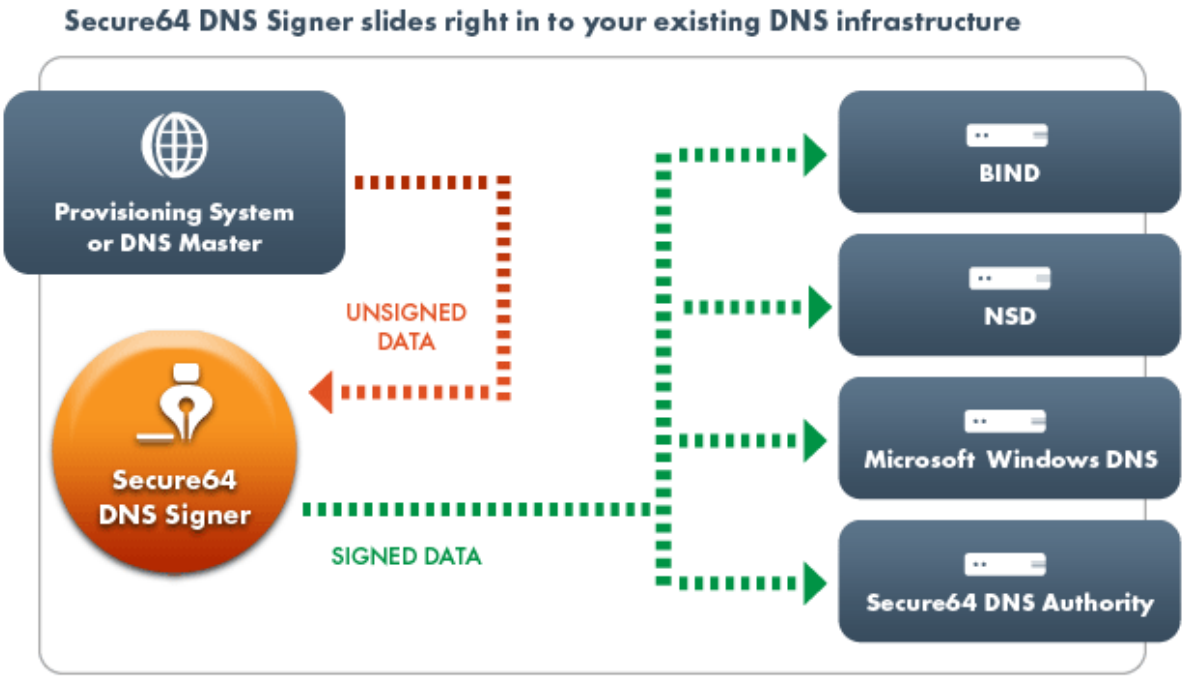
Sam sem konfiguracijo Bind na svojem Linux avtoritativnem strežniku migriral v NSD in tudi tam preizkusil serviranje DNSSEC. Sprememb pri delovanju ni bilo za opaziti. Datotek s conami pa mi sploh ni bilo treba ponovno podpisovati, saj so popolnoma združljive z datotekami od Bind, kar je sicer pričakovano, saj so zapisi DNSSEC standardizirani.

4.2.1.4 Profesionalne rešitve za DNSSEC in upravljanje s ključi

Kakor je bilo mogoče opaziti, je vzdrževanje con in upravljanje s ključi pri obeh strežnikih polavtomatsko. Pri upravljanju s ključi in podpisovanju con lahko hitro pride do nepričakovanih negativnih dogodkov, zato so potrebne rešitve, pri katerih je možnost neljubih dogodkov praktično nična. Različna podjetja so izdelala različne rešitve, namenjene upravljanju s ključi in podpisovanju con. Nekaj takih rešitev:

- Secure64 DNS Signer,
- Infoblox,
- Xelerance DNSX Secure Signer,
- Bluecat Networks DNSSEC Solution.

Gre za posebno strojno opremo, ki generira ključe, podpisuje datoteke s conami in jih redistribuira na strežnike DNS. Združljiva je s strežniki DNS različnih proizvajalcev. Proizvajalci zagotavljajo, da stranka za prehod na DNSSEC ne potrebuje nobenega predznanja, da je prehod na DNSSEC z njihovo rešitvijo hiter in preprost ter da za vse potrebno v zvezi z DNSSEC poskrbi njihova rešitev. Za vklop DNSSEC na Secure64 DNS Signer je potrebna zgolj ena vrstica v konfiguracijski datoteki. Rešitve podpirajo tudi bolj podrobno konfiguracijo, še vedno pa naj bi jo bilo mogoče izvesti na preprost način.

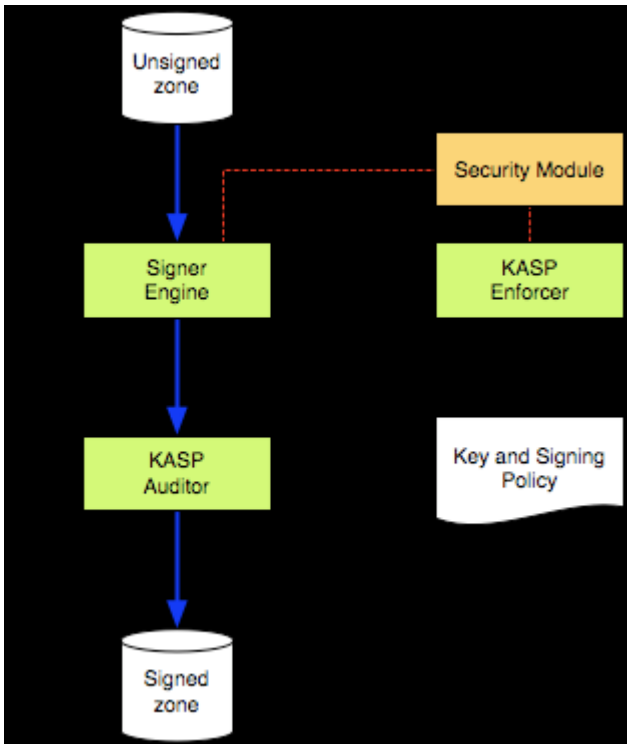


Sl. 22: Proces podpisovanja con pri Secure64 DNS Signer. Vir: secure64.com

Strojna oprema uporablja poseben kriptografski modul za generiranje, shranjevanje in podpisovanje s ključi, HSM, ki ustreza standardu FIPS 140-2 (angl. Federal Information Processing Standard). Standard FIPS je standard ameriške vlade, ki vrednoti kriptografske module. Obstajajo 4 stopnje, HSM z višjo stopnjo je boljši in varnejši, Secure64 dosega stopnjo 2, Xelerance pa stopnjo 3. Na HSM so ključi shranjeni varno, tako je z oddaljenim dostopom do naprave mogoče zgolj podpisovanje cone in generiranje novih ključev, izvoz privatnega ključa pa ni mogoč (če napadalec vdre v sistem, se ne more dokopati do ključev). [46]

4.2.1.4.1 OpenDNSSEC

Denarja za nakup in preizkušanje komercialnih rešitev za upravljanje s ključi seveda nimam, vendar obstajajo tudi odprtokodne rešitve za upravljanje s ključi. Ena boljših je OpenDNSSEC.



Sl. 23: Arhitektura OpenDNSSEC [45]

Arhitektura OpenDNSSEC je prikazana na zgornji sliki:

- KASP Enforcer je strežniški program, ki skrbi za generiranje in zamenjavo ključev.
- Signer Engine je strežniški program, ki skrbi za ponovno podpisovanje datotek s conami.
- KASP Auditor je orodje, ki skrbi, da so podatki v coni konsistentni. Preden se podpisana cona naloži na strežniku, gre skozi Auditor, ki preveri, ali je podpisana pravilno.

OpenDNSSEC ima natančno definirano politiko, ki določa generiranje ključev, njihovo menjavo in ponovno podpisovanje con. Politika se običajno nahaja v direktoriju /etc/opendnssec in je sestavljena iz naslednjih datotek:

- kasp.xml: KASP (angl. Key and Signature Policy) je politika ključev in podpisov. V njej uporabnik definira različne politike, ki določajo trajanje podpisov, algoritem ključev KSK in ZSK ter njihovo veljavnost (po poteku veljavnosti se ključi zamenjajo), algoritem za negativne odgovore (NSEC, NSEC3 oz. NSEC3 Opt-out).
- zonelist.xml: v njej je seznam in pot do con, ki jih upravljamo v OpenDNSSEC. Posamezni coni mora biti dodeljena ena izmed politik definiranih v kasp.xml (več različnih con lahko uporablja isto politiko).
- zonefetch.xml: če uporabljamo TSIG, lahko v datoteki definiramo potreben ključ, da datoteko s cono snamemo s strežnika in jo potem, ko se podpiše, ponovno naložimo na strežnik. Večje organizacije pogosto uporabljajo ločeni strežnik DNS, ki servira cone, in ločeni strežnik, na katerem cone podpisujemo.
- conf.xml: splošna konfiguracijska datoteka, ki določa pot do ostalih konfiguracijskih datotek, vsebuje podatke za dostop do HSM in definira obnašanje posameznih modulov OpenDNSSEC (KASP Enforcer, Signer, KASP Auditor).

OpenDNSSEC tako kot komercialne rešitve uporablja za shranjevanje in generiranje ključev HSM. Za dostop do HSM uporablja vmesnik, definiran s standardom PKCS #11. Podprti so kriptografski moduli različnih proizvajalcev, ki govorijo ta jezik. OpenDNSSEC je razvil tudi programski HSM, imenovan SoftHSM. Tako lahko OpenDNSSEC uporabljajo tudi tisti, ki nimajo denarja ali pa potrebe za nakup HSM. [45]

Na strežniku Linux NS1.kozic.net je strežnik Bind za upravljanje s ključi in conami uporabljal rešitev OpenDNSSEC. Na ta način strežnik Bind ni imel veliko opravkov v zvezi z DNSSEC, skrbel je samo za serviranje že podpisanih con. Politika DNSSEC je bila skladna s Tab. 7. Menjava ključev ZSK je bila popolnoma avtomatizirana, v skladu z definirano politiko so se v HSM generirali novi ključi, stari ključi pa so se brisali.

Zaplete se pri menjavi ključa KSK, saj le-ta ne more biti popolnoma avtomatizirana. Ključ KSK sem menjal po sledečem postopku:
1) Najprej sem z ukazom sprožil menjavo KSK. Nov KSK, ki je zamenjal obstoječi KSK, je prešel v stanje publish, kar pomeni, da je bil od takrat naprej objavljen v coni.
2) Počakati sem moral, da je ključ iz stanja publish prešel v stanje ready. Šele ko se je zgodil ta prehod, so vsi strežniki DNS videli nov ključ KSK. Po tem sem sporočil zapis DS novega KSK coni en nivo višje.
3) Ko sem sporočil zapis DS coni en nivo višje, sem z ukazom to tudi sporočil OpenDNSSEC. Novi ključ KSK je prešel v stanje active, stari ključ KSK pa je prešel v stanje retire. Dokler je bil stari ključ v tem stanju, je bil še vedno v coni. Ključ je bil v stanju retire, dokler ni potekel TTL zapisa DS.
4) Ključ je iz stanja retire samodejno prešel v stanje dead., v katerem ni bil več objavljen v coni. Ključ se je tudi samodejno izbrisal v času, ki sem ga nastavil v konfiguraciji OpenDNSSEC.

Konfiguracija OpenDNSSEC, ki sem jo uporabljal, je v prilogi 4.

4.2.1.5 Domena TLD SI

Arnes, register vrhnje domene SI, je trenutno v testni fazi uvajanja DNSSEC. Ima postavljen vzporedni sistem DNS, na katerem so določene domene že podpisane. Zainteresirani za testiranje DNSSEC na svoji domeni lahko vzpostavijo kontakt z Arnesom. Tudi sam sem se pridružil testiranju DNSSEC na vrhnji domeni SI in sem sporočil zapis DS svoje domene SI (sah-drustvo-ms.si). Za preizkus validacije svoje domene sem moral vzpostaviti stik z alternativnim rekurzivnim strežnikom DNS, ki zmore validirati domeno SI. Nahaja se na naslovu IP 193.2.1.79 (dnssec-recursive.arnes.si). Domena sah-drustvo-ms.si je bila uspešno validirana.
Arnes v testnem okolju za podpisovanje DNSSEC uporablja rešitev OpenDNSSEC, ki teče na dodeljenem svojem računalniku. Domena SI se podpisuje na dve uri, in sicer na način, da se cona sname z avtoritativnega strežnika, se podpiše in se nato podpisana prenese na avtoritativne strežnike. V testnem okolju so avtoritativni strežniki Bind, v produkcijskem okolju jih je večina Bind, nekaj pa tudi NSD. OpenDNSSEC je namenjen avtomatizaciji procesa podpisovanja in menjave ključev. Računalnik z OpenDNSSEC za upravljanje s ključi

KSK uporablja kriptografski modul HSM Sun SCA 6000. Tudi v primeru vdora v strežnik, ki podpisuje cono SI, se napadalec ne more dokopati do privatnega ključa, shranjenega na kartici. Za dostop do privatnega ključa je potreben fizični dostop. Za upravljanje s ključi ZSK pa OpenDNSSEC uporablja programsko rešitev SoftHSM, ki sem jo tudi sam testiral. Arnes za podpisovanje cone uporablja algoritem 8 (RSA/SHA-256), KSK je dolžine 2048 bitov, ZSK pa dolžine 1024 bitov. Za negativne odgovore uporablja NSEC3, saj NSEC zaradi možnosti sprehajanja po coni ni dopustna rešitev. Ker ima cona SI več kot 90000 delegacij, se uporablja NSEC3 Opt-out (s tem je velikost cone bistveno manjša).

