DNS

Systemweite Konfigurationsdateien

- /**etc**/**network**/**interfaces**: Hier steht die IP-Adresse des Rechners, die Netzmaske und das verwendete Gateway und der DNS-Resolver.
- /**etc**/**hostname**: Diese Datei enthält den Rechnernamen.
- /**etc**/**hosts**: Hier stehen feste Zuordnungen von IP-Adressen und Hostnamen.
- /**etc**/**resolv.conf**: Platz für einen oder mehrere Name-Server sowie die lokale DNS—Domain.
- /**etc**/**nsswitch.conf**: Hier wird definiert, in welcher Reihenfolge die Dateien zur Auflösung von Namen und IP-Adressen abgearbeitet werden.

Name-Server
Komplizierte IP-Adressen, über welche die einzelnen Rechner im Internet erreichbar sind, muss sich dank DNS niemand merken. Das »Domain Name System« verwaltet wie in einem Telefonbuch die Namensraäume im Internet und ordnet Bezeichnern wie »www.linux.org« den passenden Nummern (»198.182.796.56«) zu und umgekehrt.

DNS - so funktioniert's
In dem gesamten System spielen viele einzelne »Domain Name Server« mit, welche hierarchisch angeordnet sind und zusammenarbeiten. Weiss ein Server nichts mit einem Namen oder einer IP-Adresse anzufangen, kontaktiert er den übergeordneten Server, der wiederum andere Kollegen um Rat fragen kann, wenn er die Antwort nicht weiss. Wenn Sie in einem Browser beispielsweise die Adresse www.google.ch eingeben, kontaktiert der Rechner den Name-Server Ihres ISPs (falls zu Hause kein eigener Name-Server läuft). Kennt dieser die passende IP-Adresse nicht, kann er bei weiteren Servern anfragen, bis die richtige Adresse gefunden ist. Praktischerweise verfügt jeder Name-Server über einen Cache, welcher die Auskünfte eine Zeit lang zwischenspeichert. so dass der Server erst dann wieder Informationen einholen muss, wenn der Cache diese nicht mehr hat.
Neben den Informationen zu den Domains selbst muss es Programme geben, welche diese abrufen und auswerten können (so genannte Resolver), und mehrere Name-Server, welche sich die Arbeit teilen.

Lokales Adressbuch - die Datei »`/etc/hosts`«
DNS ist der Nachfolger eines recht einfachen Systems: In einer »Adressbuchdatei« stehen feste Zuordnungen von IP-Adressen und Hostnamen. Um die Rechner miteinander bekannt zu machen, wird die Datei verteilt und auf allen Maschinen abgelegt. Viele kleinere Heimnetze nutzen diese »privaten Adressbücher« immer noch, da sie schnell eingerichtet sind.
Zuständig ist auf Linux-Systemen die Datei `/etc/hosts` - in dieser stehen die IP-Adressen und Rechnernamen der einzelnen Maschinen. Eine typische hosts-Datei eines übersichtlichen LANs könnte beispielsweise so aussehen:

```
127.0.0.1 localhost
192.168.2.15 asteroid.hane.org      asteroid
192.168.2.16 transpluto.hane.org    transpluto
192.168.2.18 tablett.hane.org       tablett
```

Domain - Namensraum
Der Name localhost sollte immer der IP-Adresse 127.0.0.1 (Loopback-Adresse) zugeordnet werden. die spätestens, wenn das zu administrierende Netzwerk eine bestimmte Anzahl von Rechnern übersteigt, ist diese Herangehensweise ziemlich unpraktisch, und für ein riesiges Netzwerk wie das Internet ist dieses System nicht mehr realisierbar. Ein weiterer Nachteil der Datei `/etc/hosts` ist die fehlende Eindeutigkeit.

Um diese Computer eindeutig über einen Namen ansprechen zu können, organisiert man die Rechner in so genannten Domains: Wie in einem Baum angeordnet steht ganz am Anfang die Wurzel (root). Welche durch einen Punkt (**.**) repräsentiert wird. Davon ausgehend finden Sie in der nächsten Ebene die so genannte **Top Level Domain (TLD)**, wie z. B. org (für Non-Profit-Organisationen), mil (militärische Einrichtungen), com (kommerzielle Unternehmen) usw.; neben diesen organisatorischen Top Level Domains gibt es geografische TLDs (Länder-Codes). wie beispielsweise de (Deutschland), at (Österreich), ch (Schweiz) usw.
Wiederum durch einen Punkt abgetrennt folgt die **Second Level Domain (SLD)**, optional eine **Third Level Domain** und weitere Subdomains (alle jeweils durch Punkte abgegrenzt.

