

Domain Name System – DNS

Inhaltsverzeichnis

1. Bedeutung.....	3
2. Entstehung.....	3
3. Was macht DNS.....	3
4. Warum überhaupt DNS.....	3
5. Aufbau von DNS.....	3
5.1 Schriftliche Darstellung.....	4
5.2 Grafische Darstellung.....	5
6. Server- / Client-Bestandteile.....	5
6.1 Server: Nameserver.....	5
autoritativer Nameserver.....	5
nichtautoritativer Nameserver:.....	5
caching-only Nameserver:.....	6
6.2 Client: Resolver.....	6
rekursiver Resolver.....	6
iterativer Resolver.....	6
6.3 Grafisches Beispiel.....	7
7. Die Zonendatei.....	7
7.1 Die Zonendatei.....	7
7.2 Wichtige Resource Records.....	8
SOA-RR (Start of Authority).....	8
NS-RR (NameServer-Resource Record):.....	9
A-RR (A-Resource Record).....	9
AAAA-RR (AAAA-Resource Record, gesprochen „quad-A-RR“)......	9
CNAME-RR (CNAME-Resource Record).....	10
MX-RR (Mail Exchange Resource Record).....	10
PTR-RR (Pointer Resource Record).....	11
8. Diagnosetools.....	11
8.1 nslookup.....	11
8.2 DIG.....	11
9. Praxis-Übung.....	13
9.1 Grafik zur Praxis-Übung.....	13
9.2 Allgemeine Beschreibung.....	13
9.2.1 Installation eines Debian Squeeze 6.0 (NetInstallation).....	13
9.2.2 DHCP-Server.....	14
9.2.3 Dienste stoppen.....	14
9.2.4 Einstellungen in der named.conf.options.....	14
9.2.5 Einstellungen in der named.conf.local.....	14
9.2.6 Forward-Zonen-Dateien.....	15

9.2.7 Reverse-Zonen-Dateien.....	16
9.2.8 Zonen-Dateien in /var/cache/bind/ verlinken.....	17
9.2.9 DHCP-Konfiguration.....	17
9.2.10 Resolv.conf anpassen.....	19
9.2.11 Dienste starten.....	19
9.2.12 Erweitertes Logging.....	19
9.2.13 FAQ's.....	20
10. Quellen.....	21

1. Bedeutung

Domain Name System (kurz DNS), auf deutsch Domain-Namensauflösung

2. Entstehung

Entwickelt wurde das DNS-Konzept von Paul Mockapertis im Jahre 1983. Festgelegt sind die Spezifikationen in den RFC's 882 und 883. Diese RFC's wurde mittlerweile von der RFC 1034 und 1035 abgelöst.

Am 15.03.1985 wurde DNS in Betrieb genommen, als das Internet (damals ARPNET) seinen Durchbruch erlebte.

3. Was macht DNS

DNS dient zur Namensauflösung, d.h. das mit Hilfe dieses Diensts ein DNS-Name zur IP-Adresse (Forward-Lookup) umgesetzt werden kann und umgekehrt (Reverse-Lookup).

Beispiel:

- IPv4: www.heise.de <=> IP: 193.99.144.85
- IPv6: www.heise.de <=> 2a02:2e0:3fe:100::7

Der entscheidende Vorteil von DNS-Namen ist das zusätzliche Ansprechen von Rechner per Name und nicht nur per IP-Adresse.

Wieso ist das ein Vorteil?

Der Mensch kann sich nun einmal Namen einfacher merken als mehrere komplizierte Zahlenkombination und seit der Einführung von IPv6 ist/wird die IP-Adresse noch länger.

4. Warum überhaupt DNS

Bevor es DNS gab, wurden die Namen zu den jeweiligen IP-Adressen in einer **zentralen** Textdatei (**hosts**-Datei) **statisch** geschrieben und dort gepflegt. Die Pflege dieser Datei wurde jedoch mit der immer größer werdender Rechner- bzw. Serveranzahl im Internet schwieriger und vor allem fehleranfälliger.

Auf Grund dieses Sachverhalts wurde DNS entwickelt. Hier wird die Verwaltung **dezentral**, **dynamisch** und **hierarchisch** organisiert.

Die hosts-Datei wird jedoch in kleinen Netzwerken immer noch verwendet um die lokale Namensauflösung ohne DNS zu realisieren.

Nähere Infos zur Hosts-Datei finden Sie hier: [Hosts-Datei](#).

5. Aufbau von DNS

Das DNS-System besteht aus einem ganzen Netzwerk von tausend einzelnen DNS-Servern, welche in einer hierarchischen Struktur organisiert sind. Durch die hierarchische Auslegung des Systems kommt es zu folgendem Domain-Namen-Aufbau:

www.Sub-Domain.Second-Level-Domain.Top-Level-Domain.Root-Server

5.1 Schriftliche Darstellung

Die Domain-Namen werden immer von **rechts nach links** gelesen bzw. aufgelöst., d.h.

1. Der Punkt am Ende stellt den/die **Root-Server** dar und ist somit der **wichtigste** Teil, da er die Wurzel für den ganzen Domain-Namen ist. Dieser Punkt muss jedoch bei den meisten Anwendung (z.B Browser) nicht mit eingegeben werden, das macht die Anwendung automatisch. Es gibt weltweit 13 Root-Server. Ursprünglich waren es einmal 10, welche alle in der USA standen. Das ist aber gegen den Gedanken der Dezentralisierung und somit wurden noch drei weitere Server außerhalb den Staaten installiert. Die Pflege dieser Server wird von der **ICANN** (Internet Corporation for Assigned Names and Numbers) durchgeführt.

