DNS (Domain Name Service)

- Kurzakte –

Server:

Installation	
• Paket: bind (nach /usr/sbin/named) BIND= Berkeley Internet Name Domain	
Konfiguration	
/etc/named.conf (ab bind8) /etc/named.boot (bis bind7)	• zentrale Konfigurationsdatei = Startdatei (von S.u.S.E. vorgefertigt) • globale Optionen • Definition von Zonen für forward- und reverse-lookup (master, slave, hint) • Kommentarzeichen: # oder /* Text */ oder // bis Zeilenende
/var/lib/named/root.hint	• enthält Adressen aller Root-Server weltweit • erstellbar mit: " dig @a.root-servers.net > root.hint"
/var/lib/named/localhost.zone	• Auflösung von localhost zu 127.0.0.1 (lookup)
/var/lib/named/127.0.0.zone	• Rückwärtsauflösung von 127.0.0.1 zu localhost (reverse lookup)
/var/lib/named/<domain>.zone	• Datei der Zone für das forward lookup (Auflösung Namen zu IP-Adressen) • als Kommentarzeichen ist hier nur das ";" erlaubt!!! • Bspe. in: /usr/share/doc/packages/bind/sample-config
/var/lib/named/<net-id>.zone	• Rückwärtsauflösung von IP-Adressen zu Rechnernamen
/var/lib/named/domain.zone.jnl	• Datenbank für dynamisches DNS-Update via DHCP bei Windows-Clients • abgelaufene DHCP-Leases werden aus dieser Zonendatei gelöscht • Achtung: dazu muss named Schreibrecht im directory besitzen!!!
/var/run/named/named.pid	• enthält PID des Serverprozesses (kill –HUP `cat /var/…/named.pid)
/etc/sysconfig/dhcpd (bei SUSE)	• per Hand editieren: NAMED_RUN_CHROOTED="no"

häufigste Einträge in den Zonendateien		
Aufbau eines DNS-Datensatzes (Resource Record): [domain] [ttl] [class] type data		
Typ	**Bedeutung**	**Eintrag in den Zonendateien**
SOA	Start Of Authority	Einleitungssatz für Daten einer Zone
MX	Mail Exchange	Die Priorität (kleinste Nummer) und Name des Mailservers der Domain
NS	Name Server	Verweis auf den autorisierten Nameserver der Domain
A (A6)	Address	Zuordnung einer IP-Adresse zu einem kanonischen Hostnamen (FQDN)
CNAME	Canonical Name	Alias für einen kanonischen Hostnamen (www, mail, ftp – leichter merkbar)
PTR	Pointer	Zuordnung eines Hostnamens zur numerischen IP-Adresse für reverse lookup
HINFO	Host Information	ASCII Beschreibung des Hosts (CPU, OS, ...) nach RFC 1340
TXT	Text	Nicht verwertbarer Text – Kommentar HINFO und TXT müssen bei Hosts *unmittelbar nach* dem A-Record folgen!!!
SRV	Services	Hinweis auf verfügbare Services
Aufbau des SOA-Datensatzes (Start of Authority = erster Datensatz einer Zonendatei)		
TTL	(optional) gibt an, wie lange der Resource Record im Cache gehalten werden darf	
@	current origin (auch $ORIGIN) enthält den Namen der Zone-Direktive aus der /etc/named.conf	
domain.	statt dem @ kann hier auch der Domainname stehen. Der "." schließt den vollständigen Namen ab. Ohne Punkt wird der Domainname automatisch von bind angehängt.	
IN	Klasse IN für Internet-Adresse	
serial	fortlaufende Zahl zur Feststellung von Änderungen an Zonendatei des Master-Servers (JJJJMMTTnn)	
refresh	nach Ablauf dieser Zeit holt sich Slave-Server die Zonendaten erneut vom Server (sec o. M, H, D, Week)	
retry	bei Fehlschlag erfolgt nach dieser Zeit erneuter Versuch zum Abholen der Zonendaten	
expiry	nach dieser Zeit werden alle Zonendaten verworfen, wenn Masterserver nie erreicht wurde	
minimum	Verweildauer eines fehlgeschlagenen Requests im Cache des Slave, sofern RR nicht selbst TTL festlegt	