4.2.2 Rekurzivni strežniki DNS

4.2.2.1 Windows Server

V Windows Server validiramo domene z vnosom t. i. puščic zaupanja (angl. Trust Anchors). V rekurzivni strežnik DNS vnesemo ključ DNSKEY od posamezne vstopne točke (npr. ltfe-sphere.org), delegirane poddomene pa potem strežnik validira od tam naprej (strežnik pogleda zapis DS delegirane poddomene second.ltfe-sphere.org in na ta način validira ključ DNSKEY). Kot puščico zaupanja lahko torej dodamo korensko domeno in na ta način validiramo celotno verigo DNSSEC. To bi bilo mogoče, če bi bila korenska domena podpisana z algoritmom RSASHA1 in bi uporabljala zapis NSEC za validacijo negativnih odgovorov (v nadaljevanju algoritem 5), ki je edini algoritem za podpisovanje, katerega za zdaj podpira Windows Server. Prav tako strežnika Windows Server ne moremo uporabljati za validacijo od poljubne vrhnje domene naprej. Vse vrhnje domene bolj ali manj uporabljajo NSEC3, saj si ne želijo, da bi se uporabniki lahko sprehajali po coni. [40]

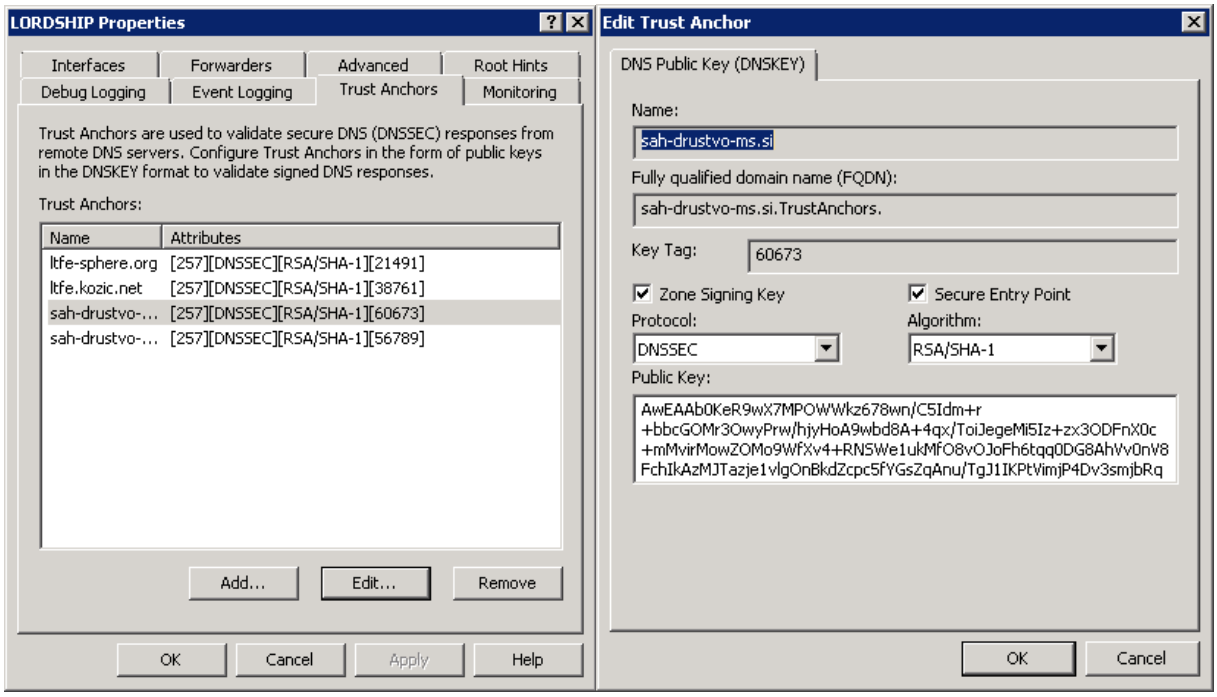
Windows Server je uporaben za validacijo DNSSEC znotraj posamezne organizacije, če za DNSSEC uporablja algoritem 5. Za vse svoje domene, ki uporabljajo algoritem 5, sem v rekurzivnem strežniku Windows Server dodal puščice zaupanja. Potem sem z orodjem dig preverjal ali strežnik DNS Windows Server uspešno validira domeno (opazoval sem zastavico AD).
Puščico zaupanja uporabnik lahko vnese s pomočjo skripte PowerShell oz. ukaza dnscmd, ki ga ta skripta kliče, ali pa v grafičnem vmesniku konfiguracije strežnika Windows DNS. Vnesti je treba zapis DNSKEY, ki ga lahko najdemo v datoteki keyset-*ime_domene* ali pa ga dobimo iz poizvedovanja DNSSEC. Primer uporabe ukaza TrustAnchor:

```
PS C:\Users\Administrator\Desktop\dnssec> .\TrustAnchor.ps1 -Action
Add  -Zone ltfe-sphere.org -Keyset .\keyset-ltfe-sphere.org

Action: add trust anchor

Add:
AwEAAaCurcvVVgLLAGyOae0VRI4RgS/UztGgSwV18CqZ2Z1ndj8y0aQy6sxGdsOSin42
LEShOqMPOgLMkRZl/hHIQDRVOtmHwS4rjvHjR9DA9ju2MU
H5TgLm9pdHiO66Vgh+njtEPwEzgvN0xm216vbXdJ8M7nvyveN4Xpv38AUMzoC2I/iecJ
yMgcJGOgn3iQvssNPZGjuPKjbXAhspMzuPqF0WE13A3UhzvIt5r
CyGQ6lIDdLugeymuzFSJ7rss97yIV1dItiFYopb8V6qzuturFmXTwkTxhZrN4p72JfSw
/c03ynC7zj03o2NIeiKNxuPM7po89NRyzObiDfHnDC9byE=

Completed: added trust anchor for zone ltfe-sphere.org
```



Sl. 24: Grafični vmesnik za vnos puščic zaupanja

Za validacijo domen sem večinoma uporabljal orodje dig. Prikaz izpisa orodja dig se nahaja v poglavju 4.3.6.1.1. Validiral sem vse svoje domene DNS, vključno z obratnimi conami. Ena izmed domen (second.ltfe-sphere.org) je bila namerno pokvarjena. Na ta način sem dodatno hotel preveriti, ali bo rekurzivni strežnik DNS zaznal napačno podpisano domeno. Prikaz rezultatov je podan v Tab. 8.

Tab. 8: Validacija domen v Windows Server DNS

Cona	Algoritem	Način zaupanja	Validacija uspešna	Naslov razrešen
sah-drustvo-ms.si	RSASHA1 NSEC	puščica	DA	DA
ltfe-sphere.org	RSASHA1 NSEC	puščica	DA	DA
->second.ltfe-sphere.org	RSASHA1 NSEC	DS (pokvarjen)	NE	NE
->unsigned.ltfe-sphere.org	ni podpisana	/	NE	DA
kozic.net	RSASHA1 NSEC3	/	NE	DA
->ltfe.kozic.net	RSASHA1 NSEC	puščica	DA	DA
->unsigned.kozic.net	ni podpisana	/	NE	DA
0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa.	RSASHA256 NSEC	/	NE	DA
143.101.212.in-addr.arpa.	RSASHA1 NSEC	puščica	DA	DA

Validiranje DNSSEC za domene, ki smo jih vnesli kot puščice zaupanja, se izvede tudi v primeru, ko uporabnik pošlje navadno poizvedbo DNS. V primeru neuspešne validacije strežnik ne vrne odgovora tudi na navadno poizvedbo (vrne napako strežnika SERVFAIL). Do napake SERVFAIL je prišlo pri delegirani poddomeni second.ltfe-sphere.org, ki ima en nivo višje namerno vnesen napačen zapis DS. Za nepodpisano delegirano poddomeno unsigned.ltfe-sphere.org sem dobil odgovor, vendar brez zastavice AD. Če delegirana poddomena z zapisom DS ni podpisana, bo strežnik uporabnika pustil nanjo, če je podpisana nepravilno, pa ne. V obeh primerih bo odgovor brez zastavice AD.

4.2.2.2 Bind

Validacijo v Bind vklopimo tako, da dodamo naslednji vrstici v datoteko named.conf.options:

```
dnssec-enable yes;
dnssec-validation yes;
```

Vpisati je treba še ključ KSK korenske domene (ključ je seveda treba prenesti na varen način):

```
managed-keys {
        "." initial-key 257 3 8
"AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
```

```
X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
QxA+Uk1ihz0=";
};
```

Ključ bomo vpisali samo enkrat, naprej pa bo strežnik Bind sam posodabljal ključ KSK, ko se bo spreminjal. Procedura je opisana v RFC 5011 [41].

Z vnosom ključa KSK korenske domene lahko validiram samo domeno ltfe-sphere.org, saj je edina, ki je vpeta v hierarhijo DNSSEC. Za preverjanje domene kozic.net uporabljam servis ISC DLV, katerega uporabo Bind prav tako omogoča. Za uporabo tega servisa v named.conf.options dodamo še vrstico dnssec-lookaside . trust-anchor dlv.isc.org.;. Rekurzivni strežnik Bind bo sedaj najprej skušal domeno validirati skozi klasično hierarhijo DNSSEC, če pa v njej ne bo našel vnosa DS, bo šel poizvedovat na alternativno drevo ISC DLV. Domeno sah-drustvo-ms.si validiram skozi puščice zaupanja. Ključ puščice zaupanja se prav tako doda pod managed-keys.

Tab. 9: Validacija domen v Bind

Cona	Algoritem	Način zaupanja	Validacija uspešna	Naslov razrešen
sah-drustvo-ms.si	RSASHA1 NSEC	puščica	DA	DA
ltfe-sphere.org	RSASHA1 NSEC	DS	DA	DA
->second.ltfe-sphere.org	RSASHA1 NSEC	DS (pokvarjen)	NE	NE
->unsigned.ltfe-sphere.org	ni podpisana	/	NE	DA
kozic.net	RSASHA1 NSEC3	DLV	DA	DA
->ltfe.kozic.net	RSASHA1 NSEC	DS	DA	DA
->unsigned.kozic.net	ni podpisana	/	NE	DA
0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa.	RSASHA256 NSEC	puščica	DA	DA
143.101.212.in-addr.arpa.	RSASHA1 NSEC	puščica	DA	DA

Strežnik Bind podpira vse algoritme, ki jih DNSSEC uporablja za podpisovanje, zato je bila validacija vseh pravilno podpisanih domen uspešna. Pri nepravilno podpisani domeni second.ltfe-sphere.org je strežnik vrnil napako tudi v primeru, ko uporabnik ni sprožil zahteve DNSSEC. Strežnik je napako tudi zapisal v dnevniško (angl. log) datoteko:[6]

[6] V strežniku Bind lahko podrobno in ločeno od ostalih procesov beležimo proces poizvedovanja in validiranja DNSSEC.

```
Apr 26 23:13:01 dk-dnssec-us1004 named[32450]: validating @0xb7e7dee0:
second.ltfe-sphere.org DNSKEY: no valid signature found (DS)
Apr 26 23:13:01 dk-dnssec-us1004 named[32450]: error (no valid RRSIG)
resolving 'second.ltfe-sphere.org/DNSKEY/IN': 93.103.130.109#53
Apr 26 23:13:01 dk-dnssec-us1004 named[32450]: error (broken trust chain)
resolving 'second.ltfe-sphere.org/A/IN': 93.103.130.109#53
Apr 26 23:13:05 dk-dnssec-us1004 named[32450]: validating @0xb80da558:
second.ltfe-sphere.org A: bad cache hit (second.ltfe-sphere.org/DNSKEY)
Apr 26 23:13:05 dk-dnssec-us1004 named[32450]: error (broken trust chain)
resolving 'second.ltfe-sphere.org/A/IN': 93.103.130.109#53
```

Za nepodpisani delegirani poddomeni unsigned.ltfe-sphere.org in unsigned.kozic.net sem dobil odgovor, vendar brez zastavice AD. Strežnik Bind bo tako kot strežnik Windows v primeru, da delegirana poddomena ni podpisana, uporabnika spustil nanjo, v primeru, da je pa podpisana nepravilno, pa ne. V obeh primerih bo odgovor brez zastavice AD. Logika delovanja rekurzivnih strežnikov Windows Server DNS in Bind je torej podobna, le da ima Bind poln nabor podprtih algoritmov, podpira samodejno menjavo ključev in uporabo alternativnih servisov DLV. [42]

4.2.2.3 Unbound

Unbound je strežnik DNS, ki se uporablja za rekurzivno poizvedovanje DNS. DNSSEC validacijo vklopimo v konfiguracijski datoteki (unbound.conf) z:

```
auto-trust-anchor-file: "root-trust-anchor"
```

Z auto povemo, da naj strežnik Unbound sam posodablja ključ KSK korenske domene po proceduri, opisani v RFC 5011. Prvič prenesemo ključ v datoteko ročno. Lahko vpišemo ključ KSK ali pa zapis DS ključa KSK. Unbound prav tako omogoča uporabo servisa DLV, ki jo vklopimo, da v konfiguracijo strežnika Unbound dodamo vrstico:

```
dlv-anchor-file: "dlv-isc-key"
```

Ključ ponovno prenesemo ročno. Unbound sicer ne podpira samodejnega posodabljanja ključa DLV. Puščico zaupanja za domeno sah-drustvo-ms.si sem prav tako dodal s parametrom auto-trust-anchor-file.
Nivo beleženja neuspešnega validiranja DNSSEC v dnevniške datoteke nastavimo v konfiguracijski datoteki z:

```
val-log-level: 0
(0 – izklopljeno, 1 – vklopljeno, 2 – podrobno)
```

Strežnik Unbound je domene validiral podobno kakor strežnik Bind (Tab. 9). Obnašanje ob neuspešni validaciji je bilo enako kakor pri strežnikoma DNS Windows Server in Bind. Konfiguracija strežnika Unbound za validiranje DNSSEC je malenkost preprostejša kakor pri strežniku Bind, poleg tega lahko pri strežniku Unbound vnesemo zapise DS namesto zapisov DNSKEY. [43]

4.2.3 Primerjava

V Tab. 10 je podan povzetek primerjave rekurzivnih strežnikov DNS, v Tab. 11 pa povzetek primerjave avtoritativnih strežnikov DNS.

Tab. 10: Primerjava rekurzivnih strežnikov DNS

	Windows Server DNS	Bind	Unbound
okolje za upravljanje	grafični vmesnik in Powershell	ukazna lupina	ukazna lupina
podprti vsi algoritmi	samo algoritem 5 (RSA/SHA-1 NSEC)	Da	Da
podpora DLV	Ne	Da	Da

Tab. 11: Primerjava avtoritativnih strežnikov DNS

	Windows Server	Bind	NSD
Podprti algoritmi	samo RSA/SHA-1	vsi	vsi
Negativni odgovori	NSEC	NSEC, NSEC3	NSEC, NSEC3
Upravljanje DNSSEC preko grafičnega vmesnika	samo vpogled	ne	ne
Upravljanje s ključi	dobro (ključi kot digitalna potrdila shranjeni v Certificate Storage)	ključi shranjeni v datoteki	ključi shranjeni v datoteki
Privatni ključ zavarovan	da (strežnik nima dostopa do njega)	ne (lahko spremenimo pravice in s tem omejimo funkcionalnost)	da (strežnik nima dostopa do njega)
Dinamičen DNSSEC	ne	da (nižja varnost)	ne
Samodejno vzdrževanje cone	ne	deloma	ne
Zunanja orodja za vzdrževanje cone	komercialna (dokler podprt samo algoritem 5 vprašanje smiselnosti)	komercialna in odprtokodna	komercialna in odprtokodna

