# Knot DNS workshop

## (aneb alternativy k BINDu)

**Jan Kadlec • jan.kadlec@nic.cz • 2.3.2014**

CZ.NIC | SPRÁVCE DOMÉNY CZ

# Program

- Řečičky o DNS
- Knot DNS
  - Konfigurace, ovládání
  - Transfery
  - DNSSEC
  - DDNS
  - Rate limiting
  - TSIG

# Akademie CZ.NIC

- internetové technologie v podání zkušených odborníků
- praktická cvičení na testovacím hardware
- kurzy na míru
- školení v Praze i v Brně

- *IPv6, protokol BGP, DNS/DNSSEC, PKI, IP telefonie, 3D TISK*

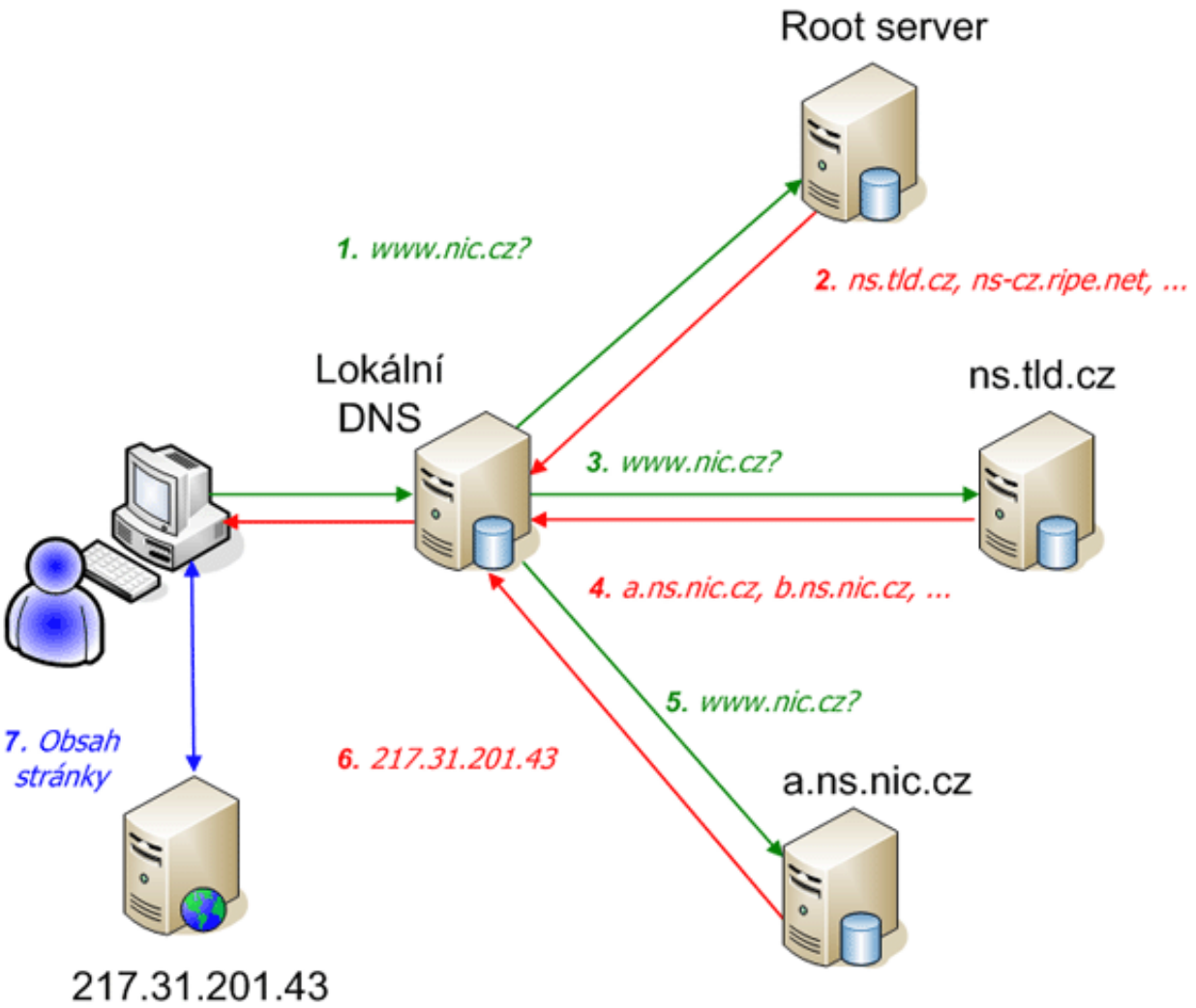
- **Principy a správa DNS**
  13. - 14. května v Praze
- **DNSSEC – zabezpečení DNS**
  18. března v Praze