Start als Stand-Alone-Dienst	
insserv named (SuSE)	• automatisches Einfügen in die Netzwerk-Runlevel • bewirkt automatischen Start von named beim Hochfahren
/etc/init.d/named	start | stop | restart | reload | status - Skript startet/stoppt DNS manuell

Konfigurationstest Nameserver	
named -g	• Kontrolle des Names-Dämon BIND (Abbruch: ^C)
rndc status	• Diagnosetool über einen Domain-Socket (löst Prozess-Signal-Diagnose ab)
netstat -nl	• Test, ob Port 53 geöffnet ist
named-checkconf/checkzone	• Prüfen der named.conf und der Zonendateien

Logdateien	
/var/log/messages	• dokumentiert Start und Laden der Zonendateien sowie Konfigurationsfehler

Dokumentation
man: named (8), named.conf (5), hosts (5), resolver (5), hostname (7), Internet RFC 952, RFC 2308 (BIND)

Client für Linux:

Konfiguration	
/etc/hosts	• es können alle Einträge bis auf 127.0.0.0 entfallen
/etc/resolv.conf	• Eintrag der Nameserver für die jeweilige Zone (max. 3) (nameserver IP-Adresse) • Angabe von Domainnamen zum Anhängen an einfache Rechnernamen (search Domainname1 [Domainname2]) - Bsp.: aus "ping host" wird "ping host.domain"
/etc/nsswitch.conf	• moderne Konfigurationsdatei für Programme gelinkt gegen glibc • Festlegung, woher der Rechner seine Namensinfos bezieht • mögliche Quellen: NIS DNS lokale Dateien • Vorhandensein der Zeile: „hosts: files dns“ → erst host-Datei dann DNS laut resolv.conf-Vorgaben
/etc/host.conf	• nur noch zur Rückwärtskompatibilität für libc4/5-Programme • wird berücksichtigt, wenn keine nsswitch.conf vorhanden • bestimmt die Art der Namensauflösung - order hosts, bind → erst host-Datei, dann DNS - multi on → Auswertung aller Adressen eines Hosts
Test der Nameserver-Konfiguration	
ping <FQDN>	• testet Erreichbarkeit über Host-Namen
nslookup	• interaktive Namensabfrage set type=A (Standard) → Auflösung in IP-Adresse set type=PTR → Auflösung in Namen set type=HINFO → Hostinfos set q=ns → Namesersuche
nslookup <eigene IP>	• Anfrage an Nameserver der resolv.conf
nslookup <Host-Name> <Server>	• s. o., Angabe des Servers, wenn anderer NS als in der resolv.conf befragt
dig <Host-Name>	• s. o.
host <Host-Name>	• s. o.
uname –n	• Abfrage des kanonischen Namens (offizieller Hostname)

Client für Windowst:

- In den Eigenschaften für TCP/IP die DNS-Serveradressen eintragen
- Test: w. o.

Aufbau eines DNS-Datensatzes (Resource Record = RR) – aus Einträgen der Zonendateien erstellt

[domain] [ttl] [class] type data

domain	Name der Domäne laut Konfigurationsdatei
TTL	Zeitdauer, die der RR in einem entfernten Cache verweilen darf, dann Neuanforderung
class	beschreibt den Netzwerktyp (IN für Internet)
type	kennzeichnet die Art der enthaltenen Daten (A, CNAME, HINFO, …)
data	Daten

Beispiel:

zurquelle.de. 800 IN SOA master.zurquelle.de. mail.zurquelle.de. 411 3600 1800 604800 1800

- Slave synchronisiert sich alle 3600 Sekunden mit seinem Master für die Zone zurquelle.de
- Ist der Master nicht erreichbar, wird alle 1800 Sekunden ein neuer Versuch gestartet.
- ist Master innerhalb 604800 Sekunden (eine Woche) nicht erreichbar, erklärt der Slave die Zone zurquelle.de für inaktiv und beantwortet keine diesbezüglichen DNS-Requests mehr.
- DNS cachet auch fehlgeschlagene Request. Die TTL beträgt hierfür 1800 Sekunden.
- der Primary dieser Zone ist master.zurquelle.de
- der Administrator ist über die Mail-Adresse mail@zurquelle.de erreichbar (die DNS-Software ersetzt den ersten Punkt selbständig durch ein "@").
- der Standard (Default) Time To Live für Resource Records dieser Zone ist 800
- die Seriennummer beträgt zur Zeit 411. Bei der nächsten Änderung wird sie auf 412 erhöht werden.