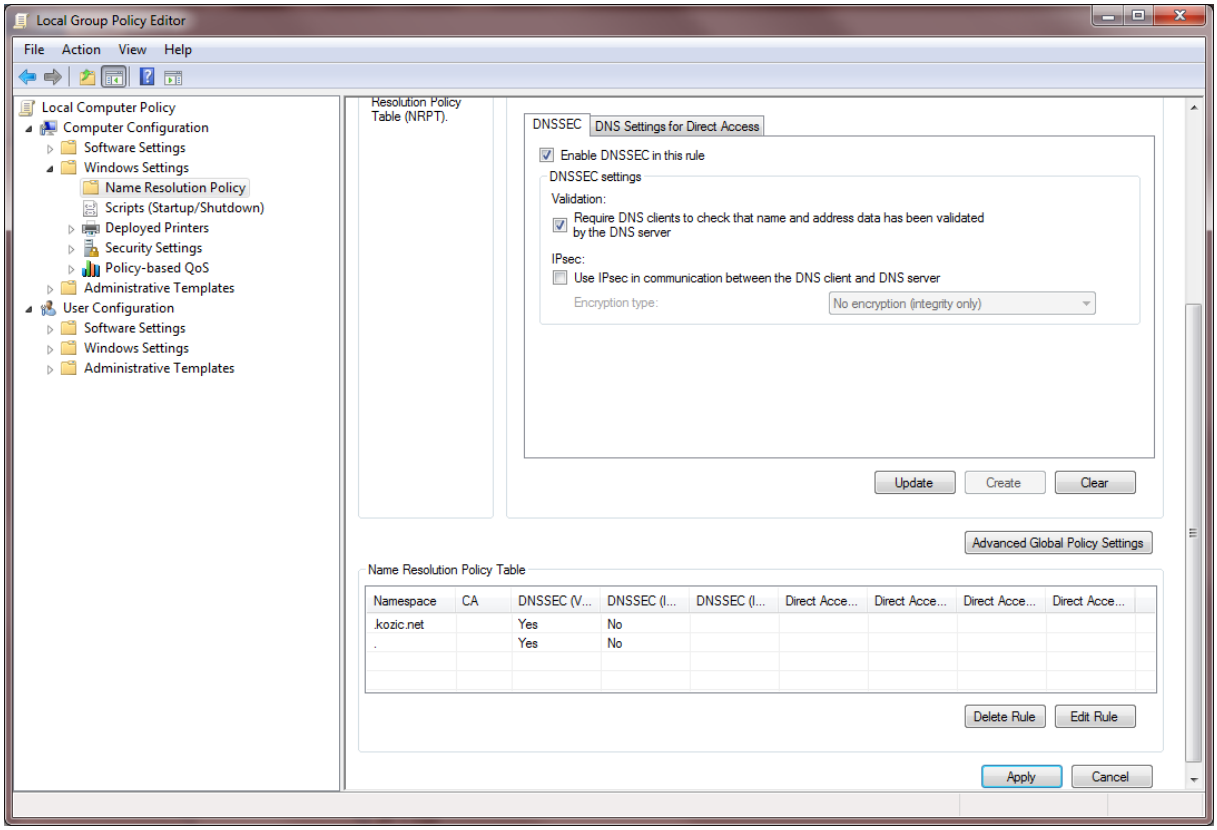
Pri sistemu Windows mi je všeč, da obstaja grafični vmesnik za pregled DNSSEC, pogrešam pa možnost konfiguriranja. Prav tako bi v sistemu Windows pričakoval hitrejšo podporo standardom DNSSEC. Strežnik Unbound mi je všeč zaradi svoje preprostosti konfiguracije, pri NSD pa mi je všeč, da DNSSEC na njem ni treba posebej vklopiti. Na splošno podpiram

idejo ločitve rekurzivnega in avtoritativnega strežnika DNS. Bind se mi glede podprtosti DNSSEC in različnih orodij, ki jih v zvezi z njim prinaša, zdi razred zase. Všeč mi je tudi, da obstajajo rešitve za vzdrževanje DNSSEC, ki so ločene od samega strežnika DNS. Zaradi kompleksnosti vzdrževanja DNSSEC se mi namreč zdi smiselno ločiti orodja za vzdrževanje od samih strežnikov, ki servirajo cone.

4.3 ODJEMALCI S PODPORO DNSSEC

4.3.1 Odjemalci v okolju Windows

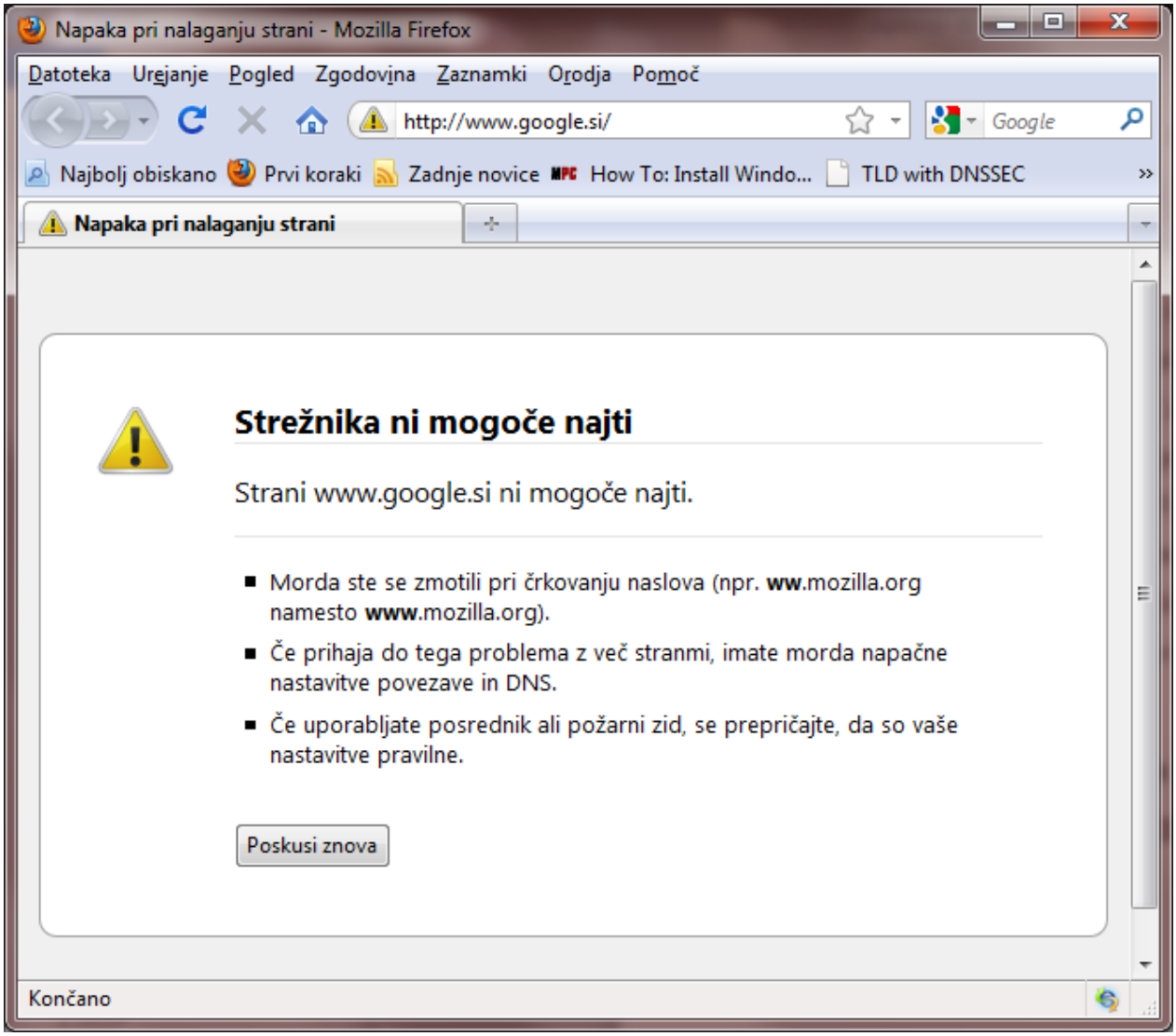
Ob implementaciji DNSSEC v strežniku DNS Windows Server je Microsoft implementiral tudi uporabo DNSSEC na odjemalcih. Podprta sta klienta Windows Vista in Windows 7. DNSSEC validiranje se vklopi kot del politike DNS sistema Windows. Tako nastavljena politika razreševanja imen DNS potem velja za vse aplikacije, ki znotraj sistema uporabljajo DNS; tako Microsoftove kakor tudi drugih proizvajalcev. Politika se lahko vklopi samo na lokalnem računalniku, če pa je računalnik del domene Windows, pa se lahko vklopi na domenskem kontrolerju in nadalje velja za vse računalnike v domeni. [40]



Sl. 25: Politika razreševanja naslovov DNS sistema Windows

Validiranje lahko vklopimo za vse domene, za domene z določeno končnico (npr. *.kozic.net) pa tudi za domene, ki se začnejo z določenim imenom (npr. kozic.*). Sam odjemalec Windows ne izvaja validacije DNSSEC, ampak se pri tem zanaša na strežnik DNS. Pri tem

postane najšibkejša točka komunikacije tista med odjemalcem in rekurzivnim strežnikom, zato nam sistem Windows tudi omogoča, da komunikacijo med njima zavarujemo z IPsec.
V mojem primeru je sistem Windows uporabljal vse izmed mojih treh rekurzivnih strežnikov. Ker poizvedovanje poteka po standardih, ni bilo nobene težave ob komunikaciji odjemalca Windows s strežnikoma Bind in Unbound. V politiki razreševanja DNS sistema Windows ne nastavimo, kateri strežnik DNS bomo uporabljali za validiranje naslovov. Uporablja se strežnik, ki ga trenutno uporablja sistem. Če strežnik, ki ga uporabljamo, ne podpira DNSSEC, bo sistem uspešno razrešil vse naslove. Če strežnik podpira DNSSEC, odjemalec vedno zahteva zastavico AD. Če te zastavice ni, odjemalec naslova ne bo razrešil. Uporabnik ne bo posebej obveščen, da je prišlo do napake pri validaciji DNSSEC, ampak bo samo dobil informacijo, da naslova ni bilo mogoče razrešiti (napaka, ki nam jo sporoči program Firefox, je prikazana na Sl. 26).
Ko sem vklopil validacijo DNSSEC za cel internet, je internet postal dokaj neuporaben. Tako ob brskanju praktično nisem mogel priti na nobeno stran (lahko sem npr. prišel na ltfe-sphere.org ali pa dnssec-tools.org). Validacijo je zatorej smiselno vklopiti zgolj za domene, ki že uporabljajo DNSSEC (npr. domene naše organizacije, če naša organizacija podpira DNSSEC). Delegiranih poddomen, ki niso podpisane, sistem ne razreši. Pri razreševanju se torej sistem striktno zanaša na zastavico AD v odgovoru strežnika DNS.



Sl. 26: Ob uporabi validiranja DNSSEC sistema Windows za cel internet je Google nedostopen

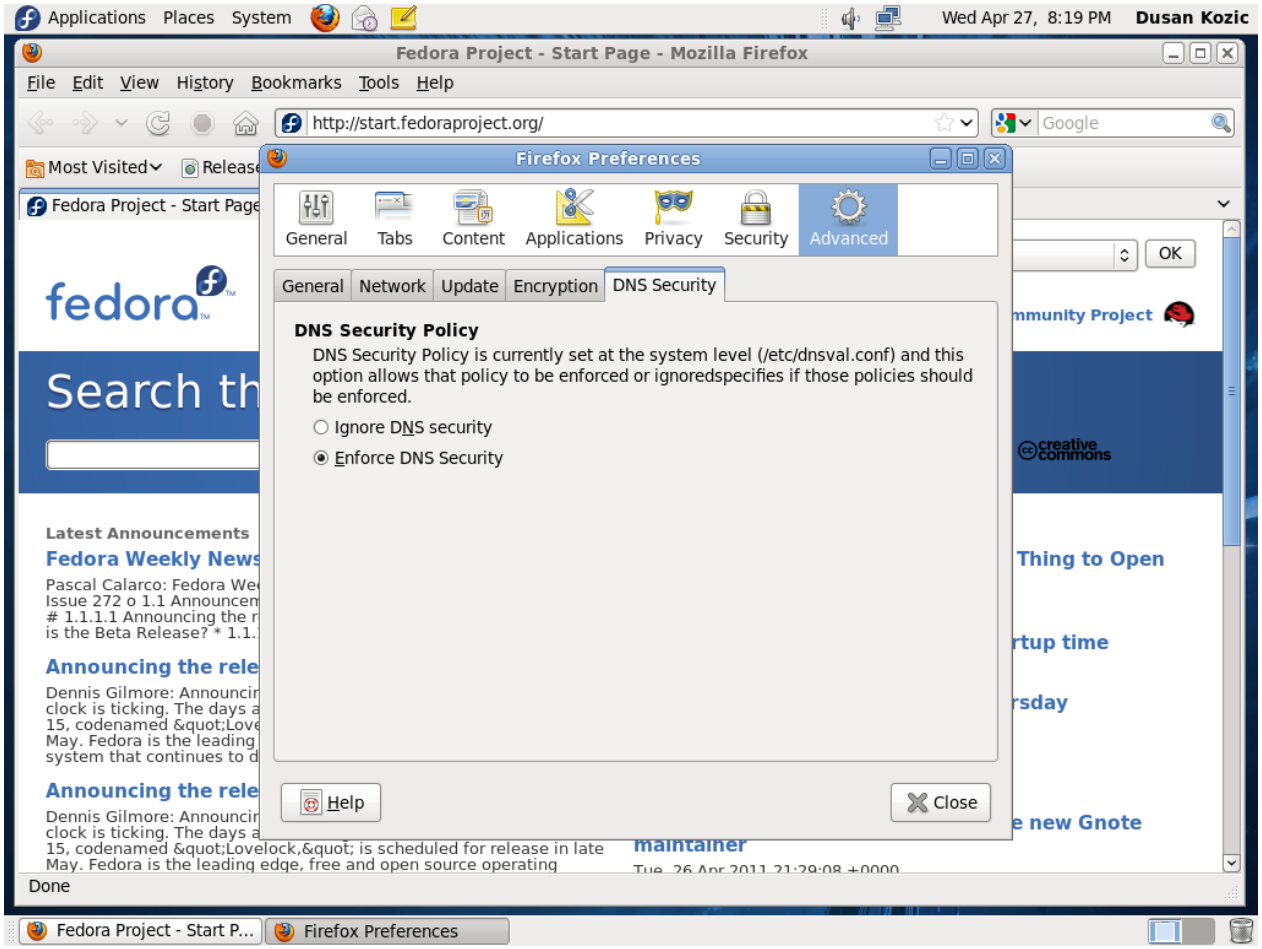
Dobra stran rešitve, ki nam jo prinaša sistem Windows, je, da gre za sistemsko politiko razreševanja naslovov DNS. Na enem mestu tako nastavimo politiko razreševanja naslovov za celoten sistem, ki nadalje vpliva na obnašanje vseh aplikacij, ki v okolju Windows uporabljajo DNS. Sami aplikaciji sploh ni treba vedeti, da obstaja DNSSEC, kar je trenutno zelo praktično, saj je DNSSEC še v začetni fazi. Za uporabnika bi bilo zaželeno, da bi ga sistem obvestil, da je prišlo do napake pri validiranju DNSSEC. Zaželeno bi tudi bilo, da bi se dalo nastaviti izjeme v politiki, ki bi omogočile uspešno razreševanje naslovov delegiranih poddomen, ki ne uporabljajo DNSSEC.

4.3.2 Odjemalci v okolju Linux

Linux v osnovi nima vgrajene podpore DNSSEC. Obstaja pa čedalje več aplikacij, ki podpirajo DNSSEC. Eden največjih projektov DNSSEC v okolju Linux je DNSSEC Tools. Prinaša nam več različnih orodij in aplikacij:

- orodja za administracijo datotek s conami: vključuje orodja za podpisovanje con, zamenjavo ključev,
- orodja za razhroščevanje: orodja za analizo obstoječih con, za grafični prikaz relacij med ključi in podpisi, za grafični prikaz poizvedovanja DNSSEC,
- sistemske knjižnice, ki aplikacijam omogočajo podporo DNSSEC,
- module za programski jezik Perl,
- spremembe in dodatke obstoječim aplikacijam, ki jim omogočijo uporabo DNSSEC.