Přihlašte se na: **akademie.nic.cz**

# DNS jednodušše

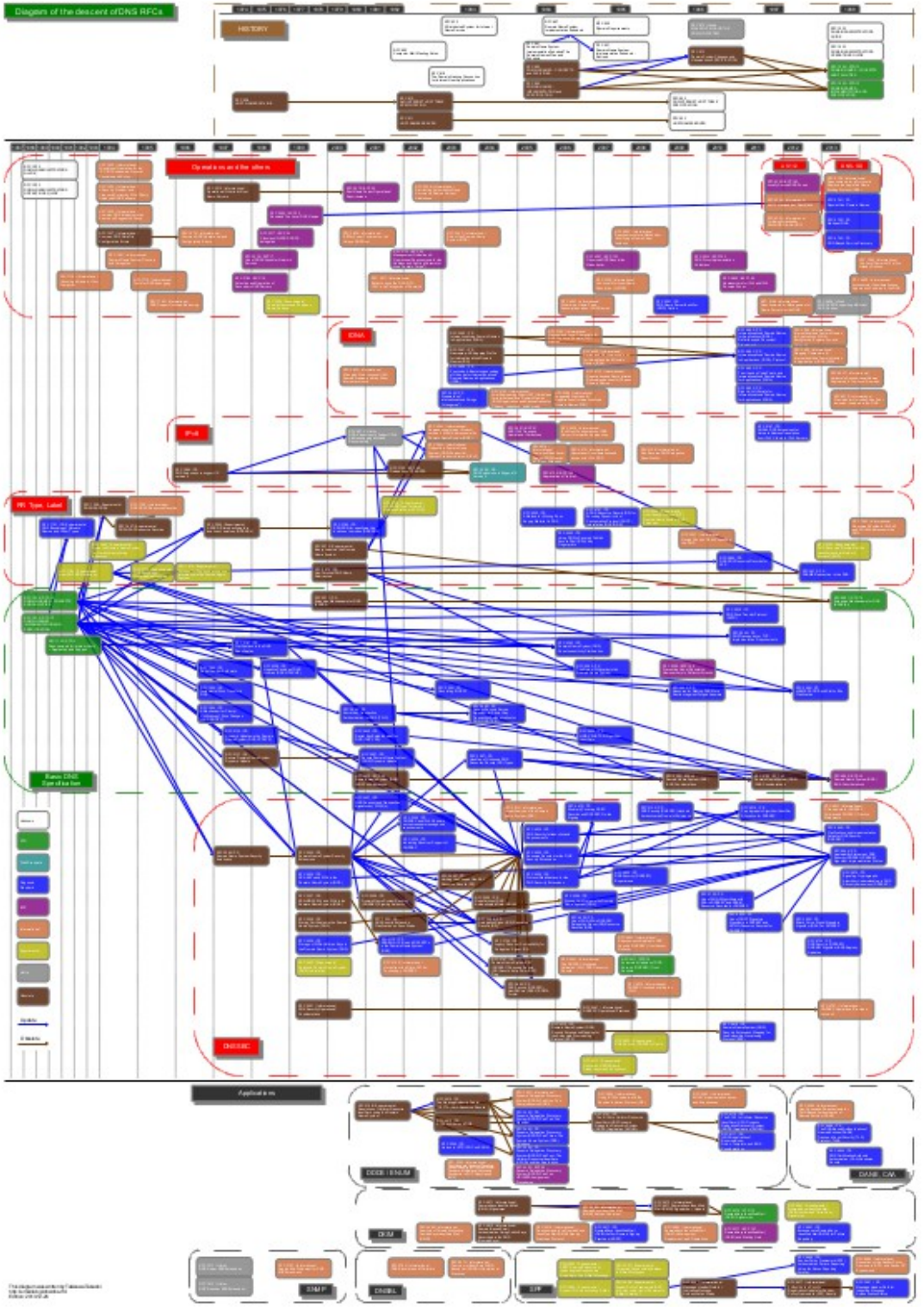
- Adresy → jména
- Distribuovaná hierarchická databáze
  - Používá cachování (TTL)
- 3 základní druhy serverů:
  - Stub resolver – v libc
  - Autoritativní – BIND, NSD, Knot DNS ...
  - Rekurzivní/resolver – BIND, Unbound …

# Obligátní obrázek



# DNS podrobně

Přijďte na školení :)



# DNS nástroje: dig

- **(k)dig**
  - Pošle DNS dotaz, odpověď hezky vypíše
  - Klient i pro transfery
  - DNSSEC, EDNS0
  - Nutnost pro debugging