Die beschriebene Aufteilung der Domains bringt einen weiteren Vorteil mit: Für die einzelnen Name-Server und die damit zusammenhängenden verteilten Datenbanken bedeutet diese klare Struktur eine klare

Aufgabenverteilung. Ganz am oberen Ende stehen die so genannten Root Root Server Server, welche Informationen zu den Top Level Domains speichern. Eine vollständige Liste finden Sie auf den Webseiten der **IANA** (»Intemet Assigned Numbers Authority); interessant für die IP-Netze in Europa und der angrenzenden Regionen, ist weiterhin die Organisation RIPE (»Reseaux IP Europäens«, http://www.ripe.net/).

Auf den Ebenen darunter tummeln sich weitere Name-Server, welche für Domains oder Subdomains zuständig sind, und natürlich gibt es auch Delegation Server, die sich nur mit den Rechnern im eigenen Heimnetz beschäftigen. Anfragen werden so weitergeleitet und erfolgreich delegiert. In der Praxis könnte das beispielsweise so aussehen:

- Sie sind von zu Hause aus über einen Internet Service Provider mit dem Internet verbunden; eine IP-Adresse wird Ihnen vom ISP per DHCP zugeteilt, und Sie nutzen ebenfalls einen oder mehrere vom Provider zugeteilte Name-Server.
- Sie geben in einen Browser eine Adresse wie z. B. www.huhnix.org ein und kontaktieren dazu den Name-Server des ISPs.
- Der Name-Server beim Provider hat von dieser Adresse noch nie etwas gehört und muss nicht lange überlegen, sondern kontaktiert die Root Server mit der Frage: »Wo kann ich mehr über .org-Namen erfahren, bitte?«
- Der antwortende Root Server delegiert die Anfrage und verweist auf einen Name-Server, der über Informationen zu .org verfügt.
- Der Name-Server des Providers spricht anschliessend mit diesem Server und fragt dort nach: »Wer weiss etwas über huhnix. org, bitte?« und erhält Auskunft, welcher autoritative Name-Server für diese Domain zuständig ist.
- Dieser schliesslich löst das Rätsel und verrät dem Server des Providers, dass zum Namen www.huhnix.org die IP 83.142.228.128 gehört. Der Name-Server des ISPs gibt die Info an Ihren Browser weiter und merkt sich darüber hinaus die Daten für eine Weile.

Die Interaktion der einzelnen Name-Server ist natürlich nur die eine Seite in diesem Szenario - der Web-Browser (oder jedes andere Programm, welches Hostnamen auflösen will) muss weiterhin die Fähigkeit haben, mit dem Name-Server zu »sprechen«. Dabei greift der Browser auf einen Resolver zurück.

Resolver
Als Resolver bezeichnet man Programme, welche die Informationen zu den Adressbucheinträgen einholen, sie erfüllen also die Funktion einer Vermittlungsstelle zwischen den jeweiligen Anwendungen und dem Name-Server. Der Resolver kann bei seinen Nachforschungen im Wesentlichen zwei Wege beschreiten:

Rekursive Anfrage: Der Resolver sendet seine Frage an einen ihm bekannten Name-Server und erwartet eine eindeutige Antwort von ihm. Dieser meldet sich entweder mit der gewünschten Information zurück oder meldet, dass es keinen passenden Eintrag gibt.

Iterative Anfrage: Der Resolver erhält als Antwort die gesuchte Information oder als Empfehlung die Adresse eines weiteren Name-Servers, den er als Nächstes fragt. Auf diese Weise hangelt sich der Resolver von Name-Server zu Name-Server, bis er die eindeutige Antwort erhält.