Nähere Infos zur ICANN finden Sie hier: [ICANN](#)

2. Als nächster Teil kommt die **Top-Level-Domain**, bei der die Unterscheidung in **ccTLD's** und **gTLD's** vorgenommen wird.
 - ccTLD's (country-code-TLD) sind länderspezifische/geografische TLD's (z.B. .de)
 - gTLD's (generic-TLD) sind generische oder organisatorische TLD's (z.B. .com)

Eine Liste aller TLD's findet man unter folgendem Link: [TLD's Datenbank](#)

3. Die **Second-Level-Domain** folgt auf TLD und ist ein eindeutiger String. Die Verwaltung der SLD's für die ccTLD .de wird von der DENIC vorgenommen.

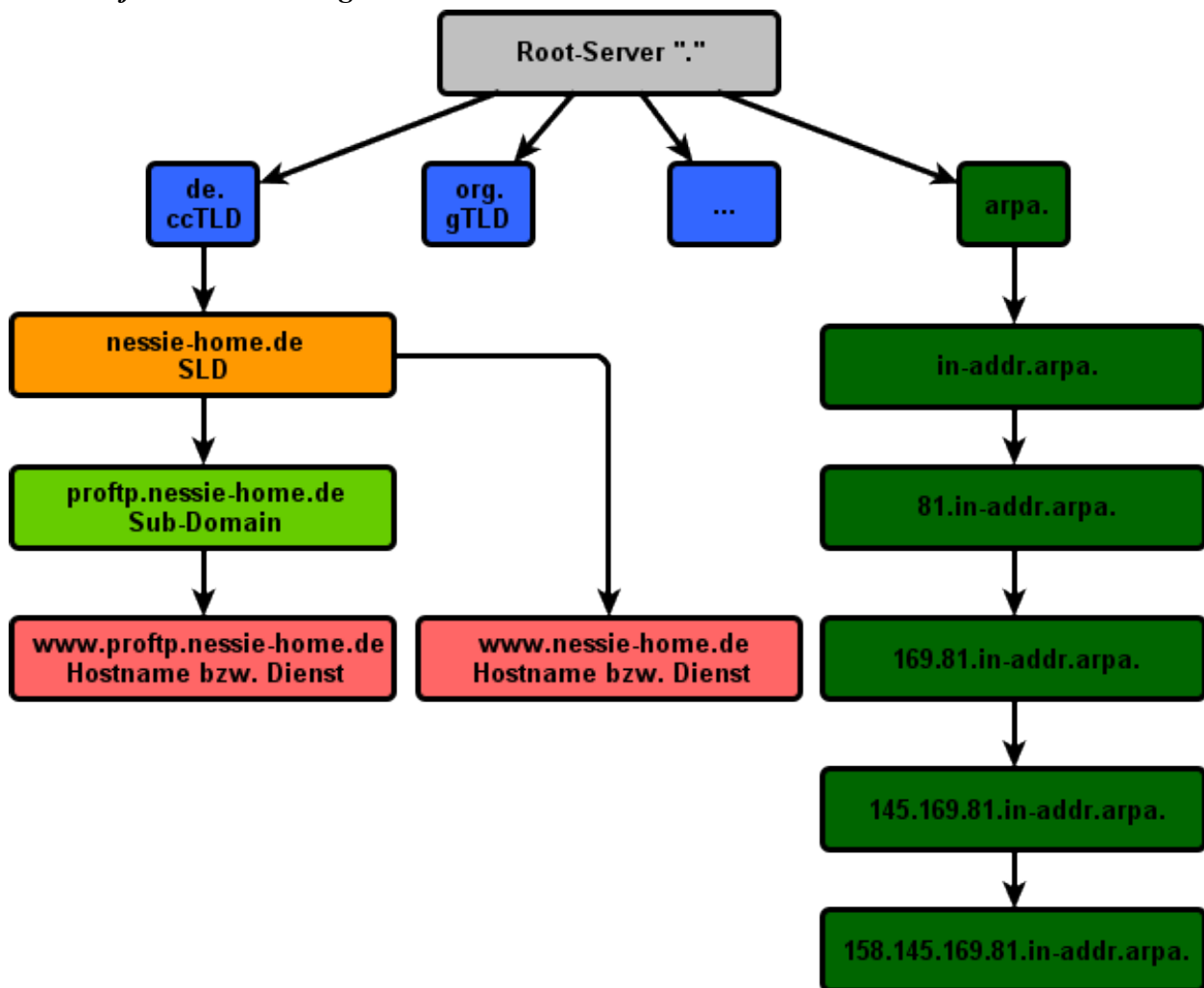
Nähere Infos zur DENIC finden Sie hier: [DENIC](#)

4. Eine **Sub-Domain** kann unterhalb einer SLD eingerichtet werden und muss wieder eindeutig sein. Die Verwaltung dieser Sub-Domain's muss der Besitzer der SLD durchführen.
5. Am Schluss kommt der Hostname (z.B. www, ftp), welcher nicht zwingend erforderlich ist.

Wenn alle Spezifikationen eingehalten werden spricht man von einem „**Fully-Qualified-Domain-Name**“ (FQDN) [www.nessie-home.de](#).

Neben den oben genannten TLD's gibt es noch die **apra**, welche für das Auflösen von Hostnamen zu IP-Adressen (Reverse-Lookup) benötigt wird. Dort werden die Adressen entsprechend umgekehrt angefordert, d.h. die IP 127.0.0.1 wird als 1.0.0.127.in-addr.arpa. aufgelöst.

5.2 Grafische Darstellung



6. Server- / Client-Bestandteile

6.1 Server: Nameserver

Der Nameserver ist der Teil des DNS-System, der die oben genannte Auflösung vornimmt. Es handelt sich hierbei um eine Software, obwohl oftmals auch Hardware, die Namensauflösungen vornimmt, fälschlicherweise als Nameserver bezeichnet werden. Es gibt grundsätzlich drei Arten von Nameservern, den **autoritativen**, **nichtautoritativen** und **caching-only** Server.