V okviru projekta DNSSEC Tools obstaja tudi ponovno prevedena različica priljubljenega brskalnika Firefox, ki podpira DNSSEC. Spremenjeni Firefox se zanaša na knjižnico dnsval, ki je namenjena validiranju DNSSEC. Politiko validiranja sem nastavil v konfiguracijski datoteki knjižnice, v brskalniku pa sem samo izbral, ali bom uporabljal DNSSEC-validiranje ali ne. Knjižnica za validiranje poizvedeuje preko strežnika DNS, ki ga uporablja sistem. Razlika med knjižnico dnsval in ostalimi odjemalci DNSSEC je, da knjižnica sama opravlja validiranje (ne zanaša se na rekurzivni strežnik DNS). [47]



Sl. 27: Vklop validiranja DNSSEC v brskalniku Firefox v Fedora Core

V okviru projekta so modificirane tudi različne druge aplikacije, kot so Thunderbird, wget pa tudi strežniški programi: OpenSSH, Sendmail, Postfix. Uporabnik lahko pri uporabi oddaljenega dostopa do računalnika preko protokola SSH s pomočjo protokola DNSSEC verificira javni ključ, s katerim se mu je predstavil strežnik. Izvleček javnega ključa, ki se

nahaja v sistemu DNS, je šele ob uporabi DNSSEC vreden zaupanja. Modificirane aplikacije s podporo DNSSEC niso vključene v klasične distribucije sistema Linux, treba jih je namestiti naknadno. Le-to je najlažje v okolju Linux RedHat in njegovih izpeljankah (Fedora Core, CentOS), za katere je moč dobiti največ že prevedenih paketov RPM.

Podpora DNSSEC v sistem Linux ni vgrajena tako globoko kakor v sistem Windows. Vendar knjižnica dnsval v sistemu Linux zmore validirati DNSSEC neodvisno od rekurzivnega strežnika. Poleg tega je prevedenih že mnogo aplikacij za Linux, ki na različne načine podpirajo DNSSEC. Linux je po razširjenosti in podprtosti DNSSEC pred sistemom Windows, kar je tudi pričakovano, glede na to, da je Linux operacijski sistem, ki ga uporablja mnogo razvijalcev, DNSSEC pa je še v začetni fazi implementacije in testiranja.

4.3.3 Odjemalci v okolju Mac OS X

Orodja DNSSEC Tools delujejo tudi v sistemu Mac OS X. Treba jih je prevesti ročno, saj niso na voljo v obliki namestitvenega paketa. Aplikacij za Mac OS X, ki bi podpirale DNSSEC oz. uporabljale knjižnico dnsval, ki je vključena v ta paket, pa ni (strežnika Bind in Unbound tečeta na sistemu Mac OS, vendar je tukaj govora o aplikacijah za končne uporabnike). Ker gre za izpeljanko sistema Unix, so sicer na voljo različni programi za poizvedovanje v ukazni lupini, kakor je npr. dig, vendar so ta orodja namenjena bolj sistemskim administratorjem za preverjanje pravilnega delovanja DNSSEC.

4.3.4 Primerjava

Tab. 12 prikazuje povzetek primerjave podpore DNSSEC med odjemalci na treh najbolj priljubljenih operacijskih sistemih.

Tab. 12: Primerjava DNSSEC na odjemalcih

	Windows	Linux	Mac OS
Sistem v osnovi podpira DNSSSEC	da	ne	ne
Možna uporaba DNSSEC	da	da	da
Zmore sam validirati DNSSEC	ne	da knjižnica dnsval	da knjižnica dnsval
Aplikacije, ki podpirajo DNSSEC	jih ni; obnašanje se kontrolira s pomočjo politike sistema	srednje veliko	malo
Orodja za poizvedovanje in preizkus delovanja DNSSEC	malo	veliko	srednje veliko

Všeč mi je, da ima sistem Windows že v osnovi vgrajeno podporo DNSSEC. Kaj takega bi si želel tudi pri ostalih operacijskih sistemih. V okolju Windows sicer pogrešam orodja za preverjanje delovanja DNSSEC, je pa žal res, da takih sistemskih orodij za okolje Windows na splošno primanjkuje (ukazna lupina v okoljih Unix/Linux ima svojo moč). Projekt DNSSEC Tools za sistema Linux in Mac OS X je zelo v redu, saj uporabniku omogoča, da sam preveri podatke, ki jih je dobil od strežnika DNS, in tako se mu ni treba zanašati na eno nastavljeno zastavico. Na splošno bi si želel več aplikacij, ki podpirajo DNSSEC.

4.3.5 Odjemalci, neodvisni od okolja

Za brskalnik Firefox obstaja vtičnik DNSSEC Validator. Gre za preprost vtičnik, ki uporabniku na prijazen način sporoča, ali je stran, po kateri brska, ustrezno zavarovana z DNSSEC. Ob naslovni vrstici se pokaže ikona s ključem, ki kaže stanje domene, na kateri se nahaja. Mogoča so sledeča stanja:

 Domena je ustrezno zavarovana.

 Domena je zavarovana, vendar rekurzivni strežnik ne more preveriti veljavnosti podpisa (ponavadi, ko imamo podpisano domeno, vendar manjka zapis DS na domeni en nivo višje).

 Domena ni zavarovana z DNSSEC (pomeni, da domena še ne uporablja DNSSEC).

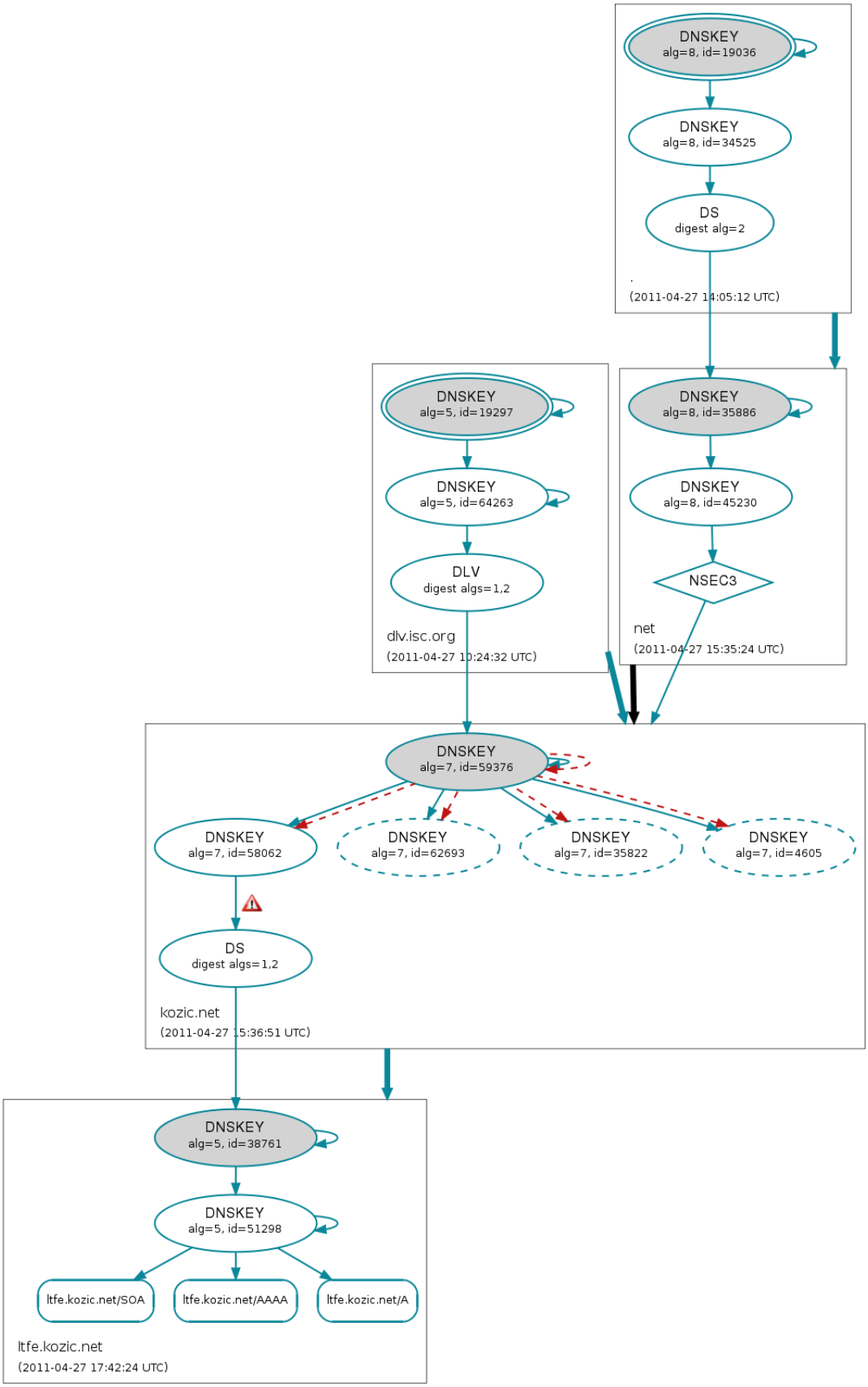
 Domena je zavarovana z DNSSEC, vendar podpis ni veljaven. Ta napaka bi naj pomenila, da ne dostopamo do prave domene.

Rdeč ključ uporabniku z gotovostjo pove, da ne dostopa do prave domene. Vendar če vemo, da je domena ustrezno podpisana, moramo biti vedno pozorni na zelen ključ. Vsa ostala stanja lahko pomenijo, da smo tarča napadalca.

DNSSEC Validator ima dokaj malo nastavitev. Kvečjemu sem lahko nastavil samo, kateri strežnik DNS bom uporabljal za poizvedovanje in validiranje. DNSSEC Validator je torej še eno izmed orodij, ki se pri validiranju DNSSEC zanašajo na rekurzivni strežnik. Poleg tega uporabniku ne ponuja nobenih možnosti za zavarovanje komunikacije med njegovim računalnikom in rekurzivnim strežnikom. Kljub svoji preprostosti je zasnovan na način, ki uporabniku na pregleden način daje nadzor nad tem, kaj se z domeno, do katere dostopa, dejansko dogaja. [48]

4.3.6 Orodja za testiranje DNSSEC

Spletno orodje DNSViz administratorju vizualno prikaže verigo zaupanja DNSSEC za posamezno domeno. Na ta način lahko administrator zazna in odpravi napake v DNSSEC. Uporaba spletne strani je priporočena, ko postavljamo DNSSEC. Podjetje Verisign ponuja orodje DNSSEC Debugger, ki podobne informacije, kakor jih DNSViz izriše, predstavi tekstovno.



Sl. 28: Veriga zaupanja za domeno ltfe.kozic.net. Vir: dnsviz.net

Obstajajo tudi strani, ki preverjajo, ali uporabljamo DNSSEC. Primer ene izmed njih je na Sl. 29.

Sl. 29: Stran, ki uporabniku prikaže, ali uporablja DNSSEC. Vir: http://xs.forfun.net/dnssec

4.3.6.1.1 Sistemska orodja v okolju Linux

Linux je zaradi svoje ukazne lupine (angl. shell), ki vključuje kopico mogočnih orodij, zelo uporaben sistem za poizvedovanje po DNS.

Orodje, ki sem ga med diplomsko nalogo in ga nasploh največ uporabljam za poizvedovanje DNS, je dig. Alternativa orodju dig je orodje drill, ki je bilo narejeno zlasti za DNSSEC. Prihaja v paketu orodij imenovanih ldns, ki bi naj bila namenjena poenostavitvi programiranja DNS. Poleg orodij za poizvedovanje DNS(SEC) vsebuje tudi orodja za generiranje ključev, podpisov DNSSEC. Razvija ga NLnet Labs. [44]

Primeri uporabe orodij:

dig: uporablja se za poizvedovanje po DNS

```
administrator@dk-dnssec-us1004:~$ dig +dnssec www2.sah-drustvo-ms.si
@212.101.143.6

; <<>> DiG 9.7.1-P2 <<>> +dnssec www2.sah-drustvo-ms.si @212.101.143.6
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45653
;; flags: qr rd ra ad[7]; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4000
;; QUESTION SECTION:
;www2.sah-drustvo-ms.si.                IN      A

;; ANSWER SECTION:
www2.sah-drustvo-ms.si. 10800   IN      A       193.77.168.17
www2.sah-drustvo-ms.si. 10800   IN      RRSIG   A 5 3 10800 20110513124841
20110412114841 11293 sah-drustvo-ms.si.
Jdd+g2mNfFOeOtKXaVd5+6ehMTvpN0dlgRyJ0MiQ8FLeYRjJxJp3QvbY
untwyI8poFimQ1bJmvzYOILQX65FWBP5d4MkvLydAVXp+QK6+1yFwLFm
my2mGQm7U7dTCJ7O+CrnIJ+mTJczFML1uwo3zv3nUr3GMwivfEve33YE Bik=

;; Query time: 8 msec
;; SERVER: 212.101.143.6#53(212.101.143.6)
;; WHEN: Wed Apr 13 14:56:48 2011
;; MSG SIZE  rcvd: 244

administrator@dk-dnssec-us1004:~$
```

ldns-key2ds: iz zapisa DNSKEY nam izračuna zapis DS

```
root@Kathy:~# ldns-key2ds -n -2 key-sah
sah-drustvo-ms.si.      3600    IN      DS      56789 5 2
401a35d7ebee6e7364dd601a5a0d1efc9db05c3c0c74672c69b49ef4f3f6a320 ; xibac-
pytit-lipev-verul-funet-temyc-pokyb-tylyz-soler-bolof-safil-gonud-sipor-
guloz-gesaz-kemod-buxex
root@Kathy:~#
```

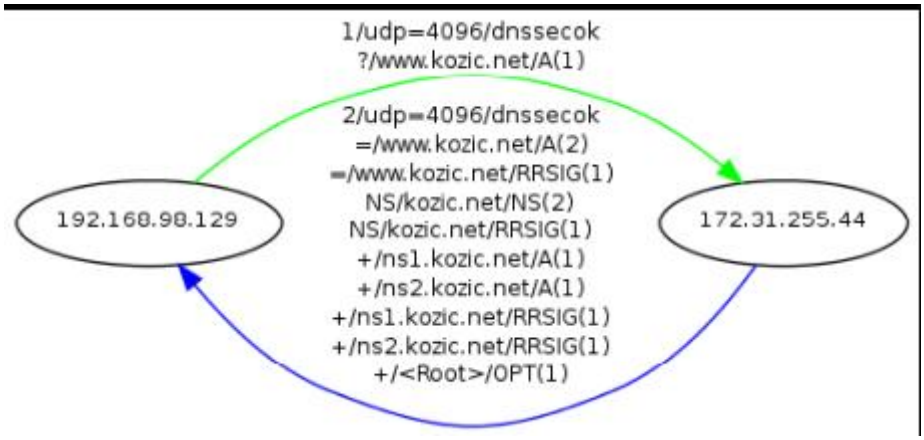
[7] Če na tem mestu piše AD, je bila validacija DNSSEC uspešna, če AD manjka, je bila validacija neuspešna.

```
ldns-walk: sprehodi se po coni, ki uporablja NSEC
root@Kathy:~# ldns-walk sah-drustvo-ms.si
sah-drustvo-ms.si.      sah-drustvo-ms.si. A NS SOA MX RRSIG NSEC DNSKEY
ezekiel.sah-drustvo-ms.si. A RRSIG NSEC
ftp.sah-drustvo-ms.si. CNAME RRSIG NSEC
karpov.sah-drustvo-ms.si. CNAME RRSIG NSEC
localhost.sah-drustvo-ms.si. A RRSIG NSEC
mail.sah-drustvo-ms.si. CNAME RRSIG NSEC
pop.sah-drustvo-ms.si. CNAME RRSIG NSEC
www.sah-drustvo-ms.si. A RRSIG NSEC
www2.sah-drustvo-ms.si. A RRSIG NSEC
www3.sah-drustvo-ms.si. A RRSIG NSEC
www4.sah-drustvo-ms.si. A RRSIG NSEC
```

ldns-nsec3-hash: izračuna nam izvleček NSEC3-imena, po katerem poizvedujemo

```
root@Kathy:~# ldns-nsec3-hash -a 1 -t 5 -s E8E3AEFA91A35DA6 a.kozic.net
mfr611lq82h5ico6rfe5630ukbvnrmad.
root@Kathy:~#
```

Paket DNSSEC Tools nam prinaša orodja, ki na podlagi zajemanja prometa grafično izrišejo potek poizvedovanja DNSSEC. Sl. 30 prikazuje izris, ki ga naredi orodje dnspktflow.