Beispiele für die Konfigurationsdateien

/etc/named.conf

```
options {
       directory "/var/lib/named";
       # forwarders { bis zu 3 Nameserver des Providers; };
       # forward first;          # zuerst Anfrage über Forwarder, erst dann über Root-Server
       #listen-on port 53 { 127.0.0.1; };
       listen-on-v6 { any; };
       # allow-query { 127.0/16; 192.168.1/24; };
       notify no;                # Info an Slaves bei geänderten Zonendaten
};

zone "." in {                            # ein Eintrag für den Root-Server
       type hint;
       file "root.hint";
};

zone "localhost" in {                    # je ein Eintrag für die DNS-Zone und die reverse-Zone
       type master;                      # mit Festlegung, ob Master oder Slave
       file "localhost.zone";            # Dateinamen sind frei vergebbar
};
zone "0.0.127.in-addr.arpa" in {         # Zonenname muss zwingend die reverse-NetID enthalten
       type master;
       file "127.0.0.zone";              # Dateinamen sind frei vergebbar
};

zone "joerg.local" in {                  # je ein Eintrag für die DNS-Zone und die reverse-
       type master;                      # mit Festlegung, ob Master oder Slave
       file "master/joerg.local.zone";   # Dateinamen sind frei vergebbar
};
zone "1.168.192.in-addr.arpa" in {       # Zonenname muss zwingend die reverse-NetID enthalten
       type master;                      # für jedes Subnetz muss eine eigene Zone erstellt werden
       file "master/192.168.1.zone";     # Dateinamen sind frei vergebbar
};
```

/var/lib/named/master/joerg.local.zone

```
$TTL 2D
@ IN SOA ns1.joerg.local. postmaster.joerg.local. (
             2003071203   ; serial (12.07.2003 Version 03)
             3H           ; refresh
             15M          ; retry
             1W           ; expiry
             1D )         ; minimum
             IN NS        ns1          # Nameserver für diese Zone, darf kein Alias sein
             IN MX 0      ns1          # Mailserver auf gleichem Rechner, darf kein Alias sein
ns1          IN A         192.168.1.1
www          IN CNAME     ns1.joerg.local.
ftp          IN CNAME     ns1          # Zuweisung eines Alias-Namens
hund         IN A         192.168.1.10 # Adresszuweisung (Eintragsschlüsel:Wert)
katze        IN A         192.168.1.11 # ohne Punkt wird Domainname noch angehängt
maus.joerg.local.   IN A  192.168.1.12 # FQDN
```

/var/lib/named/master/192.168.1.zone

```
$TTL 2D
@ IN SOA ns1.joerg.local. postmaster.joerg.local. (
             2003071203 ; serial (12.07.2003 Version 03)
             3H         ; refresh
             15M        ; retry
             1W         ; expiry
             1D )       ; minimum
             IN NS      ns1.joerg.local.
1            IN PTR     ns1.joerg.local.  # fehlender Adressteil wird
10           IN PTR     hund.joerg.local. # dem SOA-Datensatz entnommen
11           IN PTR     katze.joerg.local.
12           IN PTR     maus.joerg.local.
```

Master-, Slave- und Cache-only-Server

Eine zone-Anweisung auf einem primären Nameserver mit der Domäne example.com:

```
            IN NS  master           # authoritativer Nameserver
            IN NS  slave            # Eintrag der Slave-Server
...
zone "example.com" IN {
  type master;                      # authoritative
  file "master/joerg.local.zone";
  allow-update { none; };
  allow-transfer { 192.168.1.2; };  # IP des Slave-Servers
  notify no;                        # sollte auf yes stehen zur Slave-Info
};
```

Die Slave-Server müssen zusätzlich in der Zonen-Datei des Masters als NS-Record eingetragen sein!

Diese zone-Direktive benennt die Zone example.com, stellt als type master ein und weist den named-Service an, die Datei /var/named/master/joerg.local.zone zu lesen und weist named an, und lässt keine Updates für einen anderen host zu.

Das "notify no;" bewirkt, dass keine anderen Nameserver benachrichtigt werden, wenn an den Zonendaten Änderungen vorgenommen werden oder der Nameserver neu gestartet wird.