Das zweite = Szenario (iterativ) ist eher unüblich: Die Resolver der meisten Systeme können derartigen Empfehlungen nicht folgen. so dass es sich in der Praxis normalerweise rekursive Anfragen handelt.
In den meisten Fällen werden Sie zu einem Domain-Namen die passende IP-Adresse benötigen, dies nennt man »**forward lookup**«. Die Umwandlung funktioniert aber auch in die andere Richtung: Beim so genannten »**reverse lookup**« suchen Sie zu einer IP-Adresse den Domain-Namen. Da es unglaublich umständlich und zeitaufwändig wäre, den kompletten Domain-Namensraum von Level zu Level nach der gesuchten IP-Adresse zu durchforsten, gibt es eine spezielle Domain, die bei dieser umgekehrten Suche (»reverse mapping«) hilft: in-addr .arpa (Top Level Domain **arpa** und Second Level Domain **in-addr**). Unterhalb dieser Domain existieren weitere drei Level; diese Subdomains sind für die verschiedenen Bytes der IP-Adressen zuständig, so dass maximal drei

Schritte zur Auflösung einer IP-Adresse erforderlich sind. Von rechts nach links gelesen repräsentieren diese Subdomains, welche jeweils eine Zahl von 0 bis 255 als Namen tragen, die folgenden Bereiche:

- 1. Ebene: das erste Byte, z. B. 83.in-addr.arpa
- 2. Ebene: das zweite Byte, z. B. enthält 142 .83 .in-addr. arpa die IP-Adressen aus dem Bereich 83.142.0. 0/ 16 (also der Bereich von 83.142.0.0 bis 83.142.255 .255)
- 3. Ebene: das dritte Byte, so enthält 228.142.83.in-addr.arpa IP-Adressen aus dem Bereich 83.142.228.0/24 (83.142.228.0 bis 83 . 142. 228. 255)

Name-Server

Ein Name-Server übernimmt die Verantwortung für eine oder mehrere so genannte Zonen, d.h. seine Informationen können als verbindlich, also autoritativ, angesehen werden. Ganz am oberen Ende stehen 13 so genannte Root Server, welche die volle Autorität besitzen. Diese verteilen als Chef des ganzen Systems allerdings immer nur Informationen zu Anfragen der Art »wer ist für .ch autoritativ?«.

Zonen sind »Verwaltungseinheiten« einer Domain oder ihrer Unterebenen: Für den Fully Qualified Domain Name huhnix.org ., der sich aus dem Domain-Namen (huhnix) und der Top Level Domain .org zusammensetzt, kann die verantwortliche Zone beispielsweise huhnix.org Zone heissen. Handelt es sich um eine sehr grosse Zone, gibt es die Möglichkeit, Arbeit an andere Name-Server zu delegieren.

Neben den autoritativen Name-Servern gibt es auch noch solche, welche ihre Informationen zu den Zonen aus zweiter oder dritter Hand beziehen; diese Server nennt man entsprechend nicht-autoritativ. Damit ein Name-Server, der viele Anfragen zu beantworten hat, nicht die ganze Last alleine auf den Schultern tragen muss, gibt es so genannte Primary und Secondary Name-Server, die ihre Aufgaben im Teamwork erfüllen:

- Primary bzw. Master Server: Für die verschiedenen Zonen, die lokal konfiguriert werden, ist der Server autoritativ.

- Secondary bzw. Slave Server: Ein Secondary Server bezieht die Daten zu den Zonen von dem für ihn zuständigen Master, legt diese auf der eigenen Platte ab (so dass die Informationen auch nach einem Neustart noch verfügbar sind) und meldet sich bei Anfragen für diese Zonen ebenfalls als autoritativ. Zusammen mit den Zonendaten erhält der Secondary Server Angaben, wie lange die Informationen gültig sind: Kann der Secondary seinen Primary Server nicht erreichen, wenn die Zeit abgelaufen ist, beantwortet er keine Anfragen mehr zu der Zone.

Zonendefinitionen

Eine Zone enthält alle für eine Subdomain relevanten Informationen. Diese Informationen werden als sogenannte Resource Records (RR) hinterlegt. Es gibt über 30 verschiedene RRs, die im Laufe der Zeit entwickelt wurden. Wirklich gebräuchlich sind aber nur wenige. Bevor Sie den ersten RR definieren, müssen Sie mit $TTL die Standard-Time-To-Live für alle RRs angeben. Dieser Wert sagt aus, wie lange Einträge gecacht werden dürfen. Die Angabe erfolgt in Sekunden. Mit unterschiedlichen Suffixen können aber auch Wochen w, Tage d, Stunden h oder Minuten m angegeben werden.