Sl. 30: Primer izrisa diagrama z orodjem dnspktflow ob poizvedovanju na rekurzivnem strežniku

5 SKLEP

Protokol DNS nima vgrajenih varnostnih mehanizmov, zato obstajajo napadi nanj, ki jih je preprosto izvesti in težko preprečiti. Zastrupljanje medpomnilnika je eden najnevarnejših napadov na protokol DNS. Leta 2008 je Dan Kaminsky iznašel preprost in učinkovit način zastrupljanja medpomnilnika DNS. Takrat je postalo popolnoma jasno, da je protokol DNS v obstoječi obliki neprimeren za nadaljnjo uporabo.

DNSSEC je modifikacija obstoječega protokola DNS. Sistemu DNS dodaja mehanizme za zagotavljanje integritete in zagotavljanje avtentikacije ter tako zagotovi, da nas napadalec s ponarejanjem sporočil DNS ne more preusmeriti na lažen računalnik. Na ta način odpravlja večino napadov na star in luknjast protokol DNS. Zastrupiti medpomnilnik DNS je praktično nemogoče.

Korenska domena že uporablja DNSSEC, prav tako ga uporablja večina vrhnjih domen. Razvija se čedalje več programske opreme, ki podpira DNSSEC, na voljo so tudi komercialne rešitve. Kako hitro in če bo DNSSEC zaživel pri končnih uporabnikih, pa je trenutno dokaj težko napovedati.

V testnem okolju sem vzpostavil sistem DNSSEC. Preizkusil sem delovanje DNSSEC na avtoritativnih strežnikih DNS Bind in Windows Server DNS, rekurzivnih strežnikih DNS Bind, Windows Server DNS, Unbound ter odjemalcih.

Ugotovil sem, da se za DNSSEC na avtoritativnih strežnikih DNS najbolje obnese kombinacija strežnika Bind na operacijskem sistemu Linux in uporaba tehnologije OpenDNSSEC za podpisovanje domen DNS. Za validiranje DNSSEC na rekurzivnih strežnikih sta primerna tako strežnik Bind kakor Unbound, Windows Server DNS pa ni najbolj primeren, saj ne podpira novejših algoritmov DNSSEC, s katerimi je podpisanih večina domen, vključno s korensko.

Kar se tiče odjemalcev, je DNSSEC podprt v operacijskem sistemu Windows od različice Vista naprej. Za Linux obstaja nabor orodij DNSSEC Tools, ki nam prinesejo možnost validiranja DNSSEC in podporo DNSSEC v različnih aplikacijah. DNSSEC Tools je mogoče uporabljati tudi v okolju Mac OS X.

Kar se tiče nadaljnjega dela s področja DNSSEC, je na voljo še veliko prostora za razvoj novih, preprostih orodij za vzdrževanje ključev in samodejno podpisovanje domen. Obstoječa orodja imajo zaradi svoje mladosti veliko število hroščev, zato velja nameniti poseben poudarek tudi razvoju orodij, ki bodo aktivno preverjala naše podpisane domene in nas tudi obveščala o napakah. Ogromno pa je treba še postoriti glede uporabe DNSSEC na odjemalcih, zato nam nadaljnjih izzivov zagotovo ne bo zmanjkalo.

6 LITERATURA

[1]. R. Aitchison, *Pro DNS and Bind*, Apress: Berkeley CA, 2005, pogl. 11.

[2]. P. Albitz, C. Liu, *DNS and Bind*, Sebastopol: O'Reilly Media, 2006, pogl. 11.

[3]. (2010) J. Abley, et al, DNSSEC Root Zone High Level Technical Architecture, Root DNSSEC Design Team, Junij 2010. Dostopno na: http://www.root-dnssec.org/wp-content/uploads/2010/06/draft-icann-dnssec-arch-v1dot4.pdf (14.2.2011)

[4]. (2005) R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, DNS Security Introduction and Requirements, RFC 4033, Marec 2005. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=4033 (17.1.2011)

[5]. (2005) R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, Resource Records for DNS Security Extensions, RFC 4034, Marec 2005. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=4034 (17.1.2011)

[6]. (2005) R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, Protocol Modifications for DNSSEC Security Extensions, RFC 4035, Marec 2005. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=4035 (18.1.2011)

[7]. (2008) R. Arends, D. Blacka, B. Laurie, G. Sisson, DNS Security (DNSSEC) Hashed Authenticated Denial of Existence, RFC 5155, Marec 2008. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=5155 (20.1.2011)

[8]. (2007) S. Ariyapperuma, C. J. Mitchell, Security vulnerabilities in DNS and DNSSEC, 2007. Dostopno na: http://portal.acm.org/citation.cfm?id=1250514 (3.2.2011)

[9]. (2010) J. Bau, J. Mitchell, A Security Evaluation of DNSSEC with NSEC3, Marec 2010. Dostopno na: http://eprint.iacr.org/2010/115.pdf (3.2.2011)

[10]. (2005) S. Bellovin, Using the Domain Name System for System Break-ins, Junij 1995. Dostopno na: http://www.usenix.org/publications/library/proceedings/security95/full_papers/bellovin.pdf (7.2.2011)

[11]. I. Brajović, U. Bajželj, J. Guna, A. Kos, J. Sterle, Varnost v telekomunikacijah, Fakulteta za elektrotehniko, Februar 2008.

[12]. (2008) M. Ciglarič, Zaščita in kriptiranje, Fakulteta za računalništvo in informatiko, 2008. Dostopno na: http://ucilnica0910.fri.uni-lj.si/mod/resource/view.php?id=2750 (8.2.2011)

[13]. (1976) W. Diffie, M. E. Hellman, New Directions in Cryptography, November 1976. Dostopno na: www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf (7.2.2011)

[14]. (2003) C. Florent, Security Issues with DNS, SANS Institute, 2003. Dostopno na: http://www.sans.org/reading_room/whitepapers/dns/security-issues-dns_1069 (7.1.2011)

[15]. (2010) F. Kammüller, Lecture 2: Cryptography, Technische Universität Berlin, April 2010. Dostopno na: http://user.cs.tu-berlin.de/~flokam/Seminar10/2_Cryptography.pdf (8. 2. 2011)

[16]. (2006) W. Hardaker, Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs), RFC 4509, Maj 2006. Dostopno na: http://www.faqs.org/rfcs/rfc4509.html (8.2.2011)

[17]. (2009) J. Jansen, Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC, RFC 5702, Oktober 2009. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=5702 (9.2.2011)

[18]. (2006) O. Kolkman, R. Gieben, DNSSEC Operational Practices, RFC 4641, September 2006. Dostopna na: http://rfc-ref.org/RFC-TEXTS/4641/ (8.2.2011)

[19]. (2010) F. Ljunggren, et al., DNSSEC Practice Statement for the Root Zone KSK Operator, Root DNSSEC Design Team, Maj 2010. Dostopno na: http://www.root-dnssec.org/wp-content/uploads/2010/06/icann-dps-00.txt (14.2.2011)

[20]. (1987) P. Mockapetris, DOMAIN NAMES – CONCEPTS AND FACILITIES, RFC 1034, November 1987. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=1034 (5.1.2011)

[21]. (1987) P. Mockapetris, DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION, RFC 1035, November 1987. Dostopno na: http://www.rfc-archive.org/getrfc.php?rfc=1035 (5.1.2011)

[22]. (2010) T. Okubo, et al., DNSSEC Practice Statement for the Root Zone ZSK Operator, Root DNSSEC Design Team, Maj 2010. Dostopno na: http://www.root-dnssec.org/wp-content/uploads/2010/06/vrsn-dps-00.txt (14.2.2011)

[23]. (2010) ARNES, DNSSEC napreduje počasi, November 2010. Dostopno na: http://www.register.si/obvestila/obvestilo/article/-a3fb26c58b.html (14.2.2011)

[24]. (2010) CircleID, Leading Registrars Supporting DNSSEC, Julij 2010. Dostopno na: http://www.circleid.com/posts/20100720_leading_registrars_supporting_dnssec/ (14.2.2011)

[25]. (2010) Cisco Networking Academy, CCNA Security 1.0. Dostopno na: http://cisco.netacad.net (7.2.2011)

[26]. (2009) Community DNS, A paper on DNSSEC – NSEC3 with Opt-Out, September 2009. Dostopno na: http://www.communitydns.eu/DNSSEC.pdf (15.2.2011)

[27]. (2010) DNSSEC Deployment Iniciative, Februar 2011. Dostopno na: https://www.dnssec-deployment.org/ (14.2.2011)

[28]. (2010) Google Public DNS, Frequently Asked Questions, 2011. Dostopno na: http://code.google.com/intl/sl/speed/public-dns/faq.html (14.2.2011)

[29]. (2011) IANA, Verisign, Root DNSSEC, Februar 2011. Dostopno na: http://www.root-dnssec.org/ (13.2.2011)

[30]. (2011) S. Langerholm, TLD with DNSSEC, Februar 2011. Dostopno na: http://www.tldwithdnssec.se/ (14.2.2011)

[31]. (2010) C. D. Marsan, 80% of government Web sites miss DNS Security deadline, Network World, Januar 2010. Dostopno na: http://www.networkworld.com/news/2010/012010-dns-security-deadline-missed.html (14.2.2011)

[32]. (2010) Niobos, DNSSEC – the NSEC and NSEC3 record, Long-term Memory Blog, Januar 2010. Dostopno na: http://blog.dest-unreach.be/2010/01/20/dnssec-the-nsec-and-nsec3-record (9.2.2011)

[33]. (2005) NLnet Labs, A short history of DNSSEC, November 2010. Dostopno na: http://www.nlnetlabs.nl/projects/dnssec/history.html (7.2.2011)

[34]. (2011) DNSSEC Security Extensions, Wikipedia, Februar 2011. Dostopno na: http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions (14.2.2011)

[35]. (2011) Public-key cryptography, Wikipedia, Februar 2011. Dostopno na: http://en.wikipedia.org/wiki/Public-key_cryptography (8.2.2011)

[36]. (2011) ORG Registrars, Februar 2011. Dostopno na: http://www.pir.org/get/registrars?order=field_dnssec_value&sort=desc (14.2.2011)

[37]. P. Vixie, »Preventing Child Neglect in DNSSECbis Using Lookaside Validation (DLV),« *IEICE Transactions on Communications*, št. E88-B, zv. 4, str. 1326-1330, 2005.

[38]. (2007) S. Weiler, DNSSEC Lookaside Validation (DLV), RFC 5074, November 2007. Dostopno na: http://www.faqs.org/rfcs/rfc5074.html (14.3.2011)

[39]. (2010) NIST, The Secure Naming Infrastructure Pilot (SNIP), Marec 2011. Dostopno na: http://www.dnsops.gov (14.3.2011)

[40]. (2010) Microsoft, DNSSEC Deployment Guide. Dostopno na: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=7a005a14-f740-4689-8c43-9952b5c3d36f&displaylang=en (25.4.2011).

[41]. (2007) M. StJohns, Automated Updates of DNS Security (DNSSEC) Trust Anchors, RFC 5011, September 2007. Dostopno na: http://tools.ietf.org/html/rfc5011 (30.4.2011)

[42]. (2011) Internet Systems Consortium (ISC), Bind Documentation, Marec 2011. Dostopno na: http://www.isc.org/software/bind/documentation (15.4.2011)

[43]. (2011) NLnet Labs, Unbound Documentation, Februar 2011. Dostopno na: http://unbound.net/documentation/index.html (17.4.2011)

[44]. (2011) NLnet Labs, ldns, Marec 2011. Dostopno na: http://nlnetlabs.nl/projects/ldns/ (4.4.2011)

[45]. (2011) OpenDNSSEC, OpenDNSSEC Documentation, Januar 2011. Dostopno na: http://www.opendnssec.org/documentation/ (18.4.2011)

[46]. (2010) C. Strotmann, Men & Mice, Commercial DNSSEC Solutions, 2010. Dostopno na: http://distance.ktu.lt/terena/dnssec-workshop/commercial-dnssec-solutions-carsten-strotmann-men-mice (1.4.2011)

[47]. (2010) DNSSEC-Tools, 2010. Dostopno na: http://www.dnssec-tools.org/ (3.4.2011)

[48]. (2011) CZ Domain Registry, DNSSEC Validator, 2011. Dostopno na: http://www.dnssec-validator.cz/ (20.3.2011)

[49]. (2008) S. Friedl, An Illustrated Guide to the Kaminsky DNS Vulnerability, 2008. Dostopno na: http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html (2.5.2011)

[50]. (2008) US-CERT, Vulnerability Note VU#800113, 2008. Dostopno na: http://www.kb.cert.org/vuls/id/800113 (15.3.2011)

[51]. (2011) CZ Domain Registry, Maj 2011. Dostopno na: http://www.nic.cz/ (3.5.2011)

[52]. (2007) Technical Info, The Pharming Guide (part 2), 2007. Dostopno na: http://www.technicalinfo.net/papers/Pharming2.html (22.5.2011)

[53]. (2010) Internet Society, DNSSEC – A Review, Junij 2010. Dostopno na: http://www.potaroo.net/ispcol/2010-06/dnssec.html (23.5.2011)

[54]. Cisco Networkers 2008, The DNSSEC Standard, 2008.

