Fachhochschule Aargau
Nordwestschweiz

Domain Name System

Fachhochschule Aargau

Departement Technik

Studiengang Informatik

Modul IT System Management I

Betreuender Dozent: Dr. H. von Fellenberg

Student: Luca Marinucci
Version: 1.3
Revision: 20. Dezember 2004

Inhaltsverzeichnis

1 Einführung

Das Domain Name System ist eine verteilte Datenbank, die den Namensraum im Internet verwaltet, und Namen in Adressen umsetzt, oder umgekehrt („Reverse Lookup“).

Es wurde 1983 von Paul Mockapetris entwickelt und im RFC 882 spezifiziert, welches später revidiert, und als Basis für die RFCs 1034 und 1035 diente. Das DNS ersetzt die `/etc/hosts` Datei.

Das DNS zeichnet sich durch folgende Eigenschaften aus:
- dezentrale Verwaltung
- hierarchische Strukturierung des Namensraums in Baumform
- Eindeutigkeit der Namen
- Erweiterbarkeit

Wegen seiner existenziellen Wichtigkeit für das Internet, und weil Ausfälle bzw. Angriffe auf dieses System so dramatische, wirtschaftliche Auswirkungen haben können, wurde DNS um zwei Sicherheitsfunktionen erweitert. Die erste Erweiterung ist TSIG (Transaction Signatures), welches auf einem System mit symmetrischen Schlüsseln beruht. Die zweite ist DNSSEC[1]. Hierbei handelt es sich um ein Public Key-Verfahren, mit dem nahezu alle DNS-Sicherheitsanforderungen erfüllt werden können.

2 Komponenten

Das DNS besteht im Grunde aus drei Grundelementen, die hier kurz vorgestellt werden.

2.1 Domänennamensraum

Der Domänennamensraum hat eine baumförmige Struktur, dessen Knoten und Blätter als „Labels“ bezeichnet werden. Die Wurzel des Baumes ist ein Punkt. Auf der nächsten Stufe folgen die sog. „Top Level Domains“, kurz TLD. Traversiert man den Baum von unten nach oben zur Wurzel hin, und separiert die Labels durch einen Punkt, ergibt dies den vollständigen Domänennamen (Fully Qualified Domain Name, FQDN). Deshalb enthält der genaue, formale Domänenname am Ende eigentlich einen Punkt. Dieser kann aus praktischen Gründen aber weggelassen werden.

```
.
|---- ch
|     |---- switch
|     |---- fh-aargau
|            |---- cs
|                  |---- itsm
|---- net
|     |---- root-servers
```

Der FQDN der Domäne `itsm ist itsm.cs.fh-aargau.ch.` .

2.2 Nameserver

Ein Nameserver enthält Informationen über einen oder mehrere Teile des Domänenbaums. Er wird durch eine höhere Ebene im Baum delegiert, und ist für den Namensraumes unterhalb der delegierten Ebene zuständig. Dabei beantwortet er Anfragen über seinen Zuständigkeitsbereich über den UDP-Port 53. Für Teile seines eigenen Namensraums kann er ebenfalls weitere Nameserver delegieren.

Es wird grundsätzlich zwischen Primary- und Secondary Nameserver (bzw. Master und Slave) unterschieden, wobei beide „autoritative“ Informationen retournieren. Im Gegensatz dazu stehen „nicht-autoritative“ Nameserver, die ihre Daten aus zweiter oder dritter Hand erhalten

1 http://www.onlamp.com/pub/a/onlamp/2004/10/14/dnssec.html

haben, und somit keine gesicherten Angaben über den Namensraums geben können. Diese werden auch als „caching only“ Nameserver bezeichnet.

Secondary Nameserver dienen als Backup und erhalten ihre Daten von ihrem Master, dem Primary Nameserver. Dieser Vorgang wird als „Zonentransfer“ bezeichnet, und wird über den TCP-Port 53 abgewickelt.

2.2.1 Zone

Eine Zone enthält die Teile des Domänenbaums, für dein Nameserver autoritative Informationen besitzt. Sie wird durch einen Primary Nameserver verwaltet und aus Redundanz-Zwecken auf mindestens einen oder mehreren Secondary Nameserver gespiegelt.