Nach der TTL folgt der wichtigste Resource Record, der SOA RR. SOA steht für **Start Of Authority** und definiert die Eigenschaften der Zone:

```
$TTL 12h
SOA Resource Record:
@      IN     SOA    ns1.example.net.  admin.example.net. (
                     1      ; serial
                     12h    ; refresh
                     1h     ; retry
                     10d    ; expire
                     1h     ; minimum (negTTL)
                     )
```

Das @-Zeichen am Anfang der Zeile gibt an, dass sich alle Einträge auf die in der /etc/named.conf definierte Domain beziehen. Die Schreibweise hat den Vorteil, dass man von mehreren Zonendefinitionen auf die gleiche Zonendatei verweisen kann.

Mit **IN** wird angegeben, dass es sich bei dem folgenden RR um die Klasse Internet handelt. Nach der Angabe des

SOA-Records folgt der **FQDN des primären Nameservers**, in diesem Fall ist es ns1.example.net. (die Angabe muss mit einem Punkt abgeschlossen werden). Danach folgt die **E-Mail-Adresse** des zuständigen Administrators.
In der Klammer des SOA-Records werden folgende Angaben gemacht:

serial: Die serial gibt die Versionsnummer der Zone an. Bei jeder Anpassung der Zonendatei muss sie inkrementiert werden. Die Secondary Nameserver benutzen die Seriennummer, um zu sehen, ob sie die Zone neu herunterladen müssen. Der Wertebereich für serial beträgt 32 Bit. Im Prinzip können Sie bei 1 anfangen zu zählen, wenn Sie die Zone anlegen. Es hat sich jedoch eingebürgert, die serial in folgendem Format darzustellen:
YYYYMMDDXX. Dabei steht YYYY für die vierstellige Jahreszahl, MM für den aktuellen Monat und DD für den Tag des Monats. XX ist ein fortlaufender Zähler und gibt an, wie oft die Zone bereits an diesem Tag geändert wurde.

refresh : Wird angegeben, nach welcher Zeit die Secondary Nameserver beim Primary nachschauen sollen, um zu prüfen, ob sich die serial geändert hat.

retry: Legt fest, nach welcher Zeit ein Secondary Nameserver erneut nachfragen soll, wenn der erste Versuch fehlgeschlagen ist.

expire: Wird quasi das Haltbarkeitsdatum für die Zone der Secondary Nameserver angegeben. Sollte der Secondary Nameserver nach Ablauf dieser Zeit noch immer keinen Kontakt zum Primary herstellen können, so wird die Zone ungültig.

Secondary-Server
Um ein gewisses Mass an Verfügbarkeit zu garantieren, brauchen Sie mehrere DNS-Server. Die DENIC schreibt für ch-Domains zwei bis fünf Nameserver vor. Einer der Server ist der Primary DNS und damit der Zonenmaster. Nur auf ihm werden die Zonendateien gepflegt. Die Secondary Nameserver oder Slave-Server holen sich in regelmässigen Abständen (siehe den SOA-Record) die Zonendateien vom Master. Um einen Slave-Server einzurichten, müssen Sie lediglich einen weiteren BIND-Server aufsetzen und ihm in der named.conf mitteilen, für welche Zonen er den Slave spielen soll. Als Zonentyp wird slave gesetzt. Zusätzlich muss mit masters angegeben werden. von welchem Server sich der Slave die Zone holen soll. Ein Beispiel für die Einträge in der named.conf für einen Slave-Server:

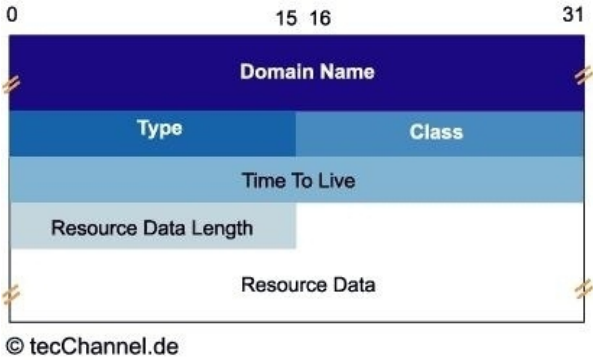
```
zone "example.net" {
type slave;
file “slave/exampie.net";
masters { 192.168.2.10; };
};
```

