DNS mit Bind 9

Wolfgang Dautermann

FH Joanneum

Chemnitzer Linuxtage 2008

Was ist das Nameservice?

DNS (Domain Name Service) ist eine weltweit verteilte, dezentrale Datenbank, welche Rechnernamen IP-Adressen zuordnet (und umgekehrt).

- weltweit verteilt, dezentral: nicht 'ein zentraler Server' vorhanden.
- Zuordnung Rechnernamen/IP-Adressen: ursprünglich als Erleichterung für Menschen gedacht (Namen sind einfacher zu merken als IP-Adressen)
- heute häufig auch als Marketingwerkzeug verwendet (`www.meinetollefirma.com`)...
- Zunehmend wird DNS auch für andere Einsatzzwecke verwendet.

Nameservice - Geschichte

- Ursprünglich: `/etc/hosts.txt` (vgl. /etc/hosts), auf einem zentralen Master-Server upgedatet, download von allen Rechnern im Internet. Zunehmend unhandlich.
- 1983 Spezifikation von DNS, erster DNS-Server (Jeeves).
- Etwas später: Entwicklung von Bind (Berkeley Internet Domain System)
- 1997 Bind Version 8
- Heute: Bind 9.4.2 (ISC)

Hierarchische Struktur

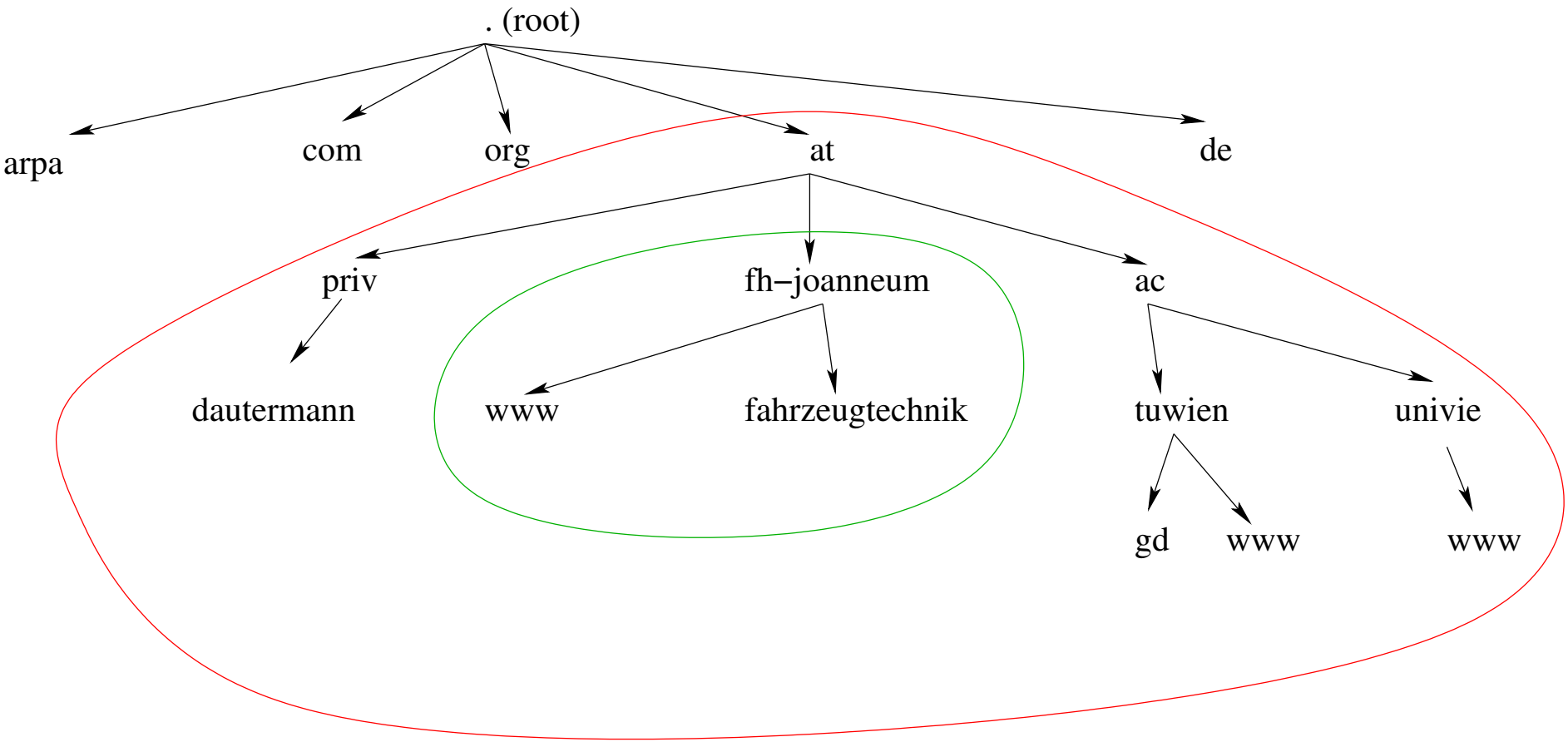

Abbildung: hierarchische Domain-Struktur

Hierarchische Struktur - Zuständigkeiten

- "." wird verwaltet von ICANN (Festlegung Top-Level-Domains, Betrieb der 13 Root Nameserver.
- ".at" delegiert an und verwaltet von nic.at (österr. Registry).
- ".priv.at" ("Privat") Für private Homepages interessant (gratis, für österreichische Privatpersonen, für nichtkommerziellen Gebrauch). delegiert an und verwaltet von www.nic.priv.at (Verein www.vibe.at)
- ".fh-joanneum.at" delegiert an und verwaltet von der FH Joanneum.
- ".linux-tage.de" delegiert an und verwaltet vom Individual Network Chemnitz e.V.

Ausfallsicherheit

In der Regel mehrere DNS-Server / Zone erforderlich.

- Root-Zone: 13 Server, weltweit verteilt.
- AT-Zone: 9 Server
- Second Level Zonen: mindestens zwei Server (Master/Slave), Konfiguation am Master, Slave übernimmt Konfiguration “automatisch”.

Schützt vor Nichterreichbarkeit durch Ausfall des Nameservers.

Ausfallsicherheit - Beispiele

Beispiel: Nameserver von at.

```
$ host -t ns at.
at name server ns2.univie.ac.at.
at name server ns1.univie.ac.at.
