DNS mit Bind 9

Wolfgang Dautermann

FH JOANNEUM

Grazer Linuxtage 2006

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 9.3.2 (ISC)

Hierarchische Struktur

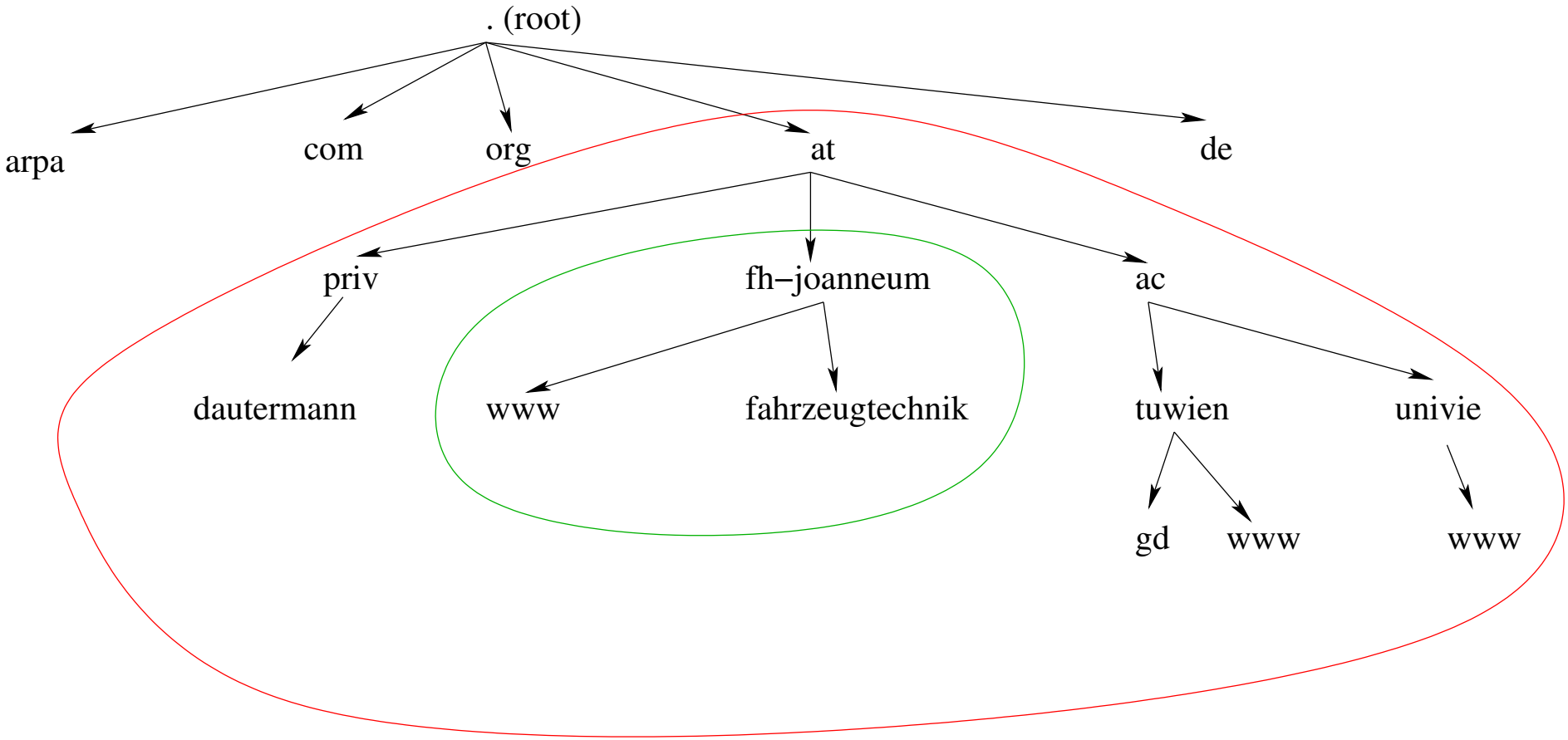

Abbildung: hierarchische Domain-Struktur

Hierarchische Struktur - Zuständigkeiten

- "." wird verwaltet von ICANN (Festlegung Top-Level-Domains, Betrieb der 13 Root Nameserver.