Die Objekte einer Zone, z.B. Mailserver oder Netzwerkdrucker, werden in einer Zonendatei aufgeführt, deren Original auf dem Primary Nameserver liegt. In regelmässigen Abständen wird diese per Zonentransfer zu den Secondary Nameservern übertragen.

Eine Zone kann eine gesamte Domäne umfassen. Normalerweise werden Subdomänen aber in eigenen Zonen verwaltet.

Beispiel FH-Aargau:

```
.
|---- ch
|     |---- fh-aargau
|             |---- cs (Zone)
|                   |---- loki            Primary NS
|                   |---- freya           Secondary NS
|                   |---- itsm (Zone bzw. Subdomäne von cs)
|                         |---- itsm      Primary NS
|                         |---- bs7800e   Secondary NS
```

Die Domäne `cs.fh-aargau.ch.` besitzt die Subdomäne mit dem Label `itsm`, welche in einer eigenen Zone durch den Primary Nameserver `itsm.itsm.cs.fh-aargau.ch.` verwaltet wird. Sie delegiert ihm diese Zone. Die Objekte dieser Zone sind die Rechner `itsm` und `bs7800e`. Die Zonendatei auf dem Primary Nameserver `loki.cs.fh-aargau.ch.` enthält deswegen folgende Einträge (Resource Records):

```
@                 1d   IN SOA loki oser.fh-aargau.ch. (
                                  1       ; serial
                                  2H      ; refresh
                                  6H      ; retry
                                  2w1d    ; expire
                                  1H )    ; minimum

@                      IN NS  loki.cs.fh-aargau.ch.
@                      IN NS  freya

localhost              IN A   127.0.0.1
loki                   IN A   147.86.130.1
freya                  IN A   147.86.130.6
```

Die Zonendatei von `itsm.cs.fh-aargau.ch.` könnte wie folgt aussehen:

```
@                1d   IN SOA itsm fellenberg.fh-aargau.ch. (
                                  1       ; serial
                                  2H      ; refresh
                                  6H      ; retry
                                  2w1d    ; expire
                                  1H )    ; minimum

@                     IN NS  itsm.itsm.cs.fh-aargau.ch.
@                     IN NS  bs7800e

localhost             IN A   127.0.0.1
itsm                  IN A   147.86.130.53
bs7800e               IN A   147.86.130.78
```

@ ist dabei der Name der Zone und kann einfachheitshalber weggelassen werden. In diesem Fall steht es für `cs.fh-aargau.ch.` bzw. `itsm.cs.fh-aargau.ch. .`

Die Regeln, wie eine Zonendatei auszusehen hat, sind im Anhang (Siehe Anhang I – Regeln Zonendatei) beschrieben.

2.2.2 Reverse Lookup

Um DNS Spoofing-Angriffe verhindern zu können, muss ein Nameserver, wie in der Einführung bereits erwähnt wurde, auch Adressen in Namen umwandeln können. Der bisher beschriebene Domänennamensraum wird deshalb um einen weiteren Namensraum, den der IP-Adressen, erweitert. Das bedeutet auch, dass stets zwei Dateien für eine Zone erstellt werden müssen, eine für den Domänen-, und eine für den Adressnamensraum.

Diese Domäne wird aus historischen Gründen `arpa` bezeichnet. Da Internet-Adressen als `in-addr` bezeichnet werden, heisst die volle Domäne `in-addr.arpa.` (auch hier wieder mit schliessendem Punkt). Dieser Baum enthält schliesslich alle IP-Adressen. Die Besonderheit ist nun, dass die Teile der Netzwerk-Adresse zwar rückwärts, also wie wir das bisher bei den Namen taten, notiert werden, aber auf den ersten scheint das ziemlich verwirrend. Wollen wir beispielsweise eine Zone für das Netzwerk `192.168.100.` einrichten, so sieht die korrekte Bezeichnung `100.168.192.in-addr.arpa.` aus.