- *autoritativer Nameserver*

Bei einem autoritativen Nameserver handelt es sich um einen Nameserver (Primary-Nameserver) der eine bestimmte Zone mit dessen Zonenfile verwaltet (z.B. de). Die Auskunft eines solchen Server wird als gesichert angesehen. Um eine Ausfallsicherheit und bessere Lastenverteilung einer Zone zu garantieren wird eine Zone auf mehrere Server (ServerCluster) verteilt. Es liegt jedoch auf jedem dieser Nameserver die identische Zonendatei. Die Verteilung (Zonentransfer) einer geänderten Zonendatei auf die einzelnen Secondary-Nameserver übernimmt der Primary-Nameserver.

- *nichtautoritativer Nameserver:*

Der nichtautoritative Nameserver bezieht seine Informationen von anderen Nameserver, womit seine Auskunft als nicht gesichert angesehen wird. Damit der Nameserver aber nicht

bei jeder einzelnen Anfrage wieder erneut beim autoritativen Server anfragen muss beherrschen nichtautoritative Nameserver das sogenannte „Caching“. Das bedeutet eine Auflösung wird für eine bestimmte Zeit (wird vom autoritativen Server geliefert) im RAM vorgehalten und dann wieder gelöscht. D.h im Klartext das es bei Einträgen die sich häufig ändern evtl. zeitweise zu einer falschen Auskunft kommt, weil noch der alte Eintrag im Cache steht.

- *caching-only Nameserver:*

Dieser Server verwaltet keine eigene Zone, somit leitet er Anfragen nur an andere Nameserver weiter (**Forwarder**).

6.2 Client: Resolver

Ein Resolver ist eine Software, die heutzutage in jedem Betriebssystem enthalten ist. Anfragen an Nameserver werden von ihm gesendet und die Antworten vom Nameserver von ihm wiederum entgegen genommen. Der Resolver stellt also das Bindeglied zwischen einer Anwendung (z.B. Browser) und dem Nameserver dar. Resolver arbeitet in zwei Modi **rekursiv** oder **iterativ**.

- *rekursiver Resolver*

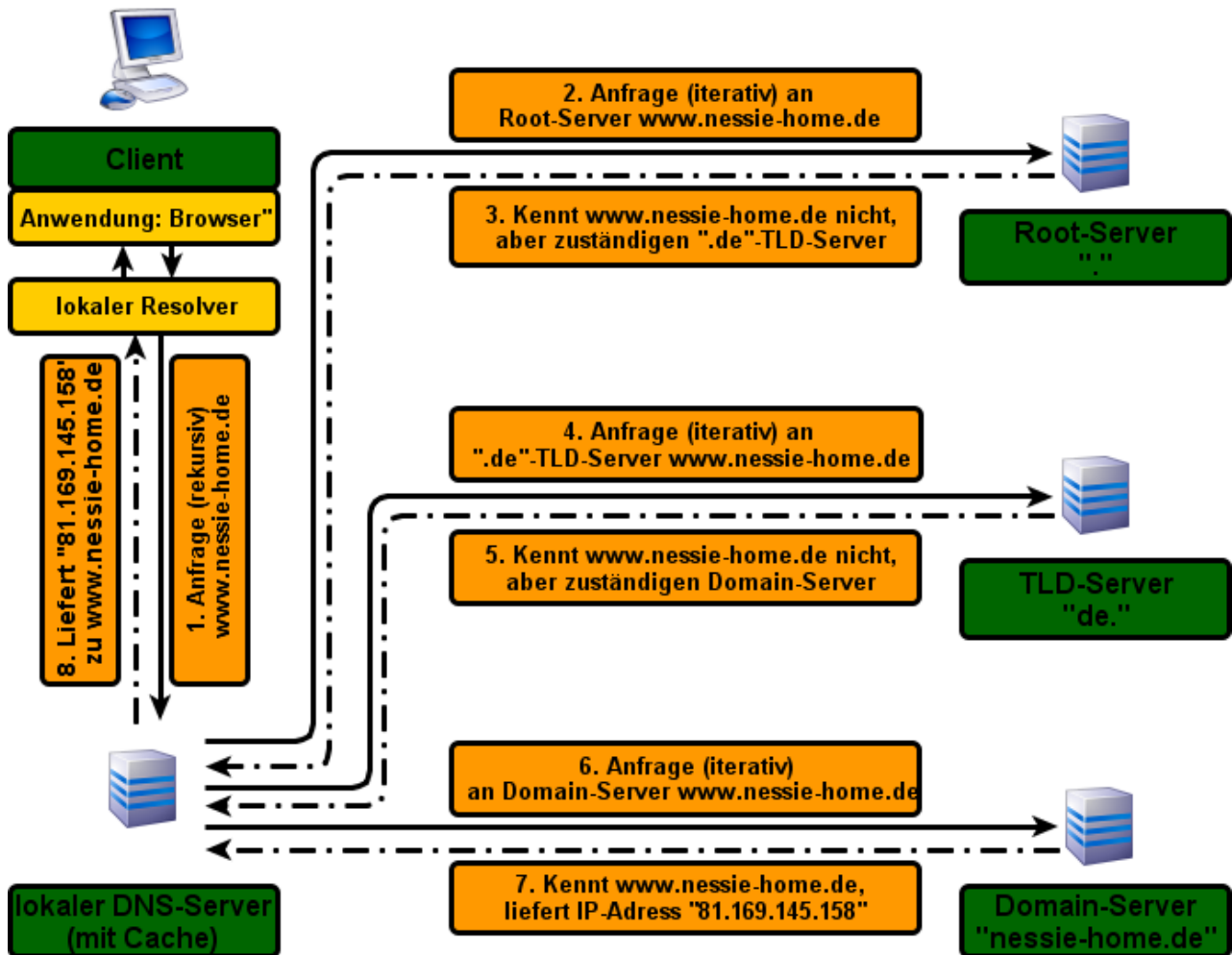
Hier stellt der Resolver eine Anfrage an den zugeordneten Nameserver, dieser Nameserver sieht in seinem Datenbestand nach dem angeforderten DNS-Eintrag. Findet er nicht den passenden Eintrag so kontaktiert er weitere Nameserver und zwar solange bis er von einem autoritativen Server eine positive oder negative Antwort bekommen. Somit erkennt man schnell das hier der Resolver die komplette Arbeit an den Nameserver übergibt.