at name server ns9.univie.ac.at.
at name server sss-nl.nic.at.
at name server ns-us1.nic.at.
at name server ns-uk.nic.at.
at name server sss-us2.nic.at.
at name server sss-jp.nic.at.
at name server ns-de.nic.at.
```

Beispiel: Nameserver von linux-tage.de.

```
$ host -t ns linux-tage.de
linux-tage.de name server ns1.first-ns.de.
linux-tage.de name server robotns2.second-ns.de.
linux-tage.de name server robotns3.second-ns.com.
```

Installation

- `rpm`, `yast`, `apt-get`, `pkg-get`, ... oder:
- Download der aktuellen Version (dzt. 9.4.2) von http://www.isc.org/sw/bind/ (Dabei sind u.a. relevante RFCs, Bv9ARM (BIND 9 Administrator Reference Manual))
- `./configure ; make ; make install`

Installiert werden: Bind-Server (`named`), DNS-Client-tools (`nslookup`, `dig`, `host`, `nsupdate`, ...), Dokumentation (Manpages), Libraries, Include-Files, Admin- und Diagnosetools (`named-checkconf`, `named-checkzone`, `rdnc`, `rdnc-confgen`, `dnssec-keygen`, `dnssec-signzone`, ...)

Konfiguration

Konfiguration von Bind 9 als:

- Caching only Nameserver (nicht authorativ)
- authorative Nameserver für Zonen (Master/Slave/Stealth).
- beides (caching und authorative).

Konfigurationsdateien von Bind 9:

- Globale Konfigurationsdatei (`/etc/named.conf`)
- Zonendateien (1 pro Zone)
- ev. weitere Dateien (*.key, ...), die includiert werden.

Konfiguration - named.conf

“named.conf” besteht aus:

- Globalen Optionen: Zugriffsberechtigungen, Krypto-Keys und weitere Optionen
- (ev.) Server-Liste: Informationen über Partner-Server
- Zoneliste: ein Eintrag / Zone

named.conf - Caching only Nameserver

Beispiel /etc/named.conf

```
options {
  directory "/var/named"; /* Working directory */
  forwarders {129.27.2.3;129.27.3.3;} // Provider NS
};
zone "." { # Infos ueber Root-Nameserver. Download:
           # ftp://ftp.internic.net/domain/named.root
           # ftp://208.77.188.26/domain/named.root
        type hint;
        file "named.root";
};
// Reverse mapping der Loopback-Addresse 127.0.0.1
zone "0.0.127.in-addr.arpa" {
  type master;
  file "localhost.rev";
};
```