Damit der Zonentransfer funktioniert, muss die bei `file` angegebene Datei für den BIND-Server beschreibbar sein. Auf dem Master sind im Prinzip keine weiteren Einstellungen notwendig. Um aber zu verhindern, dass jeder Rechner im Netz einfach Ihre Zonendateien abfragen kann, sollten Sie Zonentransfers unbedingt auf die Slave-Server beschränken. Um das zu erreichen, müssen Sie im `options`-Block auf dem Master mit der Direktive `allow-transfer` eine Liste von IP-Adressen setzen. Ein Beispiel für die Einträge:

```
options {
directory "/var/cache/bind";
listen-on { any; };
iisten-on-v6 { none; };
allow-transfer { 127 0.0.1; 192.168.100.10 ; };
};
```

Protokoll

Die **Resource Records** werden bei einer Anfrage an den Client übermittelt. Dazu werden diese an den DNS-Header angehängt.



Resource Record: Für jeden DNS-Eintrag existiert ein Resource Record, der bei einer Anfrage an einen Nameserver als Antwort übermittelt wird.

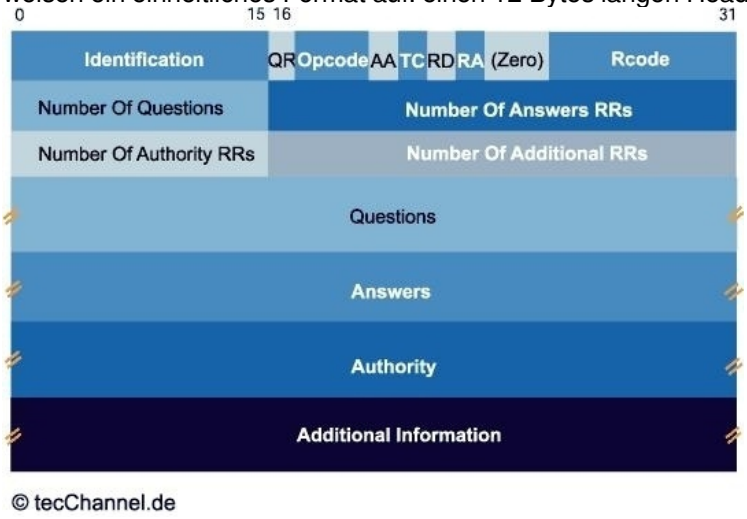
Der RR einer DNS-Antwort enthält folgende sechs Felder. Resource Records im Detail:

Typ	Beschreibung
Domain Name	Enthält den Domainnamen, der aufgelöst werden soll
Type	Spezifiziert den RR-Typ
Class	In diesem Feld steht in der Regel 1 für Internet-Daten
Time To Live	Enthält die Anzahl in Sekunden, wie lange ein anderer Nameserver das Ergebnis zwischenspeichert; meistens sind dies zwei Tage
Resource Data Length	Gibt die Länge des Feldes "Resource Data" an
Resource Data	Enthält die IP-Adresse

Zusätzlich gibt es so genannte Mail Exchange Records (MX Records), die für das Versenden von E-Mails von Bedeutung sind.

DNS-Nachrichten

Über DNS-Nachrichten tauschen Resolver und Server Abfragen und Antworten aus. DNS-Nachrichten weisen ein einheitliches Format auf: einen 12 Bytes langen Header und die Resource Records.



Bit für Bit: DNS-Nachricht im Detail.