Bei Eintrag "notify yes" werden die Slave-Server bei Änderungen informiert. (moderner als die SOA-Daten)
Wichtig: Vorher muss die Seriennummer des SOA-Records erhöht werden.

Eine zone-Anweisung für einen Slave-Server mit der Domäne example.com:

```
zone "example.com" {
  type slave;                # wird zum Slave-Server, non authoritative
  file "slave/joerg.local.zone";
  masters { 192.168.1.1; };  # IP des Master-Nameservers
};
```

Diese zone-Anweisung weist named auf dem Slave-Server an, bei dem Master-Server mit der IP 192.168.1.1 nach Informationen für die Zone example.com zu suchen. Die Informationen, die der Slave-Server vom Master-Server erhält, werden in der Datei /var/named/slave/joerg.local.zone gespeichert.

An die Stelle der Zeile allow-update tritt eine Anweisung, die named die IP-Adresse des Master-Servers mitteilt.

Eine options-Anweisung für einen Cache-only-Server:

```
options {
        ...
        forwarders { 10.0.0.1; 10.0.0.2; };
        forward first;
        ...
};
```

Cache-only-Server stellen keine eigenen Daten bereit.

"forwarders" verwendet man, um den oder die Nameserver (meist des Providers) anzugeben, an den oder die die DNS-Anfragen weitergereicht werden, die nicht direkt beantwortet werden können. Der Datenverkehr wird also über einen Link reduziert.
Damit ist die Abfrage der Root-Name-Server nicht mehr nötig.

"forward first" bewirkt, dass die DNS-Anfragen zu erst geforwarded werden, bevor versucht wird diese über die Root-Nameserver aufzulösen. Anstelle von forward **first** kann man auch forward **only** schreiben, dann werden alle Anfragen weitergeleitet und die Root-Nameserver werden gar nicht mehr angesprochen. Das kann für Firewall-Konfigurationen sinnvoll sein.
Für nicht authoritative Server bei nicht gecacheten Antworten .

Zonendelegation

- Über NS-RRs (Ressource-Records) können Verweise zu Subdomänen definiert werden. Die entsprechende Subdomäne wird gewissermaßen aus der Zonendatei ausgelagert. Ein derartiger NS-RR dient damit als Pointer, der auf einen anderen Nameserver verweist.
- Die Resolver-Anfragen werden zu einem anderen Nameserver delegiert.

Beispiel

Nameserver : ns1.joerg.local sei verantwortlich für die Zone joerg.local
Nameserver: ns2.subdom.joerg.local sei autoritativ zuständig für die ausgelagerte Zone subdom.joerg.local

Einträge in die Zonendatei von ns1.joerg.local

```
; autoritativer NS für Zone joerg.local
joerg.local.    IN NS           ns1.joerg.local.
ns1             IN A            192.168.1.1
; delegation records
; Delegation einer Subdomain-Zone an einen anderen authoritativen NS
; der Nameserver fuer die Subdomain:
subdom.joerg.local.     IN NS   ns2.subdom.joerg.local. ; NS-Pointer
; glue record mit der IP des Subdomain-NS, da Aufloesung nicht moeglich
ns2.subdom      IN A            192.168.1.70
; Rechner der Zone joerg.local
debian          IN CNAME        ns2.subdom      ; ns2 ist auch als debian erreichbar
bla             IN A            192.168.1.100
```

Der Ressource-Record auf ns2.subdom.joerg.local. liegt ausserhalb der Zone joerg.subdom, da ns2 in der delegierten Zone der Subdomain liegt.
Henne–Ei–Problem: ns1.joerg.local kennt die IP von ns2.subdom.joerg.local nicht, da diese in der Zone subdom.joerg.local von ns2 selbst verwaltet wird. Wohin soll man also die Anfragen an subdom.joerg.local senden?
Dazu ist der "glue record" in der übergeordneten Zone nötig, der als A-Record die IP von ns2 liefert.

jetzt ist schon die Namensauflösung für ns2 möglich (hier von einem Win-Client mit NS 192.168.1.1):

```
C:\ nslookup ns2.subdom.joerg.local
Server:  ns1.joerg.local
Address:  192.168.1.1
Nicht autorisierte Antwort:
Name:    ns2.subdom.joerg.local
Address:  192.168.1.70
```