- *iterativer Resolver*

Hier stellt der Resolver ebenfalls wieder die Anfrage an dem ihm zugeordneten Nameserver und wartet auf seine Antwort und bekommt von diesem einen passenden Eintrag (Resource-Record) oder einen Verweis (Delegation) zum nächsten zu befragenden Nameserver. Somit übernimmt hier der Resolver die Arbeit selbst.

Normalerweise arbeiten Client-Resolver **ausschließlich** rekursiv und werden als **Stub-Resolver** bezeichnet. Jeder Nameserver beinhaltet ebenfalls einen Resolver, welche iterativ arbeiten.

6.3 Grafisches Beispiel



7. Die Zonendatei

7.1 Die Zonendatei

Die Zonendateien beinhalten Einträge für die ordnungsgemäße Namensauflösung einer einzelnen Domain (Zone). Auf einem Nameserver liegt jeweils eine oder mehrere Zonendateien. Der Inhalt der Dateien wird im ASCII-Format verfasst und ist somit mit sämtlichen Texteditoren einsehbar und editierbar.

- Forward-Zone: dient zur Auflösung von DNS-Namen zu IP-Adressen
- Reverse-Zone: dient zur Auflösung von IP-Adressen zu DNS-Namen

Die Einträge einer Zonendatei nennt man **Resource Record's**, während die Menge aller Einträge als **Record Set** (RRset) bezeichnet wird.

Jeder RR hat folgendes Format:

<Name> <TTL> <Klasse> <Typ> <Daten>

Ich werde hier nicht auf alle Klassen, Typen oder Daten eingehen, jedoch die wichtigsten RR's hier auflisten und kurz erklären.

Nähere Infos zur Zonendatei finden Sie hier: [Resource Records](#)

7.2 Wichtige Resource Records

- *SOA-RR (Start of Authority)*

Der wohl wichtigste Eintrag in einer Zonendatei, den er legt fest, dass diese Zonendatei und somit der Nameserver der sie verwaltet der autoritativ Nameserver der Zone ist!

nessie.lan.	IN	SOA	master.nessie.lan.	root.nessie.lan.	2010010401	3600	1800	604800	1800
Name	Zonen-Name (nessie.lan.)								
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet								
SOA	Kürzel für SOA-Eintrag (Start Of Authority)								
Primary-Master	Hat nur noch wenig Bedeutung für die Praxis (master.nessie.lan.)								
Mail-Adresse	Mail-Adresse des Zonenverantwortlichen, das "@" wird als "." dargestellt. Punkte müssen mit einem "\" masikert werden. (webmaster.nessie.lan.)								
Serien-Nummer	Dient der Versionierung der Zonendatei (2010010401 = Version 01 vom 04.01.2010)								
Refresh	Wann ein Slave-Namensserver die Zonendatei aktualisieren soll (Default = 24 Std., hier 1 Std.)								
Retry	Wann ein Fehlschlag des Refreshs wiederholt werden soll (Default = 2 Std., hier 30 Min.)								
Expire	Wann der Slave die Zone nach nicht Beantwortung eines Zonefile-Request deaktiviert (Default = 1000 Std., hier 168 Std.)								
TTL	Time To Live: Wie lange eine Anfrage gecacht wird (Default = 2 Tage, hier 30 Min.)								

- **NS-RR (NameServer-Resource Record):**

Dient zur Auflistung aller autoritativen Nameserver einer Zone und ist mindestens einmal in jeder Zonendatei vorhanden. Der Eintrag befindet sich immer direkt hinter dem SOA-RR. Des weiteren wird dieser NS-RR für die Delegation benötigt, d.h. als Zeiger auf einen

anderen Nameserver.

nessie.lan.	1800	IN	NS	names1.nessie.lan.
Domain	Wert für welche Domain der Eintrag ist (nessie.lan.)			
TTL	Time To Live: Wie lange eine Anfrage gecacht wird (Default = 2 Tage, hier 30 Min.)			
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet			
Dienst	Name Service			
Server	Name des autoritativen Servers (names1.nessie.lan.)			

- *A-RR (A-Resource Record)*

Mithilfe des A-RR wird ein DNS-Name einer IPv4-Adresse zugeordnet. Wenn ein Host mehrere Netzwerkschnittstellen besitzt oder ein Dienst auf mehrere Server verteilt ist, so können mehrere A-RR's für die Netzlastverteilung verwendet werden oder die Ausfallsicherheit erhöhen.

nessie.lan.	3600	IN	NS	192.168.1.200
Hostname	Wert für den öffentlichen Hostnamen (nessie.lan.)			
TTL	Time To Live: Wie lange eine Anfrage gecacht wird (hier 1 Std.)			
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet			
Typ	A-Record, welche eine IPv4-Adresse angibt			
Adresse	IP-Adresse des Hosts (192.168.1.200)			

- **AAAA-RR (AAAA-Resource Record, gesprochen „quad-A-RR“)**

Dieser RR entspricht dem A-RR, nur das die IPv4-Adresse durch die IPv6-Adresse ersetzt.

