DNS-Server an einem Standort konfigurieren

Automatisches Erzeugen der Konfiguration

Im Folgenden sei ein erstes Script, „zones2conf.sh“, vorgestellt. Es dient zur Erzeugung der Subzonenkonfiguration für den BIND DNS-Server an einem Digipeater. Am Anfang der Scriptdatei müssen einige Parameter konfiguriert werden.

Editieren und Anpassen

```
# --- CONFIGURE ME
# Am I Hub?
# NO?
#I_AM_HUB=1
I_AM_HUB=0

# What is your DNS-Hub? [nord|sued|mitte|west|ost]
MY_HUB=ost
# IP of my DNS serer
MY_DNS_IP=44.130.36.200

# Hubs which are too far away to reach directly for notify
NOT_NOTIFY_HUBS_TOO_FAR="nord"

FILEPATH=/var/named/maps

HUB_NORD="44.130.0.100; // db0hht (GrossRegion-NORD)"
HUB_SUED="44.130.60.100; // db0fhn (GrossRegion-SUED)"
HUB_WEST="44.130.146.101; // db0res-svr (GrossRegion-WEST)"
HUB_OST="44.130.90.100; // db0tud (GrossRegion-OST)"
HUB_MITTE="44.130.14.100; // db0smg (GrossRegion-MITTE)"
[...]
```

Das Script wird gestartet mit:

```
# /etc/bind/bin/zone2conf.sh < /etc/bind/etc/zones-hub-de.txt >
/etc/bind/zones-region-de-ampr-org.conf
```

Es liest dann die Konfigurationsdatei „zones-hub-de.txt“ ein und generiert daraus eine Datei „zones-region-de-ampr.org.conf“. Sie enthält Einträge in folgender Form:

```
zone "stgt.de.ampr.org" {
type slave;
file "/var/named/maps/stgt.de";
masters {
44.130.48.23;
```

```
44.130.146.101; // db0res-svr (GrossRegion-WEST)
44.130.90.100; // db0tud (GrossRegion-OST)
44.130.14.100; // db0smg (GrossRegion-MITTE)
};
also-notify {
44.130.146.101; // db0res-svr (GrossRegion-WEST)
44.130.90.100; // db0tud (GrossRegion-OST)
44.130.14.100; // db0smg (GrossRegion-MITTE)
};
allow-notify {
44.130.48.23;
44.130.0.100; // db0hht (GrossRegion-NORD)
44.130.146.101; // db0res-svr (GrossRegion-WEST)
44.130.90.100; // db0tud (GrossRegion-OST)
44.130.14.100; // db0smg (GrossRegion-MITTE)
};
};
```

Dieses Verfahren erspart eine Menge Tipp-Arbeit und vermeidet Fehler (einen der vielen „;“ vergisst man gerne..). Bei ca. 138 Einträgen ist das ein nicht zu verachtender Vorteil ;)

Die Datei „zones-hub-de.txt“ wird auf einem der Hubs gepflegt und täglich auf den anderen Hubs synchronisiert. Von „ihrem“ nahe gelegenen Hub können ihrerseits die Regions-DNS-Server diese Datei (regelmäßig, auch automatisierbar z.B. mit rsync) beziehen.

Das Script kann, wie man an den Einstellungen sieht, für einen Hub oder für einen Regions-Server vorkonfiguriert werden. Damit lässt sich der komplette Prozess der Konfiguration und Wartung (also das Aktuell halten der bekannten Zonen und Regions-NS) des DNS automatisieren, auf den Hubs wie auch auf Servern in den Regionen. Bind9 erlaubt es, die Konfiguration in mehrere Dateien zu fächern. So lassen sich die mit obigem Script erzeugten und in einer separaten Datei abgelegten Zonen-Definitionen durch den folgenden Eintrag in der „named.conf“ von den ggf. lokal nötigen Konfigurations-Parametern abspalten:

```
include "/etc/bind/zones-region-de-ampr-org.conf";
```

- Ein Beispiel für eine named.conf (s. Anhang) Fix Me!