[55]. ICANN, Internet Corporation for Assigned Names and Numbers. Dostopno na: http://www.icann.org/ (9.6.2011)

[56]. IANA, Internet Assigned Numbers Authority. Dostopno na: http://www.iana.org/ (9.6.2011)

[57]. Verisign. Dostopno na: http://www.verisign.com/ (9.6.2011)

[58]. NTIA, National Telecommunications and Information Administration. Dostopno na: http://www.ntia.doc.gov/ (9.6.2011)

[59]. United States Department of Commerce. Dostopno na: http://www.commerce.gov/ (9.6.2011)

[60]. IETF, Internet Engineering Task Force. Dostopno na: http://www.ietf.org/ (9.6.2011)

[61]. CENTR, Council of European National Top Level Domain Registries. Dostopno na: https://www.centr.org/main/index.html (9.6.2011)

[62]. Arnes, Akademska in raziskovalna mreža Slovenije. Dostopno na: http://www.arnes.si/ (9.6.2011)

[63]. APEK, Agencija za pošto in elektronske komunikacije Republike Slovenije. Dostopno na: http://www.apek.si/ (9.6.2011)

[64]. T-2. Dostopno na: http://www.t-2.net/ (9.6.2011)

PRILOGE

1 Primer podpisane cone

```
ltfe-sphere.org.        3600  IN SOA      throne.ltfe-sphere.org.
hostmaster.kozic.net. (
                              2011040523   ; serial number
                              900          ; refresh
                              600          ; retry
                              86400        ; expire
                              3600       ) ; default TTL
                        3600  RRSIG SOA 5 2 3600 20110526140539 (
                              20110426140539 55812 ltfe-sphere.org.
                              RFhlmeAWVsujx04VsDNtTS4SQrqp8p34XmEI
                              ZxE8L5NqY815YvqXQdFbREOifnrG2s6C6ZKz
                              leWJaKzi6JzWB2NRgAnQGPiGsxQUOEKOJiL+
                              wlZte8AUTtEQ9SMaHfGSALz0OopTSLTYR03B
                              XZvXwrSdQcybZqkofYEmr8/VZyw=
                              )
                        3600  A     212.101.143.177
                        3600  RRSIG A 5 2 3600 20110526140539 (
                              20110426140539 55812 ltfe-sphere.org.
                              GWWiOm5aiDysixFPQEsxwGQoIF0mT5qd3vWz
                              SoxENHe7S1wMOkOTwd95TUP7qiRPrJ5u9Ufp
                              ECjR69+FXGf/+ym7PO3AcvBI78/A3/FHflCl
                              z9P31K+dAEOCfDVpFCND1mE8zhSd8kkMFKwY
                              2PoIBCqboZDvgwUKKUpcV7Oi6aE=
                              )
                        3600  NS    throne.ltfe-sphere.org.
                        3600  RRSIG NS 5 2 3600 20110526140539 (
                              20110426140539 55812 ltfe-sphere.org.
                              JEpjsHUck2Q3eJS7rfXSOA0VtcWpE5E6RPZW
                              APNQB8k1GHBxbLA+k6BE01d34JvV7p9in4ND
                              GX1xhc1MYO1bxAIJ3wj16mz0kWQg5Q1aWdLL
                              k0oD31wqcFpplAzajr95FVwynkY9wxDYM8dv
                              KwbXjh4ltpKk3pz5HbogAS0MOTY=
                              )
                        3600  MX    10    mail.kozic.net.
                        3600  RRSIG MX 5 2 3600 20110526140539 (
                              20110426140539 55812 ltfe-sphere.org.
                              e2sijKjL4j854gWAxerEdV8Tcf6bbMyIYyPx
                              EeOFYcGEUTikuoNEB9N7sHD1c+m0ft1dY6sy
                              XIK9WmByoY6a/5ZDNMX1oF8/5NOkxkcg2lka
                              OWy8jB5jsFJuaFWV6CAoiFUrkP8R+F6Ev4ah
                              t+PRMxnPAWW33g8dZNcUksA58hs=
                              )
                        3600  AAAA  2a00:1368:1000:20::177
                        3600  RRSIG AAAA 5 2 3600 20110526140539 (
                              20110426140539 55812 ltfe-sphere.org.
                              IQyhwSmXaNQrk+ctwd9aEGopxvN62oVy4wqD
                              QCJPkR6oNgbzFdLUJEFyMgRadtbmlCl4QIFL
                              tTiWcdA3bthqvE7RHXWOpI+6RHN8qpO0H9sg
                              a/Nbnhs2Cu9coQnkB/781fv+DoYvg/gjujt9
                              EFVOWCGeFsNoBnsPhIx+2Le1Bwk=
```

```
                                )
                        3600  NSEC  lordship.ltfe-sphere.org. A NS SOA MX
AAAA RRSIG NSEC DNSKEY
                        3600  RRSIG NSEC 5 2 3600 20110526140539 (
                                20110426140539 55812 ltfe-sphere.org.
                                0hUzBg4fLU+3/XGtKpZOSPq9OOpsVH2dWZQ2
                                +TXLz57z/Lu37fpceQWOGyK3bCHtBID2vr1K
                                TvWrD7oU3E3kWDyeVT1XT/Z7ZT/PBmEWFRqC
                                AucLJGJH+RJo1Fqy2gd0dNP3nn6Cc2Nh8T1d
                                xtcHlkGcPGW0J8jC1MEjNx8Q75w=
                                )
                        3600  DNSKEY      257 3 5 (
                                AwEAAaCurcvVVgLLAGyOae0VRI4RgS/UztGg
                                SwV18CqZ2Z1ndj8y0aQy6sxGdsOSin42LESh
                                OqMPOgLMkRZl/hHIQDRVOtmHwS4rjvHjR9DA
                                9ju2MUH5TgLm9pdHiO66Vgh+njtEPwEzgvN0
                                xm216vbXdJ8M7nvyveN4Xpv38AUMzoC2I/ie
                                cJyMgcJGOgn3iQvssNPZGjuPKjbXAhspMzuP
                                qF0WE13A3UhzvIt5rCyGQ6lIDdLugeymuzFS
                                J7rss97yIV1dItiFYopb8V6qzuturFmXTwkT
                                xhZrN4p72JfSw/c03ynC7zj03o2NIeiKNxuP
                                M7po89NRyzObiDfHnDC9byE=
                                ) ; key tag = 21491
                        3600  DNSKEY      256 3 5 (
                                AwEAAd+//+GX4sRsJFQm0vLKXugHorWMVeym
                                XPpqyMmTpeFGBOl1J2LPQPM5NkYZE0Kud8F5
                                YgYEPWkLse3ulahHSZFCrrvWZE9YVdnTVTC9
                                3DM5//tIIxKgOuWZGp9q5DmPO1TaXKDVpDLg
                                TWIFEmIlqIU0V1eYR3IjsIl3b8ykMr4N
                                ) ; key tag = 55812
                        3600  RRSIG DNSKEY 5 2 3600 20120520140539 (
                                20110426140539 21491 ltfe-sphere.org.
                                gd6ebrB17EB/3cEwAEPm3jWkQ8iFjPKxzoqs
                                nLyfqYmI7b0/+eq84K8k8nX3l8Uze7ZyLd0X
                                r3XO4h1xyc90uRJ0XK3syPBj7k7rAkXnS/Gy
                                wLVngKwh15IZjjtdDXuqNagWu8QpvVWp1lW9
                                1zyxEEIjGyqYwsXY/y60Tni1LWpnrb1nwse9
                                VzexBHNQ+VNDbt7OFC2CSLasI425dZNITqqC
                                4oeM75jiZIK4E9BcpxeVawaUSHJ8ILCdNqs9
                                +C51Bqk/hQtlnI1TSthHQEPppb1iWYdQ4ZCc
                                yXzas8REdsw1vE4eReMSumkRFtOuBYmqX8mA
                                7kpBKAGp/67PuFChlA==
                                )
                        3600  RRSIG DNSKEY 5 2 3600 20110526140539 (
                                20110426140539 55812 ltfe-sphere.org.
                                gezlnMhDlTfQZtDsHgjU9ghBuxY6RqxF1KAN
                                lvzqIvdCAb3R0LYiA8kfDzdydBU20LXXhSPH
                                GJ25uTXUchqpS1+NAuVubRUFlYv/7KbFr7DE
                                JA9EKtZShcUsHbxNwqJUi+jkXZtsZhUXUaSe
                                2oyBy8aA9IiYhWY0rwKf1Pe0f6g=
                                )
lordship.ltfe-sphere.org. 3600      IN A  212.101.143.6
                        3600  RRSIG A 5 3 3600 20110526140539 (
                                20110426140539 55812 ltfe-sphere.org.
                                dtOG7A92vKX+E6WxM8KTx/gAPsBMyHVgqziU
                                IVdPGv5pdsKkwje8eMSCZHL43CvRBSGB0e10
                                KGZRCNvLIdUZBNiWg6MRucDhARsu0wbGEvCY
                                bKYDW/tih/MmdsOThILZLbnJtVGmEPQbCD6r
                                qR4KQsJN3Sq7ZEgpypySfzQFJhg=
```

```
                                )
                        3600  AAAA  2a00:1368:1000:20::6
                        3600  RRSIG AAAA 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    EANSl1yeOZwu3u4ff2abe6P1VyWx+P2rWS9n
                                    xYK9znT8RXXRlrg2IB/XrymLlji57MjyG1xn
                                    Gyu4/2D3UTEGvGZx1M7TGCSeD0FGi8hI7OEs
                                    1jFr8nARywU/ZNxpwVz5W7ys3t3zX/WIPktA
                                    ckhqjZh0hZH0L9wY04DBmIwTXQg=
                                    )
                        3600  NSEC  second.ltfe-sphere.org. A AAAA RRSIG
NSEC
                        3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    elhuesj+H/6xOLnpifG1e+n3F93Vn7bw1iYe
                                    1B2iBgp5sQtucuLUCgOTy3nCJ2KAvPfigW7l
                                    SOl0RWWF12q4y8yyhUfSdQaXTA8zF9d9ga/p
                                    xbk3W9qKNVF5bTRj/mxScFe9JMvglIVtA13R
                                    iwYjP04xfgJYXc28pEGPSaHzFLM=
                                    )
second.ltfe-sphere.org. 3600  IN NS ns1.kozic.net.
                        3600  DS    38761 5 1 (
                                    C03E559F6FD891CA3EAF0325F83712DEEA0B
                                    8678 )
                        3600  DS    38761 5 2 (
                                    4D36CDAC1790AABD07BB0D02F4780F82F156
                                    367FB9A078912CC7B3481326B48D )
                        3600  RRSIG DS 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    YbomgabgVjJcqvrJkjUSwvh4yPOVdL57mcBT
                                    hHXCT1QNtL4enB45EnZ9H4mbHcmgrc7xEhql
                                    YP/4LzgKAGBYh6A0grSftvsoI3HCA9HVFg5h
                                    Ohlq5CGoJK1Jz10kABemdv3RiwBt8Fwlmipm
                                    youuwltY3jJ+Pr4lSuWiT1u44W0=
                                    )
                        3600  NSEC  stronghold1.ltfe-sphere.org. NS DS RRSIG
NSEC
                        3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    dbR0eDqDL9nGLhUmvBhWMqhj6XjlPph8hnFs
                                    deVqd7w+1rDoUOEJtNRePQqhfuOtBScohuc1
                                    lhRbcN6VfVu3IBXVuNHnNNh3yMwEdJEgeUgy
                                    S/DQuAgpQ5mgJor0RqstKwds0ZKANbxxmWmP
                                    dlyo+l4dp0kfAAgV+HD4293AhZE=
                                    )
ns1.kozic.net.          3600  IN A  93.103.130.109
stronghold1.ltfe-sphere.org. 3600   IN A  212.101.143.7
                        3600  RRSIG A 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    XtagcbDpDDurLBbdOJOmJIsNlE+iweFCSfAB
                                    8rOI317Z3EfSyplV5HH/w2i3C/uS3mCo41vI
                                    v/3hEDpeIkXLvAKIarPjUF+Tjxt2A8ERsr41
                                    3u7uUYghwiYr1v6MlCDtnS4IOmHrBoBfRwpJ
                                    pYHPPR9sa46a119ysXqPGh7P300=
                                    )
                        3600  AAAA  2a00:1368:1000:20::7
                        3600  RRSIG AAAA 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    WfCMRtOKMlSQXx2HYDn8kD0/K5QSkZvVeeqJ
```

```
                                    Q6k2kJeXcVi0kMBcC9JuBPB9vs35u3L2AXDS
                                    h447B1RlKMd/vW7D8lhaY9zu+ldE5hVF7cWP
                                    hzLy71t2jDt0QjpiRmZx+//XTrQhWu/43UCJ
                                    CVMAzNevmytMUGFGi0WcbCHfuJU=
                                    )
                          3600  NSEC  stronghold2.ltfe-sphere.org. A AAAA
RRSIG NSEC
                          3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    G4kKYR+yUEz2WNgnTNZV4NGML7kXOL0Pk3/S
                                    TtGIca8E93nG96h/2SMqznyCEcbw/J+v3I9j
                                    y+RLfoJhnvHc6fVXhcEK2RpFOzRnjPi+MRcE
                                    mF8Bei1jvV0hOWG8iyAP5vjL4++Xsh9BJgR8
                                    l+TeUo4UwGKcD6NSpNvUeeowuiA=
                                    )
stronghold2.ltfe-sphere.org. 3600   IN A  212.101.143.7
                          3600  RRSIG A 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    E5NBPmnoZ+YON/AFK9JWuYrID6/OumEHHr3L
                                    X+KOQsVYi8Ejg6awAhxi/+BBrUFh1Kledg/s
                                    mWleysvF988MYeua39EZGr6h0uACnztk8Fl3
                                    Ww53Ak9jhUSSsgd85nyd/8vyn5NmVoZ+FDrW
                                    qK1YHigWCMdjbEF5lyLvgkHg47g=
                                    )
                          3600  AAAA  2a00:1368:1000:20::7
                          3600  RRSIG AAAA 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    ps0R6DdfecgAZA4LOv7bCefrVVdOzscdq7DU
                                    hQbMva83026NNwgoRuE62idfru5CmmDXXF/3
                                    BTaAikRxQDGx/moXOay8J8Qa94c8CC1SsJ5i
                                    fK4B7yRdZtOSINog8jKKwK+5++RBoc8H/at6
                                    6/mgWTNPW3COQ/Wfol5HBoF4kjw=
                                    )
                          3600  NSEC  throne.ltfe-sphere.org. A AAAA RRSIG
NSEC
                          3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    srSZyi/t7QAlnXoTtzdj1DPO69uqn/u/LfZV
                                    v3wb3p+p2uT65lnInkwQyVdw/XRj3LxhZwyY
                                    qa3Y1bGnJPK+PTCeZGC5HwDVdm17GaX5gm59
                                    iM0mWK6VoH46STxA7PDnB1CVJIz80U5VPJBr
                                    foLZMiTanlFi0Dv/bIsdlQh4L5A=
                                    )
throne.ltfe-sphere.org.  3600  IN A  212.101.143.177
                          3600  RRSIG A 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    PsLR9xjzj7fXV7QSS2z2jxajZYa4RhcLHhzg
                                    P7mqMpuyutZruQCduASyq7X2ZSzzhpqGU9Gj
                                    0RZ0ZWghiGYsjQWkg9kE6a25kAlQcqQ20NM6
                                    I2Z5ZUQcf7W5VeGHPa9hbpGBYmhVoAxkTpZN
                                    EJAFkoJdOdoWRKXN6rvucMdZ0wY=
                                    )
                          3600  AAAA  2a00:1368:1000:20::177
                          3600  RRSIG AAAA 5 3 3600 20110526140539 (
                                    20110426140539 55812 ltfe-sphere.org.
                                    p1CexOfWXrPvgGGBpCXFfy1Op2IW9XoQrp9A
                                    +wTV41ZxQmnE9mErL2w3rpZsHujr4iTCFUlp
                                    SNf2QRBxNK5vQ27NtdiG1oMB4Hld7UGyTVtT
                                    m1j7kiWZDT4z7Ie2bzuuYs9dZq3QkBa/YZyY
```

```
                                /TQOuIKnI/hm+5e1mUX6RxjpOsI=
                                )
                        3600  NSEC  unsigned.ltfe-sphere.org. A AAAA RRSIG
NSEC
                        3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                20110426140539 55812 ltfe-sphere.org.
                                MvTE1RW3BLLB//q1FBIly2Xs36TgPnrZsl8W
                                vzYmMsm6v/9VWY+liPdm4Hd05CRuJksNdjYE
                                6mk5FUBh4xYFURGZDTNQDqhaC7lBLWGqtgvt
                                2D7Zf4QYr/vPrGLyPEGfjVKVHb1EQMQbL06U
                                +IZ16fjKXRa3JFlBT0SJ2Dhj2Kg=
                                )
unsigned.ltfe-sphere.org. 3600      IN NS ns1.kozic.net.
                        3600  NSEC  ltfe-sphere.org. NS RRSIG NSEC
                        3600  RRSIG NSEC 5 3 3600 20110526140539 (
                                20110426140539 55812 ltfe-sphere.org.
                                fsCiQk5awKQEFLL8qhd2G4CPce4so9E9HB5/
                                kgTI7aysby5F/mdUDNsKscECuBROpQLr6awc
                                Wc4syPEH1PNRQV6njiNeVkVVOGInwVa0lDx7
                                FnlMANmaGUbGplBPd38OimK4HhPUFYmQGI9K
                                L2DLc/OTVsI5hjsguEu5w7mxpV4=
                                )
ns1.kozic.net.          3600  IN A  93.103.130.109
```