- **CNAME-RR (CNAME-Resource Record)**

Ein CNAME-RR wird verwendet um für einen A- oder AAAA-RR-Eintrag ein bzw. mehrere Alias(s) zu erstellen (kanonischer Name)

www.nessie.lan.	3600	IN	CNAME	nessie.lan.
Alias	Wert des Alias (www.nessie.lan.)			
TTL	Time To Live: Wie lange eine Anfrage gecacht wird (hier 1 Std.)			
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet			
Typ	CNAME = kanonischer Eintrag			
CNAME	Kanonischer Name des Originaleintrags (nessie.lan.)			

- **MX-RR (Mail Exchange Resource Record)**

Der MX-RR gibt an unter welchem Namen der Mail-Server für die Domain erreichbar ist. Üblicherweise werden hier wiederum mehrere Server angelegt um die Ausfallsicherheit zu erhöhen.

nessie.lan.	IN	MX	10	mail.nessie.lan.
Domain	Wert der Domain für die der Mail-Server gültig ist (nessie.lan.)			
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet			
Typ	MX = Mail Exchange, sprich es handelt sich um einen Domain-Mail-Server-Eintrag			
Priorität	Priorität mit der der Eintrag behandelt werden soll. Je niedriger der Wert um so größer ist die Priorität des Servers (10)			
Name	Hostname über den der Domain-Mail-Server angesprochen werden kann. (mail.nessie.lan.)			

- **PTR-RR (Pointer Resource Record)**

Ein PTR-RR ermöglicht das Auflösen von IP-Adressen zu einem bestimmten Hostnamen, sprich es ist das Gegenstück zum A- bzw. AAAA-RR. Damit ist dieser Eintrag der Hauptbestandteil für einen Reverse-Lookup.

Neben der **in-addr.arpa.-Domain** für IPv4-Adressen wurde die **ip6.arpa.-Domain** für IPv6-Adressen eingerichtet. Durch diese Einrichtung ist ein schnellerer Reverse-Lookup möglich, denn es müssen nicht so viele „Äste“ durchsucht werden wie unter den normalen Domain's.

201.1.168.192.in-addr.arpa.	3600	IN	PTR	test.nessie.lan.
Adresse + in-addr.arpa.	Die IP-Adresse wird in umgekehrter Reihenfolge (Reverse-Lookup) angegeben (201.1.168.192.in-addr.arpa.)			
TTL	Time To Live: Wie lange eine Anfrage gecacht wird (1 Std.)			
Zonenklasse	„IN“ (Internet), es gibt noch weitere Zonenklasse werden aber nur selten verwendet			
Typ	PTR = Zeiger			
Name	Hostname der der IP-Adresse zugeordnet werden soll. (test.nessie.lan.)			

8. Diagnosetools

8.1 nslookup

nslookup (name server lookup) ist eines des ältesten Tools und liegt eigentlich sämtlichen Betriebssystemen bei. Mit dem Tool können sowohl DNS-Namen zu IP-Adressen gesucht werden sowie IP-Adressen zu DNS-Namen. Es kann in zwei Betriebsmodi bedient werden. Man kann *interaktiv* damit arbeiten, sprich nach dem eingeben von nslookup erhält man einen Eingabeprompt in den man seine Befehle eingeben kann. Durch *exit* kann man den Befehlsprompt wieder verlassen. Aber auch das Arbeiten mit der ganz normalen Befehlszeilenmodus ist möglich. Die Optionen werden einfach an den Befehl nslookup angehängt.

Nähere Infos zu nslookup finden Sie hier: [nslookup](#)

8.2 DIG

DIG (Domain Inter Groper) liegt dem BIND-Paket (Linux-DNS-Server) bei oder kann als Programm für Windows heruntergeladen ([DIG for Windows](#)) werden. Dieses Programm ist das perfekte Tool um eine Fehleranalyse durchzuführen, den es gibt seine Ergebnisse im Zonenfile-Format aus.

Nähere Infos zu dig finden Sie hier: [DIG](#)

Beispiele

- dig bzw. dig . NS gibt alle Root-Server aus
- dig nessie-home.de

```
;; <<>> DiG 9.3.2 <<>> nessie-home.de
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 464
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;nessie-home.de.                IN      A

;; ANSWER SECTION:
nessie-home.de.                7200    IN      A      81.169.145.158

;; Query time: 7 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Tue Jan 04 15:39:43 2011
;; MSG SIZE rcvd: 48
```

In der Ausgabe sieht man das in der **ANSWER SECTION** für nessie-home.de die IP-Adresse 81.169.145.158 ermittelt wurde.

Die drittletzte Zeile gibt an von welchem Nameserver die Antwort auf die Anfrage stammt, hier ist es mein Internet-Gateway. Da dieser Server kein autoritativer DNS-Server ist, steht in der sechsten Zeile von oben unter **AUTHORITY** eine 0.