Ressource Records oder: was steht in den Zonefiles

`<owner name> [<ttl>] <class> <type> <rdata>`

- `<owner name>`: Name des Records. Abkürzung mit `$ORIGIN`
- `<ttl>`: Time to live – Gültigkeitsdauer
- `<class>`: **IN**(ternet), CH(aosnet) (MIT), HS (Hesiod (MIT))
- `<type>`: Recordtyp[1], z.B.
  - SOA - Record (Start of Authority)
  - NS - Records (Nameserver)
  - A - Records (Zuordnung Name $\longrightarrow$ IP(v4)-Adresse)
  - MX - Records (Mail Exchanger)
  - PTR - Record (Zuordnung IP-Adresse $\longrightarrow$ Name)
- `<rdata>`: weitere Daten (Recordtyp-spezifisch)

[1] Anmerkung: das sind die wichtigsten Record-Typen. Im Bind 9 Administrator Reference manual sind (dzt.) 33 verschiedene Typen aufgelistet...

Ein erstes Beispiel - die Zonendatei für localhost:

```
$TTL    604800          ; Time to live
// SOA Record
@       IN      SOA     localhost. root.localhost. (
                              1    ; Serial
                         604800    ; Refresh
                          86400    ; Retry
                        2419200    ; Expire
                         604800 )  ; Negative Cache TTL
// NS Record
@       IN      NS      localhost.
// A Record
@       IN      A       127.0.0.1
```

SOA Record

Name	der Zone (abgekürzt durch “@”)
TTL	(optional) wie lange darf dieser Eintrag gecached werden
IN	Klasse; üblicherweise **IN**ternet.
SOA	Recordtyp: **SOA** Record
Primary	Primary Nameserver für diese Zone
Mailaddr.	des Verantwortlichen für diese Zone (“@” $\Rightarrow$ “.”)
Serienr.	wird bei jeder Änderung inkrementiert (JJJJMMTTnn)
Refresh	Intervall in dem die Slaves anfragen, ob sich etwas geändert hat
Retry	Intervall in denen ein Slave die Anfrage wiederholt, falls sein Master nicht antwortet
Expire	falls der Master auf einen Zonentransfer-Request nicht reagiert, deaktiviert ein Slave nach dieser Zeitspanne die Zone
TTL	negativ-Caching-TTL (Caching der Info, daß Eintrag nicht existiert)

A Record: Zuordnung DNS-Name ⇒ IP(v4)-Adresse

TTL (optional) gibt an [in Sekunden], wie lange dieser Resource Record in einem Cache gültig sein darf

IN Klasse. (Internet)

A Recordtyp: A-Record (Adress)

IP IP(v4) Adresse

Beispiel A-Record

```
www.example.com.    3600   IN   A   192.0.2.1
```

A Records: Lastverteilung per DNS.

Es dürfen mehrere A-Records zu einem Namen existieren, diese werden in wechselnder Reihenfolge zurückgeliefert.

Beispiel Lastverteilung per DNS

```
www.example.com.    3600   IN   A   192.0.2.1
www.example.com.    3600   IN   A   192.0.2.2
www.example.com.    3600   IN   A   192.0.2.3
```

Dadurch ist eine einfache Lastverteilung auf mehrere Server möglich.

NS Record: Definition der Nameserver

TTL (optional) gibt an, wie lange dieser RR in einem Cache gültig sein darf

IN Internet

NS

Server Name des für diese Domäne autoritativen Nameservers

Beispiel: Nameserver (NS) Record

```
example.com.   1800   IN   NS   ns1.provider.com.
example.com.   1800   IN   NS   ns2.provider.com.
```

Es *müssen* Namen angegeben werden - keine IP-Adressen.

NS Record - Zonendelegation.

Zonendelegation:

Beispiel: Zonendelegation an andere Nameserver

```
subdomain.example.com.   IN   NS   ns1.provider2.com.
subdomain.example.com.   IN   NS   ns2.provider2.com.
```

Damit ist für die Auflösung von

`irgendwas.subdomain.example.com.`

nicht mehr der Nameserver `ns1.provider.com.` sondern `ns1.provider2.com./ns2.provider2.com.` zuständig.