automatische Aktualisierung der Zonendateien

Zur automatisierten Aktualisierung bietet sich ein Cronjob an.

```
/etc/crontab:
15 03 * * * root /usr/local/etc/update-ampr-dns.sh
```

Hier ein Beispiel-Shell-Script „update-ampr-dns.sh“:

```
#!/bin/sh
old_sum=$(md5sum /etc/named/etc/zones-hub-de.txt | awk '{print $1}')
# update zone definitions
/usr/bin/rsync -qaz --timeout=600 --partial rsync://dl-ost.ampr.org/ampr-
```

```
dns/lib/zones-hub-de.txt /etc/named/etc/
# host unreachable?
if [ $? -gt 0 ]; then exit 1; fi
new_sum=$(md5sum /etc/named/etc/zones-hub-de.txt | awk '{print $1}')
# no change? - done
if [ "$old_sum" != "$new_sum" ]
then
# generate new zone definitions
/etc/bind/bin/zone2conf.sh < /etc/bind/etc/zones-hub-de.txt >
/etc/bind/zones-region-de-ampr-org.conf
# make named learn the changes
/etc/init.d/bind reload
fi
# done
```

Danach

```
$ chmod 755 /usr/local/etc/update-ampr-dns.sh
```

nicht vergessen.

Verwaltung der eigenen Region

Im Beispiel von Hub-Süd ist db0fhn „master“ für die Region nbg.de.ampr.org und die PTR (Rückwärtsauflösungen) für 44.130.60.x (60.130.44.in-addr.arpa). Damit das zuvor vorgestellte Script, welches die Regionen und Dateien definiert, nicht angepasst werden muss, lässt sich für die eigene Region unter Unix/Linux elegant mit „symbolischen Links“ arbeiten, wenn der DNS-Server nicht in einer „chroot-Umgebung“ arbeitet:

```
/var/named/maps:
lrwxrwxrwx 1 root root 23 Feb 22 23:13 nbg-60.de.rev ->
/etc/bind/nbg-60.de.rev
lrwxrwxrwx 1 root root 16 Feb 22 23:12 nbg.de -> /etc/bind/nbg.de
```

Die eigentlichen Dateien in /etc/bind/nbg*:

```
-rw-rw-r-- 1 dl9nec dnsadm 5436 Dec 2 15:54 nbg-60.de.rev
-rw-rw-r-- 1 dl9nec dnsadm 12135 Dec 2 15:54 nbg.de
```

In unserem Beispiel ist der Nutzer dl9nec (Regional-Koordinator nbg) in der Unix-Gruppe dnsadm. Er und andere Mitglieder dieser Gruppe können diese Dateien dann ändern, wenn er einen shell-Account auf dem Server besitzt. „rndc“ ist ein Werkzeug, das mit dem laufenden named kommuniziert. Dieses Programm liegt der named Installation bei.

Ein Schlüssel regelt die Zugangskontrolle. Er ist in der Datei „/etc/rndc.key“ bzw. „/etc/named/rndc.key“ (variiert mal wieder je nach Distribution) gespeichert. Versieht man die Datei mit 640 Rechten für root.dnsadm, dann kann der Regional-Koordinator seine Änderungen dem Nameserver mitteilen, ohne Administrations-Rechte erlangen und den named neu starten zu müssen:

```
$ rndc reload nbg.de.ampr.org
$ rndc reload 60.130.44.in-addr.arpa
```

Anhang I dokumentiert den Aufbau eines Zonefiles. Fix Me!

Die weltweite Bereitstellung der Daten

Die Daten kommen aus den Regionen zu ihren assoziierten Hubs, entweder per „notify“ angestoßen, oder über den regulären AXFR. Ein eigener DNS-Server ist nicht zwingend nötig. Er ist eine Erleichterung, mag aber in einer Region mit nur zehn Teilnehmern wie mit Kanonen auf Spatzen geschossen wirken. Deshalb bietet die DL-IP-Koordination in Zusammenarbeit mit den Betreibern der Hubs auch die Möglichkeit, die Zonendaten direkt auf dem zuständigen Hub zentral zu verwalten und zu aktualisieren.