- dig nessie-home.de +trace

```

; <<>> DiG 9.3.2 <<>> nessie-home.de +trace
;; global options: printcmd
.      82670      IN      NS      d.root-servers.net.
.      82670      IN      NS      k.root-servers.net.
.      82670      IN      NS      h.root-servers.net.
.      82670      IN      NS      b.root-servers.net.
.      82670      IN      NS      g.root-servers.net.
.      82670      IN      NS      e.root-servers.net.
.      82670      IN      NS      f.root-servers.net.
.      82670      IN      NS      m.root-servers.net.
.      82670      IN      NS      l.root-servers.net.
.      82670      IN      NS      i.root-servers.net.
.      82670      IN      NS      c.root-servers.net.
.      82670      IN      NS      a.root-servers.net.
.      82670      IN      NS      j.root-servers.net.
;; Received 449 bytes from 192.168.1.254#53<192.168.1.254> in 3 ms

de.    172800     IN      NS      z.nic.de.
de.    172800     IN      NS      s.de.net.
de.    172800     IN      NS      a.nic.de.
de.    172800     IN      NS      l.de.net.
de.    172800     IN      NS      f.nic.de.
;; Received 286 bytes from 128.8.10.90#53<d.root-servers.net> in 109 ms

nessie-home.de. 86400     IN      NS      shades16.rzone.de.
nessie-home.de. 86400     IN      NS      docks04.rzone.de.
;; Received 83 bytes from 194.246.96.1#53<z.nic.de> in 134 ms

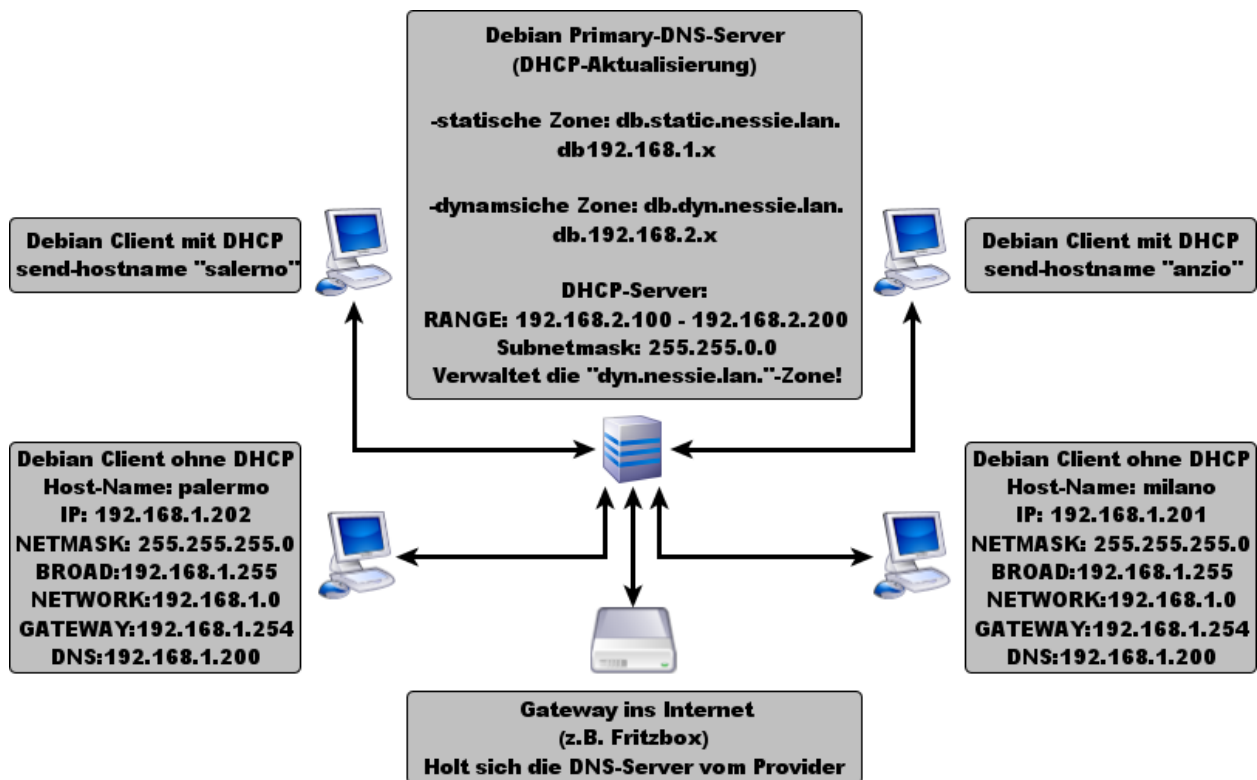
nessie-home.de. 7200      IN      A       81.169.145.158
nessie-home.de. 7200      IN      NS      docks04.rzone.de.
nessie-home.de. 7200      IN      NS      shades16.rzone.de.
;; Received 99 bytes from 85.214.0.246#53<shades16.rzone.de> in 24 ms

```

Durch den Zusatz *+trace* wird genau aufgelöst bzw. angezeigt wie sich der lokale DNS-Server durch die DNS-Hierarchie „hangelt“ um eine Auskunft zu seiner Anfrage zu bekommen.

9. Praxis-Übung

9.1 Grafik zur Praxis-Übung



9.2 Allgemeine Beschreibung

In dieser Praxis-Übung wird erläutert wie Sie für Ihr lokales Netzwerk einen lokalen DNS-Server einrichten. Dieser Server wird sowohl statische IP-Adressen wie dynamischen IP-Adressen vom lokalen DHCP-Server verwalten.

Die Rechner mit festen IP-Adressen werden in der Zone „static.nessie.lan.“ verwaltet, während die DHCP-Adressen in der Zone „dyn.nessie.lan.“ organisiert werden. Diese Trennung wird gemacht, da somit die Verwaltung der einzelnen IP-Adressbereiche einfacher zu handhaben ist. Für die Auflösung von Internet-Adressen wird ein Forward zu einem dafür geeigneten DNS-Server erstellt.

9.2.1 Installation eines Debian Squeeze 6.0 (NetInstallation)

```
Welche Software soll installiert werden?
```

<input type="checkbox"/>	Grafische Desktop-Umgebung
<input type="checkbox"/>	Web-Server
<input type="checkbox"/>	Druck-Server
<input checked="" type="checkbox"/>	DNS-Server
<input type="checkbox"/>	Datei-Server
<input type="checkbox"/>	Mail-Server
<input type="checkbox"/>	SQL-Datenbank
<input checked="" type="checkbox"/>	SSH-Server
<input type="checkbox"/>	Laptop
<input checked="" type="checkbox"/>	Standard-Systemwerkzeuge

<Weiter>

9.2.2 DHCP-Server

Der DHCP-Server muss nachträglich von Ihnen installiert werden.

```
apt-get install dhcp3-server
```