Glue-Records

Nameserver der FH Joanneum

```
$ host -t ns fh-joanneum.at
fh-joanneum.at name server dallas.fh-joanneum.at.
fh-joanneum.at name server denver.fh-joanneum.at.
```

Problem:

- Die Katze beisst sich in den Schwanz:
- Zuständig für die Auflösung (IP!) von `dallas.fh-joanneum at.` sind die Nameserver von `fh-joanneum.at.`
- ...und um deren IP zu erfahren fragen wir am besten den Nameserver von `fh-joanneum.at`,
- also z.B. `dallas.fh-joanneum at...`

Lösung: Glue-Records.
Der A-Record (die IP-Adresse) für `dallas.fh-joanneum.at` ist zusätzlich(!) in der übergeordneten Zone (`at`) eingetragen.

CNAME, Wildcard, TXT

CNAME: Aliases im DNS

```
www     1800   IN CNAME server.example.com.
```

CNAMEs verursachen manchmal Schwierigkeiten (können z.B. nicht mit anderen Recordtypen zugleich existieren, sind als Mailexchanger (MX) nicht zugelassen) – eher vermeiden).

Wildcards

```
*.example.com. IN CNAME server.example.com.
```

TXT - ein frei definierbarer Text

```
               IN TXT "Hello World"
```

Wird z.B. verwendet für SPF (Sender policy framework):

```
               IN TXT "v=spf1 ip4:12.34.56.78 -all"
```

Email - MX Records

Beispiel: MX-Records - Definition der Mailserver

```
example.com      1800   IN MX 10 mail.example.com.
example.com      1800   IN MX 20 mail.backupdomain.com.
```

Ein oder mehrere für die Domain zuständige Mailserver. Zusätzlich eine Priorität – die niedrigere wird zuerst probiert.
Falls kein MX vorhanden ist, versucht der Mailserver den A Resource Record (die IP-Adresse) der Domain festzustellen. Falls der Mailserver die IP-Adresse ermitteln kann, versucht er eine SMTP-Verbindung zu dieser IP aufzubauen.

Ein Real-world example

Master-Server

```
zone "linux-tage.de" {
     type master;
     file "master/linux-tage.de";
     notify yes;
};
```

Slave-Server

```
zone "linux-tage.de" {
     type slave;
     file "slave/linux-tage.de";
     masters { 88.198.17.232; }; // IP vom master
};
```

Ein Real-world example II

```
$ORIGIN linux-tage.de.
@ 3205 IN SOA easy.in-chemnitz.de. (
              postmaster.linux-tage.de.
              2006011911 ; serial number
              3h         ; refresh (3 Std. = 10800sec)
              1h         ; retry (1 Stunde = 3600sec)
              31d6h      ; expire (31d6h = 2700000sec)
              1h )       ; neg. cache TTL (1h=3600sec)

              86382   IN   NS  robotns3.second-ns.com.
              86382   IN   NS  ns1.first-ns.de.
              86382   IN   NS  robotns2.second-ns.de.
              86400   IN   MX  10 mailhost1.in-chemnitz.de.
              86400   IN   MX  100 mailhost2.in-chemnitz.de.
              86400   IN   A   88.198.147.212
chemnitzer    86400   IN   A   34.109.133.7
www           86400   IN   A   88.198.147.212
```

Reverse DNS: Zuordnung IP ⇒ Name

Gegeben: IP Adresse.
Gesucht: Der Name des Servers (meist nicht eindeutig (virtuelle Hosts, ...)), der sich dahinter verbirgt.
Dazu gibt es Subdomains der **in-addr.arpa**-Domain. Die Zone `10.in-addr.arpa` enthält die IP-Adressen von 10.x.y.z, ...

Beispiel: PTR Record

```
1.0.0.10.in-addr.arpa.    IN PTR server1.example.com.
```

Korrespondierender A-Record-Eintrag der Domain example.com:

```
server1.example.com.      IN   A    10.0.0.1
```

Beispiel: Delegiert das Subnetz `10.0.1.XXX` an `ns1.example.com.`

```
1.0.10.in-addr.arpa.    IN   NS   ns1.example.com.
```

Nachteil: Das funktioniert (einfach) nur an 8-Bit-Grenzen.

Dynamische Updates

Werden im Zonefile erlaubt.