Die Reverse-Zonendatei für `itsm.cs.fh-aargau.ch.` , also `130.86.147.in-addr.arpa.` :

```
@               1d   IN SOA itsm.itsm.cs.fh-aargau.ch fellenberg.fh-aargau.ch. (
                                  1       ; serial
                                  2H      ; refresh
                                  6H      ; retry
                                  2w1d    ; expire
                                  1H )    ; minimum

@                     IN NS  itsm.itsm.cs.fh-aargau.ch.
@                     IN NS  bs7800e.itsm.cs.fh-aargau.ch.

53                    IN PTR itsm.itsm.cs.fh-aargau.ch.
78                    IN PTR bs7800e.itsm.cs.fh-aargau.ch.
```

2.2.3 Localhost

Eine weitere Besonderheit ist der Begriff `localhost`. Ein Nameserver ist auch verantwortlich, dass dieser richtig aufgelöst wird. Es gibt verschiedene Möglichkeiten, dieses Problem zu lösen. Die meist verbreitetste ist, `localhost` als eine Top Level Domain zu behandeln, und stets eine solche Zone selber zu definieren.

```
$TTL 2w1d
@       IN      SOA     localhost.      root.localhost. (
                            1               ; serial
                            2H              ; refresh
                            30M             ; retry
                            2w1d            ; expiry
                            1H )            ; minimum

@       IN      NS      localhost.
@       IN      A       127.0.0.1
```

2.2.4 Root Server

Damit ein Nameserver Informationen über andere Teile des Namensraumes finden kann, werden ihm Daten über die sog. „Root Server“ in Form einer statischen Datei hinterlegt. Derzeit gibt es 13 solcher Root Server, die auf der ganzen Welt verteilt, und mit den Namen 'A' bis 'M' bezeichnet sind, und z.T. aus Serververbünden bestehen. Sie stellen die oberste Instanz des DNS dar. Die Root Server 'B' bis 'M' beziehen ihre Informationen jeweils vom Root Server 'A', welcher von VeriSign betrieben wird.

2.3 Resolver

Resolver sind Programme, die Informationen aus dem Nameserver abrufen können. Sie bilden die Schnittstelle zum Nameserverdienst, und sind entweder eigene Programme (nslookup, dig), oder sie sind in Applikationen (Bsp. Browser) eingebettet.

Ein Resolver arbeitet entweder iterativ oder rekursiv. Bei iterativen Anfragen retourniert der Nameserver entweder einen Resource Record oder einen anderen Nameserver, der vom Resolver weiterverwendet werden kann. Bei der rekursiven Anfrage hingegen erhält der Resolver entweder den endgültigen Resource Record, oder nichts.

3 Praktisches Beispiel

Als praktisches Beispiel wird nun die Zone `marinucci.itsm.cs.fh-aargau.ch.` mit `bind` eingerichtet. Als Primary Nameserver der Zone dient der Labor-Rechner `da8250j`, der zugleich als Secondary der Zone `boss.itsm.cs.fh-aargau.ch.` Fungiert. Dessen Primary ist `bs7550c`.

3.1 Gentoo Installation von bind

```
# emerge bind
```

Als erstes werden die benötigten Zonendateien erstellt:

`/var/bind/named.ca` (für die Root Server)

```
.                       518400  IN      NS      A.ROOT-SERVERS.NET.
.                       518400  IN      NS      H.ROOT-SERVERS.NET.
.                       518400  IN      NS      C.ROOT-SERVERS.NET.
.                       518400  IN      NS      G.ROOT-SERVERS.NET.
.                       518400  IN      NS      F.ROOT-SERVERS.NET.
.                       518400  IN      NS      B.ROOT-SERVERS.NET.
.                       518400  IN      NS      J.ROOT-SERVERS.NET.
.                       518400  IN      NS      K.ROOT-SERVERS.NET.
.                       518400  IN      NS      L.ROOT-SERVERS.NET.
.                       518400  IN      NS      M.ROOT-SERVERS.NET.
.                       518400  IN      NS      I.ROOT-SERVERS.NET.
.                       518400  IN      NS      E.ROOT-SERVERS.NET.
.                       518400  IN      NS      D.ROOT-SERVERS.NET.

A.ROOT-SERVERS.NET.     3600000 IN      A       198.41.0.4
H.ROOT-SERVERS.NET.     3600000 IN      A       128.63.2.53
C.ROOT-SERVERS.NET.     3600000 IN      A       192.33.4.12
G.ROOT-SERVERS.NET.     3600000 IN      A       192.112.36.4
F.ROOT-SERVERS.NET.     3600000 IN      A       192.5.5.241
B.ROOT-SERVERS.NET.     3600000 IN      A       128.9.0.107
J.ROOT-SERVERS.NET.     3600000 IN      A       192.58.128.30
K.ROOT-SERVERS.NET.     3600000 IN      A       193.0.14.129
L.ROOT-SERVERS.NET.     3600000 IN      A       198.32.64.12
M.ROOT-SERVERS.NET.     3600000 IN      A       202.12.27.33
I.ROOT-SERVERS.NET.     3600000 IN      A       192.36.148.17
E.ROOT-SERVERS.NET.     3600000 IN      A       192.203.230.10
D.ROOT-SERVERS.NET.     3600000 IN      A       128.8.10.90
```