9.2.3 *Dienste stoppen*

Vor der eigentlichen Konfiguration sollten Sie erst einmal die entsprechenden Dienste stoppen

```
/etc/init.d/bind9 stop
```

```
dhcpcd stop
```

9.2.4 *Einstellungen in der named.conf.options*

Sie müssen nun in der Datei „/etc/bind/named.conf.options“ folgendes ergänzen:

```
### Schnittstellen an denen der Server nach Anfragen lauscht ###
listen-on {127.0.0.1; 192.168.1.200; };

### Forward-Server fuer Anfragen die der eigene DNS-Server selber nicht
auflösen kann. ###
### Sprich Anfragen von Adressen im Internet (DSL-Router) ###
    forwarders {
        192.168.1.254;
    };

### Welche Rechner an den DNS-Server Anfragen stellen dürfen ###
allow-query{
    127.0.0.1; 192.168.0.0/16;
};
```

9.2.5 *Einstellungen in der named.conf.local*

Die Datei „/etc/bind/named.conf.local“ muss wie folgt von Ihnen bearbeitet:

```
include "/etc/bind/rndc.key";
### Forward-Zone fuer die Rechner mit statischen IP-Adressen ###
zone "static.nessie.lan." {
    type master;
    file "/var/cache/bind/db.static.nessie.lan";
    allow-update { key "rndc-key"; };
};

### Forward-Zone fuer die Rechner mit dynamischen IP-Adressen von DHCP-Server
###
zone "dyn.nessie.lan." {
    type master;
    file "/var/cache/bind/db.dyn.nessie.lan";
    allow-update { key "rndc-key"; };
};

### Reverse-Zone fuer die Rechner mit statischen IP-Adressen ###
zone 1.168.192.in-addr.arpa. {
    type master;
    file "/var/cache/bind/db.192.168.1";
```

```

        allow-update { key "rndc-key"; };
};

### Reverse-Zone fuer die Rechner mit dynamischen IP-Adressen von DHCP-Server
###

zone 2.168.192.in-addr.arpa. {
    type master;
    file "/var/cache/bind/db.192.168.2";
    allow-update { key "rndc-key"; };
};

### Den Zugriff auf den DNS-Server nur vom lokalen Rechner erlauben ###
controls {
    inet 127.0.0.1 allow { localhost; } keys { "rndc-key"; };
};

```

9.2.6 Forward-Zonen-Dateien

Als erstes legen Sie die Forward-Zonen-Dateien für die statischen Adressen an:

```

; Forward-Zone fuer die Domain "static.nessie.lan.", also die statischen IP-
Adressen
;
$ORIGIN static.nessie.lan.
$TTL 1D      ; 1 Tag
@      IN    SOA    avalanche.static.nessie.lan. root.nessie.lan. (
                2011012801 ; serial
                8H        ; refresh (8 Stunden)
                2H        ; retry (2 Stunden)
                4W        ; expire (4 Wochen)
                2D        ; minimum (2 Tage)
        )
;
        IN    NS      avalanche.static.nessie.lan.
        IN    MX      10 mail.static.nessie.lan.
;
localhost  IN    A      127.0.0.1
;
avalanche  IN    A      192.168.1.200
mail       IN    A      192.168.1.200
;
milano     IN    A      192.168.1.201
palermo    IN    A      192.168.1.202
;
fritzbox   IN    A      192.168.1.254

```

Als zweites legen Sie die Forward-Zonen-Dateien für die dynamischen Adressen an:

```

; Forward-Zone fuer die Domain "dyn.nessie.lan.", also die DHCP-IP-Adressen
;
$ORIGIN dyn.nessie.lan.
$TTL 1D      ; 1 Tag
@      IN    SOA    avalanche.dyn.nessie.lan. root.nessie.lan. (
                2011012801 ; serial
                8H        ; refresh (8 Stunden)
                2H        ; retry (2 Stunden)
                4W        ; expire (4 Wochen)
                2D        ; minimum (2 Tage)
                )
;
                IN      NS      avalanche.static.nessie.lan.
;
avalanche  IN      A      192.168.1.200

```

9.2.7 *Reverse-Zonen-Dateien*

Als nächstes müssen Sie die Reverse-Zonen-Datei für die statischen Adressen erstellen:

```

; Reverse-Zone fuer die Domain "static.nessie.lan.", also die statischen IP-
Adressen
;
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 1D      ; 1 Tag
@      IN    SOA    avalanche.static.nessie.lan. root.nessie.lan. (
                2011012801 ; serial
                8H        ; refresh (8 Stunden)
                2H        ; retry (2 Stunden)
                4W        ; expire (4 Wochen)
                2D        ; minimum (2 Tage)
                )
;
                IN      NS      avalanche.static.nessie.lan.
200    IN      PTR      avalanche.static.nessie.lan.
;
200    IN      PTR      mail.static.nessie.lan.
;
201    IN      PTR      milano.static.nessie.lan.
202    IN      PTR      palermo.static.nessie.lan.
;
254    IN      PTR      fritzbox.static.nessie.lan.

```

Für die dynamischen Adressen müssen Sie ebenfalls die Reverse-Zonen-Datei anfertigen:

```

; Reverse-Zone fuer die Domain "dyn.nessie.lan.", also die DHCP-IP-Adressen
;
$ORIGIN 2.168.192.in-addr.arpa.
$TTL 1D      ; 1 Tag
@      IN      SOA  avalanche.static.nessie.lan. root.nessie.lan. (
                2011012801 ; serial
                8H        ; refresh (8 Stunden)
                2H        ; retry (2 Stunden)
                4W        ; expire (4 Wochen)
                2D        ; minimum (2 Tage)
                )
;
                IN      NS      avalanche.dyn.nessie.lan.
200     IN      PTR      avalanche.dyn.nessie.lan.