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).

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 Österreicher, für nichtkommerziellen Gebrauch). delegiert an und verwaltet von www.nic.priv.at (Verein www.vibe.at)

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 Österreicher, 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.

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

Nameserver-Records für eu (neu!)

```
$ host -t ns eu.
eu name server m.nic.eu.
eu name server p.nic.eu.
eu name server a.eu.dns.be.
eu name server b.eu.dns.be.
eu name server l.eu.dns.be.
eu name server l.nic.eu.
```

Nameserver-Records für linuxtage.at

```
$ host -t ns linuxtage.at
linuxtage.at name server ns1.pronet.at.
linuxtage.at name server ns2.pronet.at.
```

Installation

- `rpm`, `yast`, `apt-get`, `pkg-get`, ... oder:
- Download der aktuellen Version (dzt. 9.3.2) von http://www.isc.org/sw/bind/ (Dabei sind u.a. relevante RFCs, B9vARM (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`, `dnssec-keygen`, ...)

Konfiguration

Konfiguration von Bind 9 als:

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

Konfigurationsdateien von Bind9:

- 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

```
options {
  directory "/var/named"; /* Working directory */
  forwarders {129.27.2.3; 129.27.3.3;} # Provider
};

zone "." {       // Infos ueber Root-Nameserver
        type hint;
        file "db.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

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

- SOA - Record (Start of Authority)
- NS - Records (Nameserver)
- A - Records (Zuordnung Name $\rightarrow$ IP(v4)-Adresse)
- AAAA - Records (Zuordnung Name $\rightarrow$ IP(v6)-Adresse)
- CNAME - Records (Aliases)
- MX - Records (Mail Exchanger)
- PTR - Record (Zuordnung IP-Adresse $\rightarrow$ Name)

(Anmerkung: das sind die wichtigsten Record-Typen.
Im Bind 9 Administrator Reference manual sind (dzt.) 29 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) gibt an, wie lange dieser Eintrag im Cache gehalten werden darf
IN	Klasse; üblicherweise INternet.
SOA	Recordtyp: SOA Record
Primary	Primary Nameserver für diese Zone
Mailaddr.	des Verantwortlichen für diese Zone ("@" → ".")
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 wiederholt, falls sein Master nicht antwortet
Expire	falls Master auf einen Zonentransfer-Request nicht reagiert, deaktiviert ein Slave nach dieser Zeitspanne die Zone
TTL	negativ-Caching-TTL

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

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[1] zurückgeliefert.

Lastverteilung - mehrere Adress-Records

```
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 (round-robin) Lastverteilung auf mehrere Server möglich.

[1] Genauer: Es werden immer alle Records (in wechselnder Reihenfolge) zurückgeliefert, und alle bis auf die erste Antwort ignoriert...

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 NS-Record

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

Es *müssen* Namen angegeben werden - keine IPs.

NS Record - Zonendelegation.

Zonendelegation:

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

NS Record - Zonendelegation.

Zonendelegation:

```
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 (z.B.) `ns1.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.
```

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:

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 DNS-Server von `fh-joanneum.at.`

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 DNS-Server von `fh-joanneum.at.`
- (und um deren IP zu erfahren fragen wir am besten den DNS-Server von `fh-joanneum.at`,

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 DNS-Server von `fh-joanneum.at.`
- (und um deren IP zu erfahren fragen wir am besten den DNS-Server von `fh-joanneum.at`,
- also z.B. `dallas.fh-joanneum at.`...

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 DNS-Server von `fh-joanneum.at.`
- (und um deren IP zu erfahren fragen wir am besten den DNS-Server von `fh-joanneum.at`,
- also z.B. `dallas.fh-joanneum at.`...