Allerdings schließen sich die Verfahren „eigenes Zonefile auf einem Regions-Server“ und „zentrale Verwaltung“ gegenseitig aus. D.h. letztere Möglichkeit kann nur von Regionen genutzt werden,

- deren Zonefile auf einem der Hubs liegt.
- die nicht am DNS-Regionalzonenprojekt teilnehmen.

Die Zonetransfer-Variante mit eigenem Regional-DNS-Server ist dennoch die von uns präferierte Methode, weil sie dem Regionalkoordinator auf „seinem“ Digipeater vollständige Kontrolle über seine Daten und den Verteilungsprozess gibt.

Die DNS-Daten werden zwischen den Hubs mittels Zonetransfer und aktivem „Notify“ abgeglichen. Somit verfügen alle fünf Hubs über die selben Datenbestände.

Einer der Hubs ist so konfiguriert, dass er regelmäßig die Soll-Datenstände (das sind die Zonefiles, die über AXFR von den Regional-DNS den Hub erreichten) mit den Ist-Daten (DL-Einträge auf ucsd.edu) vergleicht. Treten keine Konflikte auf und lassen sich die Daten in Form von

```
[irgendwas.]<call>.<region>.de.ampr.org
```

auf das flachere

```
[irgendwas.]<call>.ampr.org
```

abbilden, dann steht deren weltweiter Veröffentlichung nichts mehr im Wege.

Im Umkehrschluss ziehen sich die Hubs regelmäßig die Daten für die komplette Zone ampr.org / 44.0.0.0/8. Dabei wird die Datei mit den Vorwärtsauflösungen („ampr.org“), bevor sie in den DNS-Server geladen wird, über ein Script angepasst:

Existiert ein Eintrag in der ampr.org-Datei und ein korrespondierender in der Datei einer Region, dann wird in der ampr.org-Datei der „IN A“ Eintrag ersetzt durch „IN CNAME <call>.<region>“. Beispiel:

```
db0res-svr.ampr.org. 432000 IN CNAME db0res-svr.rr.de.ampr.org.
db0res-svr.rr.de.ampr.org. 864000 IN A 44.130.146.101
```

Diese Information kann nützlich sein für

- domain-basiertes Mail-Routing.
- effizientes Weiterleiten von Anfragen in einem http-Proxy-Verbund.
- beim Generieren von IP-Routing Einträgen.

Die an die hiesigen Bedingungen angepasste ampr.org kann ebenfalls per Zonetransfer aus den Regionen von den Hubs bezogen werden.

Problematik der Umsetzung in die ampr.org Zone

Wichtigste Vorbedingung:

Die Einträge auf ucsd.edu sollten der Form

```
[irgendwas.][irgendwas-]<call>[-irgendwas].ampr.org
```

entsprechen. Beispielsweise würde

```
dhcp1.beispielregion.de.ampr.org
```

umgewandelt zu

```
dhcp1.ampr.org
```

Dieser Eintrag kann also nicht übernommen werden.

Doppel-Einträge:

I) Gleicher Name (z.B. nach Umzug in eine andere Region)

```
da0aaa.region254.ampr.org. IN A 44.130.254.1
da0aaa.region255.ampr.org. IN A 44.130.255.1
```

würde zu

```
da0aaa.ampr.org IN A 44.130.254.1
da0aaa.ampr.org IN A 44.130.255.1
```

Das ist zwar ein, aus DNS-Sicht, gültiger Eintrag. Jedoch wird es schwierig, diesen Host zu erreichen. Ein DNS liefert beim Versuch, da0aaa.ampr.org aufzulösen, mal die eine, und mal die andere IP-Adresse aus.

II) Gleiche IP-Adresse

```
da1aaa.region255.ampr.org. IN A 44.130.255.1
dl1bbb.region255.ampr.org. IN A 44.130.255.1
```

würde bei Rückwärtsauflösung zu

```
1.255.130.44.in-addr.arpa IN PTR da1aaa.region255.ampr.org.
```

```
1.255.130.44.in-addr.arpa IN PTR dl1bbb.region255.ampr.org.
```

Die Eindeutigkeit des Rufzeichens ginge dabei verloren.