/var/bind/pri/localhost.zone

```
$TTL 2w1d
@       IN      SOA     ns.localhost.      root.localhost. (
                            1               ; serial
                            2H              ; refresh
                            30M             ; retry
                            2w1d            ; expiry
                            1H )            ; minimum

                IN      NS      ns
localhost       IN      A       127.0.0.1
```

/var/bind/pri/127.zone

```
$ORIGIN 127.in-addr.arpa.
$TTL 2w1d
@       IN       SOA    localhost.       root.localhost. (
                            1               ; serial
                            2H              ; refresh
                            30M             ; retry
                            2w1d            ; expiry
                            1H )            ; minimum

        IN       NS     localhost.
*       IN       PTR    localhost.
```

/var/bind/pri/mari.zone

```
$ORIGIN marinucci.itsm.cs.fh-aargau.ch.
@                1d   IN SOA da8250j ia02mari.stud.fh-aargau.ch. (
                           1               ; serial
                           2H              ; refresh
                           30M             ; retry
                           2w1d            ; expiry
                           1H )            ; minimum

                      IN NS  da8250j
                      IN NS  bs7550c.boss.itsm.fh-aargau.ch. ; Secondary

localhost             IN A   127.0.0.1
da8250j               IN A   147.86.130.210
```

Nun wird bind so konfiguriert, dass er die erstellten Zonendateien zuordnen kann.

/etc/bind/named.conf

```
options {
    directory „/var/named“;
    listen-on-v6 { none; };
    listen-on { 147.86.130.210; };
};
// Zones
zone „.“ IN {
    type hint;
    file „named.ca“;
};
zone „marinucci.itsm.cs.fh-aargau.ch“ IN {
    type master;
    file „pri/mari.zone“;
    allow-transfer { 147.86.130.42; };
};
zone „localhost“ IN {
    type master;
    file „pri/localhost.zone“;
};
zone „127.in-addr.arpa“ IN {
    type master;
    file „pri/127.zone“;
};
zone „boss.itsm.cs.fh-aargau.ch“ IN {
    type slave;
    file „sec/boss.zone“;
    masters { 147.86.130.42; };
};
```

Bevor der Nameserver gestartet wird, sollten die Konfigurationsdateien resp. die Zonendateien auf Korrektheit überprüft werden:

```
# named-checkconf /etc/bind/named.conf
# named-checkzone boss.itsm.cs.fh-aargau.ch. /var/bind/sec/boss.zone
# named-checkzone marinucci.itsm.cs.fh-aargau.ch. /var/bind/pri/mari.zone
```

Falls keine Fehler aufgetaucht sind, kann der Nameserver nun gestartet werden. Mit dieser Konfiguration sollte ein Nameserver aber nicht für Produktionszwecke eingesetzt werden. Hierfür sollten sowohl TSIG oder DNSSEC in Erwägung gezogen werden, wie auch das Ausführen von `bind` in einer chroot-Umgebung. Bei Änderungen an den Zonendateien darf nicht vergessen werden, die Seriennummer zu inkrementieren. Am besten wird dabei das Datum in der Form `YYYYMMDD` als Nummer verwendet.

```
# /etc/init.d/named start
# rc-update add named default
```

Im Verzeichnis `/var/bind/sec` erstellt bind nach dem Start die Zonendatei `boss.zone`. Falls `bind` diese Datei nach dem Starten nicht lesen kann (`tail -f /var/log/messages`), sollte der Benutzer des Verzeichnisses auf `named` geändert werden:

```
# chown -R named:root /var/bind
```