Beispiel: Update von IP 192.0.2.3 erlaubt

```
zone "update1.example.com" {
        type master;
        file "update1.example.com";
        allow-update { 192.0.2.3 ; } ;
};
```

Beispiel: Update mit dem Key `keyfile.example.com` erlaubt

```
zone "update2.example.com" {
        type master;
        file "update2.example.com";
        allow-update { key keyfile.example.com ; } ;
};
```

Dynamische Updates II

Updates können dann mit dem Befehl `nsupdate` durchgeführt werden.

Beispiel: Update mit dem Key `keyfile.example.com`

```
# nsupdate -k keyfile.example.com
> update delete a.update2.example.com A
> update add new.update2.example.com 86400 A 1.2.3.4
>
#
```

Problem: Replay-Attacke.

IPv6

- AAAA-Record: Name ⇒ IPv6-Adresse

Beispiel: Zuordnung einer IPv6-Adresse

```
www.example.com 3600 IN AAAA 2001:db8::1
```

- PTR-Record: IPv6-Adresse ⇒ Name.

Beispiel: Reverse-Zuordnung einer IPv6-Adresse

```
b.5.1.d.3.3.e.f.f.f.7.6.0.6.2.0.0.0.0.0.0.0.0.0. \\
0.0.0.0.0.3.8.e.f.ip6.int IN PTR www.example.com.
```

(= 32 4-Bit-Gruppen (“Nibbles”) (also 128Bit) werden durch “.” getrennt...)

Es gibt div. andere Ansätze (A6-Records, Bitstrings, ...)

SSHFP - SSH Hostkeys im DNS

(Aktuelle Bind (ab 9.4) und OpenSSH Versionen notwendig)

SSHFP-Records aus SSH-Hostkeys erzeugen

```
$ ssh-keygen -r host.example.com. -f /etc/ssh/ssh_host_dsa_key.pub
host.example.com. IN SSHFP 2 1 0f9380b2b476b770697cf0e8ac62fa1e5fb6de02
$ ssh-keygen -r host.example.com. -f /etc/ssh/ssh_host_rsa_key.pub
host.example.com. IN SSHFP 1 1 d91af427f2d29f54c85ce77ed5a1cd0fd69f35a2
```

Diese Records in DNS Zone example.com aufnehmen.

Genutzt wird dieser Record nur, wenn in einer `ssh_config` die Option VerifyHostKeyDNS auf `yes` oder `ask` gesetzt wurde. Ohne DNSSEC bedeutet `yes` auch nur `ask`, da der Fingerprint *prinzipiell unsicher* übertragen wurde. Der Fingerprint wird jedoch überprüft und stellt evtl. eine kleine Entscheidungshilfe dar.

SSHFP - SSH Hostkeys im DNS II

Host-Keys aus DNS verwenden

Entweder `VerifyHostKeyDNS=yes` in die `ssh_config` reinschreiben oder direkt auf der Kommandozeile angeben:

```
$ ssh root@host.example.com -o VerifyHostKeyDNS=yes
The authenticity of host 'host.example.com (10.0.0.1)' can't be established.
RSA key fingerprint is ec:12:1a:ae:85:a6:d4:b8:9c:d2:17:42:1c:c4:be:f0.
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host.example.com,10.0.0.1' (RSA) to the list of known hosts.
root@host.example.com's password:
```

$GENERATE-Direktive - viele ähnliche Einträge

Erweiterung des Bind9 Master Zonefiles; nicht bei allen Recordtypen möglich

Beispiel $GENERATE

```
; DHCP-Range 192.168.1.0/24
$GENERATE 1-254 dhcp$ A 192.168.1.$
```

ist äquivalent zu:

```
dhcp1    A 192.168.1.1
dhcp2    A 192.168.1.2
[...]
dhcp254 A 192.168.1.254
```

...ist eine Erweiterung des Bind MASTER Zonefiles – zum Slave (und zu anderen Nameservern) werden alle Records einzeln transferiert.

Master/Slave-Kommunikation

Alt: Slave holt sich die aktualisierte Konfiguration in definierten Zeitabständen (Refresh, Retry aus SOA-Record)

Neu: Slave wird durch den Master automatisch benachrichtigt (notify), wenn sich in der Zone was geändert hat.

- AXFR: vollständiger Zonetransfer
- IXFR: inkrementeller Zonetransfer

Wichtig: Serial-Number erhöhen!