```

9.2.8 Zonen-Dateien in /var/cache/bind/ verlinken

Die Zonen-Dateien sollten Sie aber nicht im Verzeichnis „/etc/bind/“ verwalten, sondern im Verzeichnis „/var/cache/bind“. Dazu müssen Sie mit den Befehlen symbolische Links anlegen:

```

ln -sf /etc/bind/db.static.nessie.lan /var/cache/bind
ln -sf /etc/bind/db.dyn.nessie.lan /var/cache/bind
ln -sf /etc/bind/db.192.168.1 /var/cache/bind
ln -sf /etc/bind/db.192.168.2 /var/cache/bind

```

Die symbolische Links müssen Sie nun noch mit den entsprechenden Zugriffsrechten versehen, so das BIND auch Zugriff hat:

```
chmod -h root:bind /var/cache/bind/*
```

9.2.9 DHCP-Konfiguration

Nun kommen wir zur Konfiguration des DHCP-Servers. Hier müssen Sie die Konfigurations-Datei „/etc/dhcpd.conf“ wie folgt editiert werden:

```

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)

```

```

ddns-update-style    interim;
ddns-updates         on;
ddns-domainname      "dyn.nessie.lan.";
ddns-rev-domainname  "in-addr.arpa.";

```

```
update-static-leases    off;
```

```
include "/etc/bind/rndc.key";
```

```
ignore client-updates;
```

```

# option definitions common to all supported networks...
default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.0.0 netmask 255.255.0.0 {
    range 192.168.2.100 192.168.2.200;
    option domain-name "static.nessie.lan dyn.nessie.lan";
    option domain-name-servers 192.168.1.200;
    option routers 192.168.1.254;
    option subnet-mask 255.255.0.0;
    option broadcast-address 192.168.255.255;
    option netbios-name-servers 192.168.1.200;
}

#DNS Zonenupdate
zone 2.168.192.in-addr.arpa. {
    primary 192.168.1.200;
    key "rndc-key";
}

zone dyn.nessie.lan {
    primary 192.168.1.200;
    key "rndc-key";
}

```

9.2.10 *Resolv.conf anpassen*

Abschließend müssen Sie nur noch in der Datei „/etc/ressolv.conf“ den Resolver auf „127.0.0.1“ ändern. Denn erst durch diese Änderung wird der eigene DNS-Server verwendet.

9.2.11 *Dienste starten*

Da jetzt alle nötigen Dateien erstellt wurden, können Sie die Dienste starten:

```
/etc/init.d/bind9 start
```

```
dhcpcd start
```

Sollten sich die Dienste nicht starten lassen, können Sie unter „/var/log/syslog“ auf Fehlersuche gehen.

9.2.12 *Erweitertes Logging*

Sie können aber auch das Logging erweitern in dem Sie folgende Zeilen in die „/etc/bind/named.conf.local“ ergänzen:

```
logging {
    channel queries_log {
        file "/var/log/bind/bind-queries.log";
        severity debug 9;
        print-category yes;
        print-severity yes;
        print-time yes;
    };
    channel update_log {
        file "/var/log/bind/bind-update.log";
        severity debug 9;
        print-category yes;
        print-severity yes;
        print-time yes;
    };
    channel security_log {
        file "/var/log/bind/bind-security.log";
        severity info;
        print-category yes;
        print-severity yes;
        print-time yes;
    };
    category default { default_syslog; };
    category queries { queries_log; };
    category update { update_log; };
    category security { security_log; };
};
```

Die Dateien des Parameter „file“ müssen Sie mit touch im entsprechendem Verzeichnis erstellen und mit passenden Dateizugriffsrechten versehen werden:

```
touch /var/log/bind/bind_queries.log
```

```
chmod 664 /var/log/bind/bind_queries.log
```

```
chown root:bind /var/log/bind/bind_queries.log
```

9.2.13 *FAQ's*

Ab Debian Squeeze ist die Root-Zonen-Datei „db.root“ signiert und die DNSSEC-Überprüfung ist bereits standardmässig aktiviert. Allerdings ist der öffentliche Schlüssel nicht in der Datei „named.conf“ nicht inkludiert, dies muss noch nachträglich gemacht werden.

```
include „/etc/bind/bind.keys“;
```

Des weiteren kommt es evtl. auch zu der Fehlermeldung

```
Mar  8 17:10:56 avalanche named[868]: /etc/bind/named.conf.local:9: open:  
/etc/bind/rndc.key: permission denied
```

```
Mar  8 17:10:56 avalanche named[868]: loading configuration: permission denied
```

Hier muss man mittels

```
chmod 644 /etc/bind/rndc.key
```

die Zugriffsrechte angepasst werden so das BIND ebenfalls lesend zugreifen kann.

Die Besitzer sollten wie folgt gesetzt sein:

```
-rw-r--r-- 1 bind bind  77  8. Mar 14:10 rndc.key
```

Jedoch gibt es auch folgende Benutzereinstellungen

```
-rw-r--r-- 1 root bind  77  8. Mar 14:10 rndc.key
```

die zur Lösung des Problems im Internet genannt wird.

10. Quellen

- DNS & BIND GE-PACKT (ISBN 3-8266-1502-6)
- Wikipedia ([Domain Name System](#))
- Internet Assigned Numbers Authority ([IANA](#))
- Internet Corporation for Assigned Names and Numbers ([ICANN](#))
- DENIC ([DENIC](#))
- Root-Server Technical Operations Ass ([Root-Servers](#))
- Elektronik Kompendium ([DNS - Domain Name System](#))
- TWEAKPC ([Was ist DNS und wie nutze ich das?](#))
- NetPlanet.org ([Domain Name System – DNS](#))
- Unterwegs im Net ([Was man über DNS wissen sollte...](#))
- Bind9 HowTo ([HowTo](#))
- Bind9 & DHCP HowTo ([HowTo für DDNS](#))