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 DNS-Server von `fh-joanneum.at.`
- (und um deren IP zu erfahren fragen wir am besten den DNS-Server 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

Ein CNAME ist ein Alias. Beispiel:

Beispiel CNAME-Records

```
www                1800   IN CNAME server
```

Wildcards:

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

TXT - ein frei definierbarer Text:

Beispiel TXT-Records

```
                   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

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

- Für die Domain zuständige Mailserver
- ev. mehrere - mit Priorität (die niedrigere zuerst)
- Wenn kein MX $\Rightarrow$ Zustellversuch an IP (falls A-Record existiert)

Ein Real-world example

Master-Server

```
zone "linuxtage.at" {
     type master;
     file "master/linuxtage.at";
     notify yes;
};
```

Slave-Server

```
zone "linuxtage.at" {
     type slave;
     file "slave/linuxtage.at";
     // IP vom linuxtage.at master server
     masters { 213.187.64.101; };
};
```

Ein Real-world example II

```
$ORIGIN linuxtage.at.
@ 3205 IN SOA ns1.pronet.at. (
                domainmaster.pronet.at.
              2006040600 ; serial number
              339940     ; refresh (339940 sec)
              4h         ; retry (4 Std. = 14400 sec)
              1w         ; expire (1 week = 604800 sec)
              1d )       ; neg. cache TTL (1 day)

              IN  NS     ns2.pronet.at.
              IN  NS     ns1.pronet.at.
              IN  MX     mail.linuxtage.at.
              IN  A      81.223.126.153
mail          IN  A      81.223.126.153
www           IN  A      81.223.126.153
glt06         IN  A      81.223.126.153 ; oder: CNAME www
glt05         IN  A      81.223.126.153
glt04         IN  A      81.223.126.153
glt03         IN  A      81.223.126.153
```

Reverse DNS: Zuordnung IP ⇒ Name

Ich kenne eine IP Adresse. Wie heisst der Server, der sich dahinter verbirgt? (Antwort meist nicht eindeutig (virtuelle Hosts, ...)).
Dazu gibt es Subdomains der **in-addr.arpa**-Domäne. Die Zone `10.in-addr.arpa` enthält die IP-Adressen von 10.x.y.z, ...

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

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

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

Delegation:

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

delegiert das Subnetz `10.0.1.XXX` an `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

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

IPv6

- AAAA-Record: Name ⇒ IPv6-Adresse
  `www.example.com 3600 IN AAAA 2001:db8::1`
- PTR-Record: IPv6-Adresse ⇒ Name.

IPv6

- AAAA-Record: Name ⇒ IPv6-Adresse
  `www.example.com 3600 IN AAAA 2001:db8::1`
- PTR-Record: IPv6-Adresse ⇒ Name.

  `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. . .)

IPv6

- AAAA-Record: Name ⇒ IPv6-Adresse
  `www.example.com 3600 IN AAAA 2001:db8::1`
- PTR-Record: IPv6-Adresse ⇒ Name.

  `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-Record,Bitstrings, ...)

Views

View-Beispiel: Unterschiedliche Antworten je nach Client

```
view "internal" {
    match-clients { 10.0.0.0/8; }; // internes Netz
    // komplette Zone (inkl. interne hosts)
    zone "example.com" {
        type master;
        file "example.com-intern";
    };
};
view "external" {
    match-clients { any; };
    zone "example.com" { // eingeschraenkte Zone
        type master;
        file "example.com-extern";
    };
};
```

Master/Slave-Kommunikation

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

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 gendert hat.

- AXFR: vollständiger Zonetransfer
- IXFR: inkrementeller Zonetransfer

Wichtig: Serial-Number erhöhen!

Access control lists

Beispieldefinitionen der Netze

```
acl netz1 { 192.168.0.0/16; };
acl test-net { 192.0.2.0/24 ; }; // RFC3330: Test-net
```

Beispiele für Einschränkungen

```
options {
    allow-query { netz1; }; // Abfrage erlaubt
    allow-recursion { netz1; }; //rekursive Abfrage ok
    allow-transfer { netz1; }; // Zonetransfer erlaubt
    blackhole { test-net; };  // nichts erlaubt
};
zone "example.com" {
    type master;
    file "master/example.com";
    allow-query { any; }; //authorativ fur 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 zone [class [view]]     Suspend updates to a dynamic zone.
 thaw zone [class [view]]       Enable updates to a frozen dynamic zone and reload it.
 reconfig       Reload configuration file and new zones only.
 stats          Write server statistics to the statistics file.
 querylog       Toggle query logging.
 dumpdb         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 pid.
 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)
 *restart       Restart the server.