Access control lists

Beispieldefinitionen der Netze

```
acl rfc1918  { 10/8; 172.16/12; 192.168/16; };
acl test-net { 192.0.2.0/24; }; // RFC3330: Test-net
```

Beispiele für Einschränkungen

```
options {
  allow-query      { rfc1918; }; // Abfrage erlaubt
  allow-recursion { rfc1918; }; // rekursive Abfrage ok
  allow-transfer   { rfc1918; }; // Zonetransfer erlaubt
  blackhole        { test-net; }; // nichts erlaubt
};
zone "example.com" {
  type master;
  file "master/example.com";
  allow-query { any; }; //authorativ für diese Zone
};
```

Administrationstool rndc

```
$ rndc
Usage: rndc [-c config] [-s server] [-p port]
        [-k key-file ] [-y key] [-V] command

command is one of the following:

  reload         Reload configuration file and zones.
  reload zone [class [view]]
                 Reload a single zone.
  refresh zone [class [view]]
                 Schedule immediate maintenance for a zone.
  retransfer zone [class [view]]
                 Retransfer a single zone without checking serial number.
  freeze         Suspend updates to all dynamic zones.
  freeze zone [class [view]]
                 Suspend updates to a dynamic zone.
  thaw           Enable updates to all dynamic zones and reload them.
  thaw zone [class [view]]
                 Enable updates to a frozen dynamic zone and reload it.
  notify zone [class [view]]
                 Resend NOTIFY messages for the zone.
  reconfig       Reload configuration file and new zones only.
  stats          Write server statistics to the statistics file.
  querylog       Toggle query logging.
```

Administrationstool rndc II

```
  dumpdb [-all|-cache|-zones] [view ...]
                Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  stop -p       Save pending updates to master files and stop the server
                reporting process id.
  halt          Stop the server without saving pending updates.
  halt -p       Stop the server without saving pending updates reporting
                process id.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
  flushname name [view]
                Flush the given name from the server's cache(s)
  status        Display status of the server.
  recursing     Dump the queries that are currently recursing (named.recursing)
  validation newstate [view]
                Enable / disable DNSSEC validation.
  *restart      Restart the server.

* == not yet implemented
Version: 9.4.2
```

Administrationstool rndc – rdnc-confgen

Zugriff muss konfiguriert werden, z.B. mit:

```
$ rndc-confgen
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "egvMd3tPSueuHALgE7psAg==";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#       algorithm hmac-md5;
#       secret "egvMd3tPSueuHALgE7psAg==";
# };
#
# controls {
#       inet 127.0.0.1 port 953
#               allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

Security

- nicht als Root laufen lassen:

```
groupadd named
useradd -m -d /var/named -g named -s /bin/false named
chown -R named.named /var/named
named -u named # run as user named
```

- Im Changeroot-Käfig laufen lassen (`named -t directory`).
- Zugriffseinschränkungen mittels ACLs. (`allow-query`, `allow-transfer`, ...
- aktuelle BIND-Version verwenden.
- (Wer glaubt, daß BIND unsicher ist und andere DNS-Server besser sind):
  - einen anderen DNS-Server verwenden.
  - (aber der sitzt grad im falschen Vortrag...) ☺

Links

- ISC Bind Homepage: `http://www.isc.org/sw/bind/`
- Bind (9.4) Adminstrator Reference Manual:
  `http://www.isc.org/sw/bind/arm94/Bv9ARM.pdf`
- Log messages for BIND:
  `http://www.menandmice.com/knowledgehub/bindlogmsgs`
- DNS/BIND/named error messages and problems:
  `http://www.reedmedia.net/misc/dns/errors.html`
- DNS Ressources Directory: `http://www.dns.net/dnsrd/`
- DNS related RFCs: `http://www.dns.net/dnsrd/rfc/`

Links II

Online- und offline DNS-Test-Tools

- DNS Sleuth: `http://atrey.karlin.mff.cuni.cz/~mj/sleuth/`
  An online tool for checking of DNS zones
- DNS Report: `http://www.dnsreport.com/`
  Nameserverkonfiguration online testen.
- Zonecheck: Softwaredownload auf `http://www.zonecheck.fr/`,
  Online-Test auf `http://zonecheck.denic.de/`

Fragen?

Ausblicke – was ich nicht behandelt habe...

- weitere Ressource-Records
- Views: unterschiedliche Antworten, je nach anfragender IP (intern, extern)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- DNS wird ständig weiterentwickelt.

Vielen Dank für Ihre Aufmerksamkeit

Wolfgang Dautermann
wolfgang.dautermann@fh-joanneum.at