Feld	Grösse (in Bit)	Beschreibung
Identification	16	Ein vom Client erzeugter Wert zur Identifikation des Resolvers
QR	1	Ist die Nachricht eine Anfrage, steht im Feld 0; bei einer Antwort 1
Opcode	4	Der Standardwert ist 0 (rekursive Anfrage); weitere mögliche Werte sind 1 für eine iterative Anfrage und 2 für eine Anfrage des Server-Status
AA	1	Authoritive Answer: Der Nameserver ist autoritativ für die aufzulösende Domain
TC	1	Truncated: Bei UDP bedeutet dies, dass die Antwort grösser als 512 Bytes ist und auch nur diese übermittelt werden
RD	1	Recursion Desired: Wenn der Client das Bit auf 1 setzt, ist dies eine rekursive Anfrage

RA	1	Recursion Available: Wenn der Nameserver rekursive Anfragen unterstützt, wird das Bit auf 1 gesetzt
(zero)	3	Dieses Flag muss immer auf Null gesetzt werden
Rcode	4	Beinhaltet einen Rückgabe-Code: 000 bedeutet keine Fehler, 111 bedeutet, dass der Domainname nicht existiert. Nur autoritative Nameserver können Fehlercodes zurückliefern.
Number of Questions	16	Anzahl der DNS-Anfragen im Question-Feld
Number of Answer RRs	16	Anzahl der Resource Records im Answer-Feld
Number of Authority RRs	16	Anzahl der Recource Records im Authority-Feld
Number of Additional RRs 1	6	Anzahl der Resource Records im Additional-Feld
Questions	200	Enthält die DNS-Anfrage

Die Felder *Answers* und *Authority* enthalten einen oder mehreren Resource Records mit einer Grösse von je 128 Bit. Das Feld *Additional Information* wird in der Regel nicht verwendet.

Beispielkonfiguration DDNS

/etc/bind/named.conf.local

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

zone "tsbe.intern" {
    type master;
    file "/var/cache/bind/db.tsbe.intern";
    allow-update {key "rndc-key";};
};

zone "tsbe.dmz" in {
    type master;
    file "/etc/bind/db.tsbe.dmz";
};

zone "210.168.192.in-addr.arpa" in {
    type master;
    file "/var/cache/bind/db.192.168.210";
    allow-update {key "rndc-key";};
};

zone "220.168.192.in-addr.arpa" in {
    type master;
    file "/etc/bind/db.192.168.220";
};
```

/etc/dhcp3/dhcpd.conf

```
# DHCP-Konfiguration fuer tsbe.intern

# gemeinsame Optionen
include "/etc/dhcp/rndc.key";
ddns-updates on;
ddns-update-style interim;
ddns-domainname "tsbe.intern";
update-static-leases on;

option domain-name "tsbe.intern";
option domain-search "tsbe.intern";
option domain-name-servers ns.tsbe.dmz;
option routers 192.168.210.1;
option broadcast-address 192.168.210.255;
```

```
default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

# statische Adressen tsbe.intern
subnet 192.168.210.0 netmask 255.255.255.0 {
    range 192.168.210.100 192.168.210.200;
}

host tsbe1 {
    hardware ethernet 00:50:56:30:21:61;
    fixed-address 192.168.210.61;
}

zone 210.168.192.in-addr.arpa {
    primary 192.168.220.12;
    key "rndc-key";
}

zone tsbe.intern {
    primary 192.168.220.12;
    key "rndc-key";
}
```

Damit bind dann die Zonen updaten kann, müssen diese wie folgt angepasst werden:

/var/cache/bind/db.tsbe.intern

```
$TTL    604800
@       IN      SOA     ns.tsbe.dmz. root.tsbe.dmz. (
                     2009102901         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@          IN           NS           ns.tsbe.dmz.

vmls4      IN     A     192.168.210.60
vmlf1      IN     A     192.168.210.1
```

/var/cache/bind/db.192.168.210

```
$TTL       604800
@          IN           SOA    ns.tsbe.dmz. root.tsbe.dmz. (
                       2009100301       ; Serial
                           604800       ; Refresh
                            86400       ; Retry
                          2419200       ; Expire
                           604800 )     ; Negative Cache TTL
;
@          IN           NS           ns.tsbe.dmz.

60         IN           PTR          vmls4.tsbe.intern.
1          IN           PTR          vmlf1.tsbe.intern.
```

Schlüssel einrichten und Berechtigungen anpassen

Kopieren des Schlüssels /etc/bind/rndc.key in das Verzeichnis /etc/dhcp oder direct mit scp. Passen Sie anschliessend die Berechtigung an.