Einträge in die Zonendatei von ns2.subdom.joerg.local

```
; autoritativer NS für Zone subdom.joerg.local
subdom.joerg.local.     IN      NS      ns2.subdom.joerg.local.
ns2     IN      A       192.168.1.70
; Rechner der Zone subdom.joerg.local
router  IN      A       192.168.1.250
maria   IN      A       192.168.1.20
paul    IN      A       192.168.1.30
```

die rekursive Namensabfrage für Rechner in der Subdomäne über ns1 an ns2 funktioniert

```
C:\ >nslookup router.subdom.joerg.local
Server:  ns1.joerg.local
Address:  192.168.1.1
Nicht autorisierte Antwort:
Name:    router.subdom.joerg.local
Address:  192.168.1.250
```

Abfrage der Nameserver

```
debian:~#  host -t ns joerg.local
joerg.local name server ns1.joerg.local.
debian:~#  host -t ns subdom.joerg.local
subdom.joerg.local name server ns2.subdom.joerg.local.
```

Tipp: eine negative Antwort von ns2 behält ns1 eine Zeitlang im Cache, behebbar mit Neustart

DHCP erzeugt dynamische Updates von DNS (RFC 2136)

In einem Netz mit Windows-Clients haben Sie das Problem zweier unterschiedlicher Namensauflösungen. Sie haben

- einerseits die Wins-Namen und
- andererseits einen Namen innerhalb der lokalen Domain.

Im Zusammenspiel mit dem DHCP-Server können beide Namensräume vereinheitlicht werden:
Wenn sich ein Windows-Client im Netz anmeldet, versucht er per DHCP eine IP-Adresse zu bekommen. Dazu übermittelt er dem DHCP-Server seine MAC-Adresse und seinen Wins-Namen.

Jan 4 17:42:55 ns1 dhcpd: DHCPDISCOVER from 00:50:bf:58:56:fd (OEMComputer) via eth0

Mit diesem Namen kann der DHCPD den Nameserver aktualisieren, wenn die Konfigurationen dies erlaubt:

1.) In der Datei **/etc/named.conf** müssen Sie die Zonen-Statements etwas erweitern, um das Update zu erlauben.

```
acl meinnetz { 192.168.1.0/24; };
zone "joerg.local" in {
type master;
file "master/joerg.local.zone";
allow-update {127.0/16; meinnetz; };
};
zone "1.168.192.in-addr.arpa" in {
type master;
file "master/192.168.1.zone";
allow-update {127.0/16; meinnetz; };
};
```

Mit der Zeile allow-update {127.0/16; 192.168/16; }; erlauben Sie dem Server selber und den Rechnern in Ihrem lokalen Netz, die Zonendateien zu aktualisieren.

2.) Nun müssen Sie noch die **dhcpd.conf** so ändern, dass der DHCPD die Zonendateien auch wirklich ändert.

```
# dhcpd.conf
ddns-update-style interim;      # interim erlaubt failover, ad-hoc nicht mehr empfohlen
ddns-domainname name;           # Domainname wird an Client-Hostnamen angehängt, Dom. für ddns
# options domain-name           # w.o., aber ddns-domainname hat Vorrang
ignore client-updates;          # die durch Client gelieferte Domäne wird ignoriert
```

An dieser Stelle stand vorher: ddns-update-style none; was das Aktualisieren unterbunden hatte. Die Veränderungen am Nameserver erfolgen nicht nur virtuell, sondern dauerhaft, der Nameserver verändert dabei die Zonendateien.

Vergabe der Schreibrechte für Benutzer named:

für Verzeichnis zum Anlegen der *.jnl-Dateien:

```
chown named /var/lib/named/master/
chmod 755 /var/lib/named/master/
```

für bestehende Zonen-Dateien:

```
chown named /var/lib/named/master/*
chmod 664 /var/lib/named/master/*
```

Test am Windows-Client:

```
ipconfig /renew         fordert IP an; bewirkt ddns-Update
      check mit nslookup <NetBIOS-Name>
ipconfig /release       gibt IP wieder frei; entfernt DNS-Eintrag wieder
      check wieder mit nslookup  → failed
```

für Linux-Clients mit nsupdate:

```
linux:~ # nsupdate
> server 192.168.1.1
> update add bla.joerg.local. 300 IN A 192.168.1.100
> send
> quit
linux:~ # nslookup bla
Name:   host3.joerg.local
Address: 192.168.1.100
```