* == not yet implemented
Version: 9.3.1
```

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
```

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`).

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`, ...

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.

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):

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.

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...) ☺

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.
  - `127.0.0.XXX` irgendwie geblacklistet, z.B.

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.
  - `127.0.0.XXX` irgendwie geblacklistet, z.B.
  - `127.0.0.5`: Offenes Mailrelay,

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.
  - `127.0.0.XXX` irgendwie geblacklistet, z.B.
  - `127.0.0.5`: Offenes Mailrelay,
  - `127.0.0.6`: Spammer, ...

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.
  - `127.0.0.XXX` irgendwie geblacklistet, z.B.
  - `127.0.0.5`: Offenes Mailrelay,
  - `127.0.0.6`: Spammer, ...
- ENUM: tElephone NUmber Mapping Abbildung von Telefonnummern im DNS.
  Aus der Telefonnummer: `+43 316 12345678` wird
  `8.7.6.5.4.3.2.1.6.1.3.3.4.e164.arpa.`
  Spezielle DNS Records (NAPTR-Records) definieren Services (SIP, Mailto,...)

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Weitere Anwendungen von DNS

- DNS-Blacklists[2]: (Spam, etc.)
  `$ host 10.2.0.192.dnsbl.sorbs.net # teste IP 192.0.2.10`
  - Kein Resultat: nicht geblacklistet.
  - `127.0.0.XXX` irgendwie geblacklistet, z.B.
  - `127.0.0.5`: Offenes Mailrelay,
  - `127.0.0.6`: Spammer, ...
- ENUM: tElephone NUmber Mapping Abbildung von Telefonnummern im DNS.
  Aus der Telefonnummer: `+43 316 12345678` wird
  `8.7.6.5.4.3.2.1.6.1.3.3.4.e164.arpa.`
  Spezielle DNS Records (NAPTR-Records) definieren Services (SIP, Mailto,...)
- Tunneln div. Services über DNS.

[2] Beispiel sorbs.net: `http://www.nl.sorbs.net/using.shtml`

Links

- ISC Bind Homepage: `http://www.isc.org/sw/bind/`
- Bind 9 Adminstrator Reference Manual:
  `http://www.nominum.com/content/documents/bind9arm.pdf`
- Log messages for BIND:
  `http://www.menandmice.com/docs/named_messages.htm`
- DNS Sleuth: `http://atrey.karlin.mff.cuni.cz/~mj/sleuth/`
  An online tool for checking of DNS zones
- DNS Report: `http://www.dnsreport.com/`
- DNS Ressources Directory: `http://www.dns.net/dnsrd/`
- DNS related RFCs: `http://www.dns.net/dnsrd/rfc/`

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- Resolver: iterativ vs. rekursiv

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- Resolver: iterativ vs. rekursiv
- DNS wird ständig weiterentwickelt.

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- Resolver: iterativ vs. rekursiv
- DNS wird ständig weiterentwickelt.

Fragen?

Ausblicke - was ich nicht behandelt habe...

- weitere Ressource-Records
- `$GENERATE`-Direktive (für viele ähnliche DNS-Einträge)
- Security: TSIG (sym. Verschlüsselung), DNSSEC (asym.Verschlüsselung)
- Resolver: iterativ vs. rekursiv
- DNS wird ständig weiterentwickelt.

Vielen Dank für Ihre Aufmerksamkeit

Wolfgang Dautermann
wolfgang.dautermann@fh-joanneum.at