Nun können wir uns mit einem Resolver, z.B. `nslookup`, mit dem Nameserver verbinden, und überprüfen, ob er wirklich funktioniert:

```
# nslookup
> server 147.86.130.210
Default server:  147.86.130.210
Address: 147.86.130.210#53
> da8250j.marinucci.itsm.cs.fh-aargau.ch.
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   da8250j.marinucci.itsm.cs.fh-aargau.ch
Address: 147.86.130.210
```

Als letztes können wir nun noch unsere `/etc/resolv.conf` entsprechend anpassen:

```
nameserver 127.0.0.1
nameserver 147.86.130.1
nameserver 147.86.130.6
nameserver 147.86.130.53
nameserver 147.86.130.56
search cs.fh-aargau.ch itsm.cs.fh-aargau.ch marinucci.itsm.cs.fh-aargau.ch
```

4 Anhang

Anhang I - Regeln Zonendatei

Regel 1
Leerzeilen sind zulässig.

Regel 2
Kommentare werden durch ";" eingeleitet.

Regel 3
Soll ein Resource Record auf mehrere Zeilen verteilt werden, so müssen Klammern verwendet werden.

Beispiel:

```
example.com.       1800  IN  SOA  ns1.example.com. mailbox.example.com. (
                                                100   ; Seriennummer
                                                300   ; Refresh Time
                                                100   ; Retry Time
                                                6000  ; Expire Time
                                                1800  ; TTL )
```

Regel 4
Erscheint Name der Zonendatei - der so genannte Origin - ohne Extension isoliert, so darf er durch ein "@" ersetzt werden.

Beispiel:

```
@                  1800  IN  SOA  ns1.example.com. mailbox.example.com. (
                                                100   ; Seriennummer
                                                300   ; Refresh Time
                                                100   ; Retry Time
                                                6000  ; Expire Time
                                                1800  ; TTL )
```

Regel 5
Erscheint der Origin (Name der Zonendatei ohne Extension) am Ende eines Namens, so darf er weggelassen werden.

Beispiel:

```
@                  1800  IN  SOA  ns1 mailbox (
                                                100   ; Seriennummer
                                                300   ; Refresh Time
                                                100   ; Retry Time
                                                6000  ; Expire Time
                                                1800  ; TTL )
```

Regel 6
Haben zwei oder mehr aufeinanderfolgende RRs den gleichen Namen, so braucht nur der erste angegeben zu werden.

Regel 7
Das Klassenfeld "IN" muss nur beim ersten RR angegeben werden.

Beispiel:

```
@                   1800  IN  SOA ns1 mailbox (
                                              100   ; Seriennummer
                                              300   ; Refresh Time
                                              100   ; Retry Time
                                              6000  ; Expire Time
                                              1800  ; TTL )
                    1800   NS  ns1        ; der Name darf weggelassen werden

ns1                 1800   A   172.27.182.17
www                 1800   A   192.168.1.2
xxx.external.net.   1800   A   1.2.3.4
```

Regel 8
Ist in einem RR kein TTL vorhanden, so wird der letzte in der Zonendatei vorher vorhandene TTL-Wert verwendet. Ist vorher kein TTL vorhanden, so wird der Wert aus dem SOA Resource Record genommen.

Beispiel:

```
@               IN  SOA  ns1 mailbox 100 300 100 6000 1234
                    NS   ns1               ; TTL=1234 aus SOA

ns1                 A    172.27.182.17     ; TTL=1234 aus SOA
www      20         A    192.168.1.2       ; ab hier gilt TTL=20
xxx.external.net.   A    1.2.3.4           ; TTL=20 aus vorherigem Eintrag
```

Regel 9
Standard-Origin ist der Dateiname ohne Extension. Mit der $ORIGIN-Anweisung können beliebige andere Origins definiert werden. Ein neu definierter Origin ist für alle folgenden Zeilen (bis zur nächsten Origin-Anweisung) gültig.

Beispiel:

```
@         IN  SOA  ns1 mailbox 100 300 100 6000 1800
              NS   ns1

ns1           A    172.27.182.17
www           A    192.168.1.2

$ORIGIN external.net.
xxx           A    1.2.3.4
```