# Unbound


- Pouze rekurzivní server
- Vyvíji NLnet Labs
- Výchozí nastavení „just works“
- Jednoduchý
- Ovládací nástroj unbound-control

# Unbound: úkol 1

- Nastartujte Unbound
  - sudo service unbound start
  - nebo: unbound-control start
- Zkontrolujte funkci
  - sudo unbound-control status
  - dig nic.cz. @127.0.0.1 (+dnssec)

# Knot DNS


- Autoritativní DNS server
- Umí:
  - DNSSEC, DDNS, {A,I}XFR, RRL ...
- Open-source: GPL
- Stále běží, stále odpovídá
- Aktivní vývoj, aktuální verze 1.4.3

# Knot DNS instalace

- Možnosti:
  - Ze zdrojových kódů
    - závislosti: flex, bison, openssl, liburcu
  - Balíčky pro:
    - Debian, Ubuntu, Fedora, Arch, OpenSUSE, OpenWRT

# Knot DNS úkol 1: instalace

- Přidejte repositář:

  $ sudo add-apt-repository ppa:cz.nic-labs/knot-dns

- Nainstalujte:

  $ sudo apt-get update

  $ sudo apt-get install knot knot-dnsutils

- Ověření:

  $ which knotd

  $ which kdig

# Knot DNS - nastavení

- knotc – ovládací nástroj

  stop/reload/signzone/flush ...
- Init skript / upstart

  /etc/init.d/knot …

  service knot start
- Výchozí konf. soubor:

  /etc/knot/knot.conf

# Knot DNS – konfigurační soubor

- Důležité sekce:
  - **system**
  - keys
  - interfaces
  - **control**
  - **remotes**
  - **zones**
  - log

# sekce system

- Globální nastevní serveru:

  **rundir “/var/lib/knot“;**

  - pracovní složka pro Knot – dočasné soubory, zónové soubory

  **workers 4;**

  - počet vláken serveru

  **rate-limit 1;**

  - vypne/zapne response rate limiting

# sekce keys

- Klíče pro zabezpečení DDNS/transferů
- Formát:

  jmeno.klice. hmac-sha1 “sdílené tajemství“;

# sekce interfaces

- Nastavení rozhraní
- Výchozí chování je pokusit se poslouchat na všech systémových rozhraních, port 53
- Je potřeba restartovat při změně
- Příklad:

```
moje_rozhrani {
  address 127.0.0.1;
  port 53;
}
```

# sekce control

- Nastavení komunikace mezi **knotc** a **knotd**
- Buď přes soket, nebo vzdáleně
- Pokud nenastavíme, lze ovládat jen přes signály

listen-on “knot-sock“;

# sekce remotes

- Vzdálené master/slave servery
- Příklad:

```
muj_master_server {
        address 127.0.0.1;
        port 53531;
        key key0.server0;  # (optional)
        via ipv4;  # (optional)
}
```

# sekce zones

- Obsahuje globální nastavení pro zóny a výčet zón
- Důležitá nastavení:
  - **storage** – složka pro slave zóny, pro žurnál
  - **semantic-checks –** důkladnější kontrola zón
  - **serial-policy** – SOA seriál increment/unixtime
  - **dnssec-enable** – automatické DNSSEC podepisování
  - **dnssec-keydir –** složka s DNSSEC klíči

# sekce zones

```
installfest.cz. {
    file “/etc/knot/installfest.cz.zone“;
    # … nastavení vlastností (přepíše globální)
    dnssec-enable off;
    xfr-in muj_master1, muj_master2;
    xfr-out muj_slave1, muj_slave2;
    notify-in muj_master1;
    notify-out muj_master2;
    update-in local_remote;
}
```

# sekce log

- Nastavení co a kam se bude logovat
- Výchozí nastavení: vše do syslogu

  ```
  $ tail -f /var/log/syslog
  ```

- Ostatní možnosti: stdout, stderr, libovolný soubor
- Kategorie: server, zone, answering, any
- Závažnosti: info, warning, error …

# Knot DNS: úkol 2: start serveru

- Vypněte Unbound:
  - $ service unbound stop
- Nastartujte Knot:
  - $ service knot start
  - $ knotd -c /etc/knot/knot.conf (pokud není initskript/upstart)
- Nemáme žádné zóny, ale server (snad) běží:
  - $ dig test @127.0.0.1
  - $ knotc status