2 Skripta PowerShell za delo z DNS v Windows Server

2.1 Podpis cone

```
PS C:\Users\Administrator\dnssec> .\SignZone.ps1 -action sign -zone ltfe-
sphere.org

Action: sign zone ltfe-sphere.org

Generated new KSK: ksk-ltfe-sphere.org-20110405-101909

Generated new ZSK: zsk-ltfe-sphere.org-20110405-101909

Commit: saved modified zone ltfe-sphere.org to server

Completed: signed zone ltfe-sphere.org
    with new KSK ksk-ltfe-sphere.org-20110405-101909
    and new ZSK zsk-ltfe-sphere.org-20110405-101909

PS C:\Users\Administrator\dnssec>
```

2.2 Ponovno podpisovanje cone

```
PS C:\Users\Administrator\dnssec> .\SignZone2.ps1[8] -action resign -zone
ltfe-sphere.org -ksk ksk-ltfe-sphere.org-20110405-104201 -zsk zsk-ltfe-
sphere.org-20110405-104201

Action: re-sign zone ltfe-sphere.org

Commit: saved modified zone ltfe-sphere.org to server

Completed: re-signed zone ltfe-sphere.org
    with existing KSK ksk-ltfe-sphere.org-20110405-104201
    and existing ZSK zsk-ltfe-sphere.org-20110405-104201

PS C:\Users\Administrator\dnssec>
```

2.3 Menjava ključa ZSK z metodo enojnega podpisovanja

```
PS C:\Users\Administrator\dnssec>
.\SignZone2.ps1 -action prezsk -zone ltfe-sphere.org -ksk ksk-ltfe-
sphere.org-20110405-104201 -zsk zsk-ltfe-sphere.org-20110405-104201

Action: pre-published ZSK rollover of zone ltfe-sphere.org

This rollover process will take approximately 7200 seconds (zoneTTL * 2)
Do you want to continue (y/n)?
y

Generated new ZSK: zsk-ltfe-sphere.org-20110405-132222

Stage 1: signed zone ltfe-sphere.org
    with old KSK ksk-ltfe-sphere.org-20110405-104201
    old ZSK zsk-ltfe-sphere.org-20110405-104201
    and added new ZSK zsk-ltfe-sphere.org-20110405-132222

Commit: saved modified zone ltfe-sphere.org to server

Sleeping zone TTL 3600 seconds...

Stage 2: signed zone ltfe-sphere.org
    with new ZSK zsk-ltfe-sphere.org-20110405-132222
    and added old ZSK zsk-ltfe-sphere.org-20110405-104201

Commit: saved modified zone ltfe-sphere.org to server

Sleeping zone TTL 3600 seconds...

Stage 3: signed zone ltfe-sphere.org
    with old KSK ksk-ltfe-sphere.org-20110405-104201
    and new ZSK zsk-ltfe-sphere.org-20110405-132222

Commit: saved modified zone ltfe-sphere.org to server
```

[8] Uporabljam modificirano skripto, ki sem jo poimenoval SignZone2.ps1. Le-ta kreira daljši ključ KSK (2048 bitov) od ključa ZSK (1024 bitov)

```
Completed: pre-published ZSK rollover of zone ltfe-sphere.org
```

2.4 Menjava ključa ZSK z metodo dvojnega podpisovanja

```
PS C:\Users\Administrator\dnssec>
.\SignZone2.ps1 -action doublezsk -zone ltfe-sphere.org -ksk ksk-ltfe-
sphere.org-20110405-104201 -zsk zsk-ltfe-sphere.org-20110405-104201

Action: double signature ZSK rollover of zone ltfe-sphere.org

This rollover process will take approximately 3600 seconds (zoneTTL)
Do you want to continue (y/n)?
y

Generated new ZSK: zsk-ltfe-sphere.org-20110405-112443

Stage 1: signed zone ltfe-sphere.org
    with old KSK ksk-ltfe-sphere.org-20110405-104201
    old ZSK zsk-ltfe-sphere.org-20110405-104201
    and new ZSK zsk-ltfe-sphere.org-20110405-112443

Commit: saved modified zone ltfe-sphere.org to server

Sleeping zone TTL 3600 seconds...

Stage 2: signed zone ltfe-sphere.org
    with old KSK ksk-ltfe-sphere.org-20110405-104201
    and new ZSK zsk-ltfe-sphere.org-20110405-112443

Commit: saved modified zone ltfe-sphere.org to server

Completed: double signature ZSK rollover of zone ltfe-sphere.org

PS C:\Users\Administrator\dnssec>
```

2.5 Menjava ključa KSK

```
PS C:\Users\Administrator\dnssec>
.\SignZone2.ps1 -Action DoubleKsk -Zone ltfe-sphere.org -Ksk ksk-ltfe-
sphere.org-20110405-104201 -Zsk zsk-ltfe-sphere.org-20110405-132222 -Ttl
86400

Action: double signature KSK rollover of zone ltfe-sphere.org

This rollover process will take at least 86400 seconds (dsTTL).

You should have obtained the TTL of the DS record in the parent
zone that corresponds to the KSK you provided. During the rollover,
you will be asked to provide the new DS record set to the owner of
the parent zone. The owner of the parent zone must then replace
the original DS set with the new "dsset-ltfe-sphere.org" and
"keyset-ltfe-sphere.org" files that point to the new KSK.

Do you want to continue (y/n)?
```

```
y
Generated new KSK: ksk-ltfe-sphere.org-20110406-114453

Stage 1: signed zone ltfe-sphere.org

    with new KSK ksk-ltfe-sphere.org-20110406-114453
    old KSK ksk-ltfe-sphere.org-20110405-104201
    and old ZSK zsk-ltfe-sphere.org-20110405-132222

Commit: saved modified zone ltfe-sphere.org to server

Provide the new DS record set to the owner of the parent zone.

The owner of the parent zone must replace the original DS
record set with the new "dsset-ltfe-sphere.org" file.
DS points to the new KSK ksk-ltfe-sphere.org-20110406-114453.

The dsset-ltfe-sphere.org file contains:

ltfe-sphere.org.        3600    IN DS   16234 5 1 (

C00662FC5964914C17AFA791985E7EEB0B7F
                                        C3BC )
                        3600    DS      16234 5 2 (

29BA07A383D0FFA1CA01457C36F7E11EFF26
                                        B0F20F7B2B88AE748EB852D3BD67 )
                        3600    DS      38831 5 1 (

2F48A897059BA4B15156AF65D53A7808B677
                                        F616 )
                        3600    DS      38831 5 2 (

0A9CD78BBDF75AFABFAF4C95285EC7C54E51
                                        0D80601582E2EFAA18878AFE49EF )

After the parent has updated the record, continue.
Script will sleep 86400 seconds after continue.

Do you want to continue (y/n)?
y

Sleeping DS TTL 86400 seconds...

Stage 2: signed zone ltfe-sphere.org
    with new KSK ksk-ltfe-sphere.org-20110406-114453
    and old ZSK zsk-ltfe-sphere.org-20110405-132222

Commit: saved modified zone ltfe-sphere.org to server

Completed: double signature KSK rollover of zone ltfe-sphere.org
```

3 DNSSEC v Bind

3.1 Primer definicije cone v /etc/bind/named.conf

Dodani so stavki za samodejno vzdrževanje cone (krepko).

```
zone "kozic.net" {
        auto-dnssec maintain;
        type master;
        update-policy local;
        file "/etc/bind/kozic.loc/db.kozic.net.signed";
        key-directory "/etc/bind/kozic.net/keys";
};
```

3.2 Primer datoteke s ključem

```
; This is a key-signing key, keyid 24633, for kozic.net.
; Created: 20110407195744 (Thu Apr  7 21:57:44 2011)
; Publish: 20110407195744 (Thu Apr  7 21:57:44 2011)
; Activate: 20110407195744 (Thu Apr  7 21:57:44 2011)
kozic.net. IN DNSKEY 257 3 10
AwEAAcxn92ovRGEauGH1e7GMiFzArroNsvx5yB4RgDfApSraFLsRaNr2
4OlcywLQwPnt90fhGldRwRQhpKhhSr32WIFll4BIEBGTjRbT7rtYlnnw
FKOXk+BA+Ew0d1Iq9zChgeKiZPNBIvrv7S1sT1zQcEGlrX1R5thwyL6r
kxuLKTxEXaZvlnmeeN7Ht6bd3eRrAdHH+vpGAXYmMsNu1/pyWN9OpyeA
1qL1tCT9d7T0LxHXJ6CNsgjKFBRFzS6tWA1AkiWWVnHYo3gTBYaqd5DV
wqRk0im+r3WqPcblx6oe/SIq9mo9Lu9NtYUqnv0xE8U9z+09HdRUaYHz Ja0jkUncyvs=

Private-key-format: v1.3
Algorithm: 10 (RSASHA512)
Modulus:
zGf3ai9EYRq4YfV7sYyIXMCuug2y/HnIHhGAN8ClKtoUuxFo2vbg6VzLAtDA+e33R+EaV1HBFCG
kqGFKvfZYgWWXgEgQEZONFtPuu1iWefAUo5eT4ED4TDR3Uir3MKGB4qJk80Ei+u/tLWxPXNBwQa
WtfVHm2HDIvquTG4spPERdpm+WeZ543se3pt3d5GsB0cf6+kYBdiYyw27X+nJY306nJ4DWovW0J
P13tPQvEdcnoI2yCMoUFEXNLq1YDUCSJZZWcdijeBMFhqp3kNXCpGTSKb6vdao9xuXHqh79Iir2
aj0u7021hSqe/TETxT3P7T0d1FRpgfMlrSORSdzK+w==
PublicExponent: AQAB
PrivateExponent:
fysrhXd0vkkGb4Nvei2pZLNHwV7mfScy9moC8Cy9VWSqQxv4tUmk8eqWLuE7PgLrtl94qNynF+Q
6cbtd6cNKmu5ejCPJQ3Yd8BbYTtw1ehWAcarLCFjDYX2+wnMSqMHCcgKAdcqPfbkpQPpCgcb4IA
94WSBMU1oPung6JIe2ltQBBOkxcFzk/D5FGFNCVMszN3Eqy0G1nR9w8udR9NDnTbvfXqSMuC9P6
LkU72b25SWUXOot0Kp1tvE7XPZYYAZFbFOsfr3xsiupZVLbV1raSfGmTUaInUOwaRde5MjMkQbo
1PrIhZZQrB+zcnet8Ti+CW3Q0lVBEwg1vTvsyS8WkQ==
Prime1:
6yw6gDs5vi09etntmW9Zdn9OktHxVg9Vs2eRK/+z9prGLcjN07tro5/d6bfs+TvUun/2Y2H9huD
vGnpQkjaS9gN6ccPcJPabd4TIou1viiMXoEH9VEmxFJd2bjX+rOipVsaBY/XwWo3KPBzkh83bej
NAeMh0ql/XQ+2FNP62cqk=
Prime2:
3oI0fpd15Jixlw13640TJsi+ODRoYQtgvjsyQkPFj0NNJxJ9RR1Y5uk8k6etN2E8Y0EPlt0Lgvn
yGwxnVnJMDjNEl7tjdIpv3bcQYft0078QrfwXyfdjN0HVM7fGRNLcGACVunvmLam51bSi6doSLg
gemSDoNSubU93iRTs4uwM=
Exponent1:
Jj3V8X4wAJXYtn5znYy2zocbHtLW3Li3penEeJQnDoXSjiVmGvUSJtRbQKMw/MJW8FDYnwaTNHi
```

```
lSPf03CHVuf2Mzq0bwkL7uu6ztZLInXzhXaT9bdRh4EhU8M0E8dcPmlXDOp8Gi6Mc7LOCEfdOpk
tpQ3rNRds55QjdrWVyA6k=
Exponent2:
WT7EK1o4kP2ljT57sOCeo9m0YfqzOulDXkmD+QwRu0p1MYy67fO9arB1+YsxcJrxi6XWyN3T9i6
mKq8o64Vgd/mcM3eXhnDYOYdpbMNcsmlaGU6ZhApToyAmkV3OimBAoufpJij3w6zluo9/IGi1Df
13uFd4i4r3/AVs8fO0510=
Coefficient:
V1DoMX1+/71VkqcycQF5NtUuS1TfHnxpXsyEvDHJF1fhPvxCzONkf0PJpiAz8dpDw1sVNITciVv
gMfz+q74+h8uE3XHWwNJ+mgVghpnQ6HS8qIEDnQSnMU77iqVz8LpRIpwYpSYKfXQ1pcd3ozkmFW
aSomhSSd1V2r0d9FsA8zk=
Created: 20110407195744
Publish: 20110407195744
Activate: 20110407195744
```

4 Konfiguracija OpenDNSSEC

4.1 conf.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<Configuration>

	<RepositoryList>
		<Repository name="SoftHSM">
			<Module>/usr/lib/libsofthsm.so</Module>
			<TokenLabel>OpenDNSSEC</TokenLabel>
			<PIN>3355</PIN>
		</Repository>
	</RepositoryList>


	<Common>
		<Logging>
			<Syslog><Facility>local0</Facility></Syslog>
		</Logging>

		<PolicyFile>/etc/opendnssec/kasp.xml</PolicyFile>
		<ZoneListFile>/etc/opendnssec/zonelist.xml</ZoneListFile>
	</Common>


	<Enforcer>
		<Datastore><SQLite>/var/lib/opendnssec/db/kasp.db</SQLite></Datastore>
		<Interval>PT3600S</Interval>
	</Enforcer>


	<Signer>
		<WorkingDirectory>/var/lib/opendnssec/tmp</WorkingDirectory>
		<WorkerThreads>8</WorkerThreads>
		<NotifyCommand>/usr/sbin/rndc reload %zone</NotifyCommand>
	</Signer>


	<Auditor>
		<Privileges>
			<User>opendnssec</User>
			<Group>opendnssec</Group>
		</Privileges>
```