# Knot DNS: úkol 3: přidání zóny 1/2

- Stáhnětě si ukázkovou zónu z:
  - https://secure.nic.cz//files/jkadlec/kurzy/dns/if/wszone
  - <NN> nahraďte posledním bytem z vaší IP
  - PC číslo 1 má IP 172.16.149.11 atd.
- Do sekce zones v konfiguračním souboru přidejte zónu “z<NN>.workshop.“ následujícím způsobem:

```
zones {
    z<NN>.workshop. {
        file “/etc/knot/z<NN>.workshop.zone“;
    }
}
```

# Knot DNS: úkol 3: přidání zóny 2/2

- Zkontrolujte změny v konfiguračním souboru
  - $ knotc checkconf
- Zkontrolujte zónu
  - $ knotc checkzone
- Reload konfigurace
  - $ knotc reload
- Ověřte dostupnost zóny:
  - $ knotc zonestatus
  - $ dig SOA z<NN>.workshop. @127.0.0.1

# DNS: Transfery zón

- Transfér zóny je způsob, jak replikovat data zóny na slave serveru
- Transfery se řídí pomocí **SOA RDAT**
  - **serial,** refresh, retry, expire
- Po každé změně zónových dat se musí seriál zvýšit!
- Dva druhy transferů:
  - AXFR – plný, přenese se celá zóna
  - IXFR – inkrementální, přenesou se jen změny

# Knot DNS: úkol 4: Transfery 1/3

- Úkolem je vyměnit si se sousedem zóny:
  - Vy budete slave pro souseda, soused slave pro vás
- Je třeba:
  - přidat do remotes sekce sousedovu IP
  - přidat novou zónu s xfr-in povoleným pro souseda
  - do vaší stávající zóny přidat xfr-out povolený pro souseda
  - lze povolit notify, ale není to nutné
  - reloadnout server

# Knot DNS: úkol 4: Transfery 2/3

- Remotes sekce:

```
soused1 {
   address 172.16.149.SS;
   port 53;
}
```

- Zones sekce:

```
zNN.workshop. {
  file “/etc/knot/zNN.workshop.zone“;
  xfr-out soused1;
  notify-out soused1;
}

zSS.workshop. {
  file “/var/lib/knot/zSS.workshop.slave““;
  xfr-in soused1;
  notify-in soused1;
}
```

**NN – váš byte**

**SS – sousedův byte**

# Knot DNS: úkol 4: Transfery 3/3

- Reloadněte konfiguraci:

  $ knotc reload

- Ověřte pomocí:
  - knotc:

    $ knotc zonestatus

  - digu na sousedovu zónu na vašem serveru:

    $ dig SOA z<SS>.workshop. @127.0.0.1

  - digu na vaši zónu na sousedově serveru:

    $ dig SOA z<NN>.workshop. @172.16.149.<SS>

**NN – váš byte**

**SS – sousedův byte**

# DNS: DNSSEC

- Způsob, jak ověřit, že DNS odpověd skutečně přisla, odkud měla, pomocí asymetrické kryptografie
- Ke standardním záznamům přidává podpisy
  - RR typ **RRSIG**
  - Resolvery tyto podpisy ověří a případně prohlásí odpověď za ověřenou (AD bit)
- Používají se dva druhy klíčů (RR typ DNSKEY):
  - Key Signing Key – silnější, podepisuje ostatní klíče
  - Zone Signing Key – slabší, podepisuje zónová data
- DNSKEY záznamy jsou s nadřazenou zónou spojeny DS záznamem
- Ruční správa téměř nemožná, nutno automatizovat

# DNS: DNSSEC nástroje

- Naprostá automatizace:
  - OpenDNSSEC, PowerDNS
- Něco mezi:
  - BIND, Knot DNS
    - umí inline podepisování, neumí sami generovat klíče ani key rollover (potřeba jednou za pár měsíců)
- “Ruční“ nástroje
  - dnssec-signzone, dnssec-keygen, dnssec-settime
- Online nástroje na ověření správnosti podepsání
  - dnsviz.net
  - dnssecheck.labs.nic.cz

# Knot DNS úkol 5: DNSSEC 1/3

- Vygenerujte klíče (v produkci použijte **/dev/random**):
  - ZSK:

    dnssec-keygen -r /dev/urandom z<NN>.workshop.
  - KSK:

    dnssec-keygen -f KSK -r /dev/urandom z<NN>.workshop.
- Klíče uložte do “/etc/knot/“

# Knot DNS úkol 5: DNSSEC 2/3

- Zapněte automatický DNSSEC pro vaši zónu:

```
zNN.workshop. {
  file “/etc/knot/zNN.workshop.zone“;
  xfr-out soused1;
  notify-out soused1;
  dnssec-enable on;
  dnssec-keydir “/etc/knot/“;
  (zonefile sync 1h;)
}
```

- Reloadněte server:
  - $ knotc reload

# Knot DNS úkol 5: DNSSEC 3/3

- Ověřte:
  - $ knotc zonestatus
  - $ dig z<NN>.workshop. @127.0.0.1 +dnssec
- Vyrobte DS záznamy:
  - $ dnssec-dsfromkey soubor_s_ksk (flags 257)
- Záznamy pošlete na lektorský počítač přes ssh
  - # scp z<NN>dsfile 172.16.149.10:/home/install
- Chvíli počkejte, až nadřazená zóna podepíše a přidá DS
- Ověřte, že vaše doména je validní
  - # dig SOA z<NN>.workshop. @172.16.149.10 **-p 53531**

# DNS: DDNS

- Způsob, jak měnit zónová data bez zásahu do zónového souboru
- Ideální pro malé změny:
  - Přidat záznam
  - Odebrat záznam
  - Změnit záznam

# DNS nástroje: nsupdate

- **k(nsupdate):**
  - vstup je textový soubor, příklad:

```
server 127.0.0.1
zone z<NN>.workshop.
update add ddns.z<NN>.workshop. 7200 TXT “mojedata“
update delete ddns.z<NN>.workshop TXT
show
send
```

# Knot DNS: úkol 6: DDNS 1/2

- Povolte dynamické updaty:
  - Do sekce remotes přidejte lokální remote:

```
local {
  address 127.0.0.1;
}
```

  - U vaší zóny v sekci zones povolte updaty:

```
zNN.workshop. {
  file “/etc/knot/zNN.workshop.zone“;
  xfr-out soused1;
  notify-out soused1;
  update-in local;
}
```

# Knot DNS: úkol 6: DDNS 2/2

- Připravte si soubor se změnami:
  - server 127.0.0.1
  - zone z<NN>.workshop.
  - update add … (nezapomenout na TTL)
  - (update delete … )
  - show send
- Pošlete změny na server:
  - (k)nsupdate < soubor_se_zmenami
- Změny by se měly projevit i na slave serveru

# DNS: response rate limiting

- Způsob, jak omezit DNS amplification útoky
  - Rozhodně není ideální, ideální je mít všude **BCP38**
- Základ: do podsítě, na kterou nejpíš jde útok, se pošle jen každá N-tá odpověď.
  - DNS tam ale nebude správně fungovat
- Lepší: každou, nebo skoro každou odpověď pošleme prázdnou s TC příznakem = RRL SLIP
  - Amplifikace není, odpovědi jsou stejně velké
  - Slušný resolver naváže TCP spojení DNS bude fungovat

# Knot DNS: úkol 7: Rate limiting

- Zapněte RRL v system sekci:

```
system {
  rate-limit 10; #zde záleží na nasazení, statisíce na TLD
  rate-limit-slip 1; #na každý dotaz TC odpověď
}
```

- Reloadněte server:

```
$ knotc reload
```

- Počkejte na útok a sledujte log
  - $ tail -f /var/log/syslog

# DNS: TSIG

- Způsob, jak podepsat transfery mezi master a slave servery (AXFR i IXFR)
- Používá **symetrickou** kryptografii
  - Podpisy se přidají do transferu
- Klíče mají 3 části, které musí být shodné na slavu a masteru:
  - Název klíče
  - Algoritmus, kterým byl klíč vytvořen
  - Sdílené tajemství klíče
- Slave server ověří celistvost dat pomocí klíče
  - Pokud podpisy nesedí, transfer odmítne

# Knot DNS úkol 8: TSIG 1/2

- Vygenerujte klíče: (v produkci **/dev/random**)

  $ dnssec-keygen -a HMAC-SHA512 -n HOST -b 512 -r /dev/urandom **jmeno-klice**

- Obsah souboru Kjmeno_klice.+165+XXXXX je klíč, který přepíšeme do konfigurace
- Nejdřív ale musíte poslat klíč sousedovi
  - scp Kjmeno_klice... 172.16.149.<SS>:/home/install/

# Knot DNS úkol 8: TSIG 2/2

- Vytvořte sekci keys:

```
keys {
  jmeno_klice hmac-sha512 “sdilene_tajemstvi“;
}
```

- Spárujte klíč se sousedovým remotem:

```
soused1 {
  address 172.16.149.SS;
  port 53;
  key jmeno_klice;
}
```

- Reload serveru
  - knotc reload
- Udělejte změnu do zóny (**serial**) a zkontrolujte přenos na slave

# Děkuji za pozornost

**Jan Kadlec • jan.kadlec@nic.cz**