```
            <WorkingDirectory>/var/lib/opendnssec/tmp</WorkingDirectory>
      </Auditor>

</Configuration>
```

4.2 kasp.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!--

  NOTE:  The default policy below is a TEMPLATE ONLY and should be reviewed
         before used in any production environment. The administrator should
         consult the OpenDNSSEC documentation before changing any parameters.

         If you can read this message, it is likely that this file has not
         been reviewed nor updated.

  -->

<KASP>

      <Policy name="default">
            <Description>Privzeta politika: RSASHA1 NSEC3 (algoritem
7)</Description>

            <!-- Veljavnost podpisa 2 uri -->
            <Signatures>
                  <Resign>PT15M</Resign>
                  <Refresh>PT30M</Refresh>
                  <Validity>
                              <Default>PT2H</Default>
                              <Denial>PT2H</Denial>
                  </Validity>
                  <Jitter>PT0H</Jitter>
                  <InceptionOffset>PT3600S</InceptionOffset>
            </Signatures>


            <!-- Za negativne odgovore uporabljam NSEC3
                  Salt menjam na 100 dni, 5 iteracij, dolzina salt 8 znakov
             -->

            <Denial>
                  <NSEC3>
                        <Resalt>P100D</Resalt>
                        <Hash>
                              <Algorithm>1</Algorithm>
                              <Iterations>5</Iterations>
                              <Salt length="8"/>
                        </Hash>
                  </NSEC3>
            </Denial>

            <Keys>
                  <!-- Parameters for both KSK and ZSK -->
                  <TTL>PT3600S</TTL>
                  <RetireSafety>PT3600S</RetireSafety>
                  <PublishSafety>PT3600S</PublishSafety>
                  <Purge>P2D</Purge>

                  <!-- KSK RSASHA1, dolzina 2048 bitov
                        veljavnost 1 leto
```

```
                 -->
                <KSK>
                        <Algorithm length="2048">7</Algorithm>
                        <Lifetime>P1Y</Lifetime>
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
                </KSK>

                <!-- ZSK RSASHA1, dolzina 1024 bit
                        veljavnost 6 ur
                 -->
                <ZSK>
                        <Algorithm length="1024">7</Algorithm>
                        <Lifetime>PT6H</Lifetime>
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
                </ZSK>
        </Keys>



        <!-- PropagationDelay = cas, da se spremenjena datoteka cone prenese
na vse sekundarne streznike
        TTL cone = 3600 sekund
        Serijska stevilka cone = datum
        -->

        <Zone>
                <PropagationDelay>PT43200S</PropagationDelay>
                <SOA>
                        <TTL>PT3600S</TTL>
                        <Minimum>PT3600S</Minimum>
                        <Serial>datecounter</Serial>
                </SOA>
        </Zone>


        <!-- TTL-ji za parent cono + TTL zapisa DS (1 ura) -->

        <Parent>
                <PropagationDelay>PT9999S</PropagationDelay>
                <DS>
                        <TTL>PT3600S</TTL>
                </DS>
                <SOA>
                        <TTL>PT172800S</TTL>
                        <Minimum>PT10800S</Minimum>
                </SOA>
        </Parent>

        <Audit>
                <!-- <Partial /> -->
        </Audit>

</Policy>



 <Policy name="RSASHA1">
        <Description>Politika RSASHA1 NSEC (algoritem 5)</Description>

        <!-- Veljavnost podpisa 1 mesec -->
          <Signatures>
                  <Resign>P1D</Resign>
                  <Refresh>P2D</Refresh>
```

```
            <Validity>
                        <Default>P1M</Default>
                        <Denial>P1M</Denial>
            </Validity>
            <Jitter>PT0H</Jitter>
            <InceptionOffset>PT3600S</InceptionOffset>
    </Signatures>


<!-- Za negativne odgovore uporabljam NSEC -->
    <Denial>
            <NSEC/>
    </Denial>

    <Keys>
            <!-- Parameters for both KSK and ZSK -->
            <TTL>PT3600S</TTL>
            <RetireSafety>PT3600S</RetireSafety>
            <PublishSafety>PT3600S</PublishSafety>
            <!-- <ShareKeys/> -->
            <Purge>P7D</Purge>


        <!-- KSK RSASHA1, dolzina 2048 bitov
               veljavnost 1 leto
         -->
            <KSK>
                    <Algorithm length="2048">5</Algorithm>
                    <Lifetime>P1Y</Lifetime>
                    <Repository>SoftHSM</Repository>
                    <Standby>1</Standby>
            </KSK>


        <!-- ZSK RSASHA1, dolzina 1024 bit
               veljavnost 6 ur
         -->
            <ZSK>
                    <Algorithm length="1024">5</Algorithm>
                    <Lifetime>P1M</Lifetime>
                    <Repository>SoftHSM</Repository>
                    <Standby>1</Standby>
                    <!-- <ManualRollover/> -->
            </ZSK>
    </Keys>



        <!--
        TTL cone = 3600 sekund
        Serijska stevilka cone = datum
        -->

    <Zone>
            <PropagationDelay>PT30M</PropagationDelay>
            <SOA>
                    <TTL>PT3600S</TTL>
                    <Minimum>PT3600S</Minimum>
                    <Serial>datecounter</Serial>
            </SOA>
    </Zone>


<!-- TTL zapisa DS 1 ura -->
    <Parent>
```

```
                <PropagationDelay>PT9999S</PropagationDelay>
                <DS>
                        <TTL>PT3600S</TTL>
                </DS>
                <SOA>
                        <TTL>PT172800S</TTL>
                        <Minimum>PT10800S</Minimum>
                </SOA>
        </Parent>

        <Audit>
                <!-- <Partial /> -->
        </Audit>

</Policy>


<Policy name="DSASHA1">
        <Description>SoftHSM ne podpira algoritma DSA</Description>
        <Signatures>
                <Resign>PT15M</Resign>
                <Refresh>PT30M</Refresh>
                <Validity>
                                <Default>PT2H</Default>
                                <Denial>PT2H</Denial>
                </Validity>
                <Jitter>PT0H</Jitter>
                <InceptionOffset>PT3600S</InceptionOffset>
        </Signatures>

        <Denial>
                <NSEC3>
                        <!-- <OptOut/> -->
                        <Resalt>P100D</Resalt>
                        <Hash>
                                <Algorithm>1</Algorithm>
                                <Iterations>5</Iterations>
                                <Salt length="8"/>
                        </Hash>
                </NSEC3>
        </Denial>

        <Keys>
                <!-- Parameters for both KSK and ZSK -->
                <TTL>PT3600S</TTL>
                <RetireSafety>PT3600S</RetireSafety>
                <PublishSafety>PT3600S</PublishSafety>
                <!-- <ShareKeys/> -->
                <Purge>P2D</Purge>

                <!-- Parameters for KSK only -->
                <KSK>
                        <Algorithm length="2048">3</Algorithm>
                        <Lifetime>P7D</Lifetime>
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
                </KSK>

                <!-- Parameters for ZSK only -->
                <ZSK>
                        <Algorithm length="1024">3</Algorithm>
                        <Lifetime>PT6H</Lifetime>
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
```

```
                        <!-- <ManualRollover/> -->
                </ZSK>
        </Keys>

        <Zone>
                <PropagationDelay>PT43200S</PropagationDelay>
                <SOA>
                        <TTL>PT3600S</TTL>
                        <Minimum>PT3600S</Minimum>
                        <Serial>datecounter</Serial>
                </SOA>
        </Zone>


      <!-- TTL zapisa DS 1 ura -->
        <Parent>
                <PropagationDelay>PT9999S</PropagationDelay>
                <DS>
                        <TTL>PT3600S</TTL>
                </DS>
                <SOA>
                        <TTL>PT172800S</TTL>
                        <Minimum>PT10800S</Minimum>
                </SOA>
        </Parent>

        <Audit>
                <!-- <Partial /> -->
        </Audit>

 </Policy>



<Policy name="RSASHA256NSEC">
        <Description>Politika RSASHA256 (algoritem 8) NSEC</Description>
        <Signatures>
                <Resign>PT15M</Resign>
                <Refresh>PT30M</Refresh>
                <Validity>
                                <Default>PT2H</Default>
                                <Denial>PT2H</Denial>
                </Validity>
                <Jitter>PT0H</Jitter>
                <InceptionOffset>PT3600S</InceptionOffset>
        </Signatures>

      <!-- Za negativne odgovore uporabljam NSEC -->
        <Denial>
                <NSEC/>
        </Denial>

        <Keys>
                <!-- Parameters for both KSK and ZSK -->
                <TTL>PT3600S</TTL>
                <RetireSafety>PT3600S</RetireSafety>
                <PublishSafety>PT3600S</PublishSafety>
                <Purge>P2D</Purge>

            <!-- KSK RSASHA256, dolzina 2048 bitov
                   veljavnost 7 dni
             -->
                <KSK>
                        <Algorithm length="2048">8</Algorithm>
                        <Lifetime>P7D</Lifetime>
```

```
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
                </KSK>

                <!-- ZSK RSASHA256, dolzina 2048 bitov
                       veljavnost 6 ur
                 -->
                <ZSK>
                        <Algorithm length="1024">8</Algorithm>
                        <Lifetime>PT6H</Lifetime>
                        <Repository>SoftHSM</Repository>
                        <Standby>1</Standby>
                </ZSK>
        </Keys>


                <!--
                TTL cone = 3600 sekund
                Serijska stevilka cone = datum
                -->
        <Zone>
                <PropagationDelay>PT43200S</PropagationDelay>
                <SOA>
                        <TTL>PT3600S</TTL>
                        <Minimum>PT3600S</Minimum>
                        <Serial>datecounter</Serial>
                </SOA>
        </Zone>


      <!-- TTL zapisa DS 1 ura -->
        <Parent>
                <PropagationDelay>PT9999S</PropagationDelay>
                <DS>
                        <TTL>PT3600S</TTL>
                </DS>
                <SOA>
                        <TTL>PT172800S</TTL>
                        <Minimum>PT10800S</Minimum>
                </SOA>
        </Parent>

        <Audit>
                <!-- <Partial /> -->
        </Audit>

 </Policy>


<Policy name="RSASHA512NSEC3">
        <Description> Politika RSASHA512 (algoritem 10) NSEC3</Description>
        <Signatures>
                <Resign>PT15M</Resign>
                <Refresh>PT30M</Refresh>
                <Validity>
                                <Default>PT2H</Default>
                                <Denial>PT2H</Denial>
                </Validity>
                <Jitter>PT0H</Jitter>
                <InceptionOffset>PT3600S</InceptionOffset>
        </Signatures>


      <!-- Za negativne odgovore uporabljam NSEC3
```

```
      Salt menjam na 100 dni, 5 iteracij, dolzina salt 8 znakov
 -->

  <Denial>
          <NSEC3>
                  <!-- <OptOut/> -->
                  <Resalt>P100D</Resalt>
                  <Hash>
                          <Algorithm>1</Algorithm>
                          <Iterations>5</Iterations>
                          <Salt length="8"/>
                  </Hash>
          </NSEC3>
  </Denial>

  <Keys>
          <!-- Parameters for both KSK and ZSK -->
          <TTL>PT3600S</TTL>
          <RetireSafety>PT3600S</RetireSafety>
          <PublishSafety>PT3600S</PublishSafety>
          <Purge>P2D</Purge>

             <!-- KSK RSASHA512, dolzina 2048 bitov
             veljavnost 7 dni
              -->

          <KSK>
                  <Algorithm length="2048">10</Algorithm>
                  <Lifetime>P7D</Lifetime>
                  <Repository>SoftHSM</Repository>
                  <Standby>1</Standby>
          </KSK>


      <!-- ZSK RSASHA512, dolzina 1024 bitov
             veljavnost 6 ur
       -->

          <ZSK>
                  <Algorithm length="1024">10</Algorithm>
                  <Lifetime>PT6H</Lifetime>
                  <Repository>SoftHSM</Repository>
                  <Standby>1</Standby>
          </ZSK>
  </Keys>


      <!--
      TTL cone = 3600 sekund
      Serijska stevilka cone = datum
      -->

  <Zone>
          <PropagationDelay>PT43200S</PropagationDelay>
          <SOA>
                  <TTL>PT3600S</TTL>
                  <Minimum>PT3600S</Minimum>
                  <Serial>datecounter</Serial>
          </SOA>
  </Zone>


<!-- TTL zapisa DS 1 ura -->

  <Parent>
```

```
                        <PropagationDelay>PT9999S</PropagationDelay>
                        <DS>
                                <TTL>PT3600S</TTL>
                        </DS>
                        <SOA>
                                <TTL>PT172800S</TTL>
                                <Minimum>PT10800S</Minimum>
                        </SOA>
                </Parent>

                <Audit>
                        <!-- <Partial /> -->
                </Audit>

        </Policy>


</KASP>
```

4.3 zonelist.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- seznam datotek s conami -->

<ZoneList>


<-- cona kozic.net uporablja policy default -->

<Zone name="kozic.net">
       <Policy>default</Policy>
       <SignerConfiguration>/var/lib/opendnssec/signconf/kozic.net.xml</SignerConfig
       uration>
       <Adapters>
              <Input>
              <File>/var/lib/opendnssec/unsigned/db.kozic.net</File>
              </Input>
              <Output>
              <File>/etc/bind/kozic.net/db.kozic.net.signed2</File>
              </Output>
       </Adapters>
</Zone>



<-- cona sah-drustvo-ms.si uporablja policy RSASHA1 -->

<Zone name="sah-drustvo-ms.si">
       <Policy>RSASHA1</Policy>
       <SignerConfiguration>/var/lib/opendnssec/signconf/sah-drustvo-
ms.si.xml</SignerConfiguration>
       <Adapters>
            <Input>
               <File>/var/lib/opendnssec/unsigned/db.sah-drustvo-ms.si</File>
            </Input>
            <Output>
               <File>/etc/bind/sah-drustvo-ms.si/db.sah-drustvo-
ms.si.signed2</File>
            </Output>
       </Adapters>
</Zone>
```

```
<-- cona second.ltfe-sphere.org uporablja policy RSASHA1 -->

<Zone name="second.ltfe-sphere.org">
    <Policy>RSASHA1</Policy>
    <SignerConfiguration>/var/lib/opendnssec/signconf/second.ltfe-
sphere.org.xml</SignerConfiguration>
    <Adapters>
         <Input>
          <File>/etc/bind/ltfe/db.second.ltfe-sphere.org</File>
         </Input>
         <Output>
          <File>/etc/bind/ltfe/db.second.ltfe-sphere.org.signed2</File>
         </Output>
    </Adapters>
</Zone>



<-- reverse za 2a00:1368:1000:20::/64 uporablja policy RSASHA256NSEC -->

<Zone name="0.2.0.0.0.0.0.1.8.6.3.1.0.0.a.2.ip6.arpa">
       <Policy>RSASHA256NSEC</Policy>
<SignerConfiguration>/var/lib/opendnssec/signconf/db.ltfe.ip6.arpa</SignerConfigura
tion>
       <Adapters>
            <Input>
             <File>/etc/bind/ltfe/db.ltfe.ip6.arpa</File>
            </Input>
       <Output>
        <File>/etc/bind/ltfe/db.ltfe.ip6.arpa.signed2</File>
       </Output>
    </Adapters>
</Zone>

</ZoneList>
```