

Namerserver mittels BIND9 und Debian

[3. Februar 2014](#)

In dieser Anleitung werden wir 2 BIND-DNS-Server, einen als Master und den anderen als Slave-Server installieren. Aus Gründen der Sicherheit werden wir bind9 in seinem eigenen Gefängnis „chroot“ festsetzen.

Mit zwei Servern für eine Domain ist eine häufig verwendete Installation und um Ihre eigene Domain hosten zu können, müssen Sie mindestens 2 Domain-Server haben. Unser Setup wird Debian Sarge 3.1 (stable) für seine Basis verwenden. Eine einfache und saubere Installation wird genug sein, da wir die benötigten Pakete mit dieser Anleitung installieren.

In dieser Anleitung werde ich die fiktive Domain "linux.lan" verwenden. Die Nameserver werden die IP's 192.168.254.1 und 192.168.254.2 verwenden.

Basis Installation:

Spoiler anzeigen

Quellcode

1. apt-get update && apt-get upgrade

Wir benötigen auch einige Debian-Werkzeuge, da müssen wir das Quellpakete herunterladen:

Quellcode

1. apt-get install devscripts

Nun können wir mit der Installation fortfahren. Zunächst installieren wir ein Self-Test Mirror auf dem NameServer:

Quellcode

1. cd ~/a/dir/for/lsb-base/
2. apt-get -y build-dep lsb-base
3. apt-get source lsb-base -b
4. dpkg-i lsb-base*.deb

Jetzt installieren wir Bind:

Quellcode

1. cd ~/a/dir/for/bind9
2. apt-get -y build-dep bind9
3. apt-get source bind9 -b
4. dpkg -i *.deb

Konfiguration des Master-DN-Server

Quellcode

Copyright bleibt bei Teris Cooper und kann jederzeit über www.root-projekte.de verädert werden.



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

Um den chroot bedienen zu können, mittel Bind, benötigen wir noch eine Option in der Konfiguration:

Quellcode

1. `nano /etc/default/bind9`

```
OPTIONS="-u bind -t /var/lib/named"
```

Es wird nun als Benutzer 'bind' in '/var/lib/named' chroot ausgeführt.

Als nächstes werden wir die Spediteure Zeile in `/etc/bind/named.conf.options` unser dns von unserem ISP überein stimmen sollte, bearbeiten:

Quellcode

1. `nano /etc/bind/named.conf.options`

```
forwarders { 194.109.6.66; };
```

Dieser Schritt ist wichtig zur Nutzung des chroots für Jail:

Quellcode

1. `mkdir -p /var/lib/named/etc`
2. `mkdir /var/lib/named/dev`
3. `mkdir -p /var/lib/named/var/cache/bind`
4. `mkdir -p /var/lib/named/var/run/bind/run`
5. `mv /etc/bind /var/lib/named/etc`
6. `ln -s /var/lib/named/etc/bind /etc/bind`
7. `mknod /var/lib/named/dev/null c 1 3`
8. `mknod /var/lib/named/dev/random c 1 8`
9. `chmod 666 /var/lib/named/dev/*`
10. `chown -R bind:bind /var/lib/named/var/*`
11. `chown -R bind:bind /var/lib/named/etc/bind`

Bind hat jetzt seinen eigenen Platz für .Pid-Dateien und Konfigurationsdateien. Um die Dinge klar machen haben wir einen Symlink zurück nach `/etc/` gelegt.

Nun müssen wir noch die `/etc/init.d/syslogd` anpassen:

Quellcode

1. `nano /etc/init.d/syslogd`

```
SYSLOGD="-a /var/lib/named/dev/log"
```

Der letzte Schritt ist, um die ip des neu installierten DNS-Server (localhost) zu `/etc/resolv.conf` hinzuzufügen und um es benutzen zu können:

Copyright bleibt bei Teris Cooper und kann jederzeit über www.root-projekte.de verädert werden.



Quellcode

1. `echo "search linux.lan" > /etc/resolv.conf`
2. `echo "nameserver 127.0.0.1" >> /etc/resolv.conf`
3. `/etc/init.d/syslogd restart`
4. `/etc/init.d/bind9 start`

Einstellung der linux.lan Domain:

Spoiler anzeigen

Die Master-DNS-Server ist derzeit nur die Weiterleitungsanfragen an den Server Ihres ISP. Also, wir werden jetzt die Domain konfigurieren und lassen diese unseren neuen Server alle Anfragen bezüglich dieser Domäne zu behandeln. Fangen wir mit der Erstellung des Verzeichnisses, in dem wir die Zonendatei speichern. Diese Datei enthält alle Informationen über die Domain.

Quellcode

1. `mkdir /etc/bind/zones/`

Jetzt erzeugen wir die master_linux.lan:

Quellcode

1. `nano /etc/bind/zones/master_linux.lan`

```
$TTL 3D
```

```
@ IN SOA ns1.linux.lan. hostmaster.linux.lan. (
199802151 ; serial, todays date + todays serial #
8H ; refresh, seconds
2H ; retry, seconds
4W ; expire, seconds
1D ) ; minimum, seconds
;
TXT "Linux.LAN, serving YOUR domain :)"
NS ns1 ; Inet Address of name server
NS ns2
MX 10 mail ; Primary Mail Exchanger
localhost A 127.0.0.1
ns1 A 192.168.254.1
ns2 A 192.168.254.2
www CNAME ns1
```

Hier haben wir eine einfache Zone-Datei mit beiden Nameserver und einem WWW-Alias für ns1 erstellt. Nur für den Fall haben wir einen laufenden Apache auf ns1;)

Natürlich müssen wir dies unserem NameServer noch mitteilen:

Quellcode

1. `nano /etc/bind/named.conf.local`

Copyright bleibt bei Teris Cooper und kann jederzeit über www.root-projekte.de verädert werden.



```
zone "linux.lan" {  
type master;  
file "/etc/bind/zones/master_linux.lan";  
};
```

Nun können wir den DNS Server neustarten und ein Pingtest aufführen:

Quellcode

1. /etc/init.d/bind9 restart
2. ping ns1.linux.lan

In dieser Phase sollten Sie ein Arbeits- und nutzbare DNS-Server haben 😊

Konfiguration des Slave-DN-Servers

Spoiler anzeigen

Grundsätzlich verwendet der Slave das gleiche Grundsystem, wie wir im ersten Teil aufgebaut (kurz bevor wir die Zonendatei hinzugefügt) haben. Wir werden einige kleine Änderungen an Master und Slave hinzufügen, um sie zusammen zu arbeiten zu lassen. Die Zonen-Datei wird über das Netz durch Verschlüsselung übertragen werden. Sofern nicht anders angegeben, sind diese Befehle für den Slave-ONLY.

Erstellen des Zone-Verzeichnisses und erste Anpassung erzeugen:

Quellcode

1. mkdir /etc/bind/zones
2. nano /etc/bind/named.conf.options

```
dnssec-enable yes;
```

Jetzt brauchen wir einen sicheren Schlüssel. Dies wird eine Private und eine Schlüsseldatei zu generieren. Die Zeile "key =" in der Privat Datei stellt die hashkey.

Quellcode

1. dnssec-keygen -a hmac-md5 -b 128 -n host linux.lan

Nun fügen wir diesen Hashkey im Master und Slave Server hinzu, dazu müssen wir nochmals die named.conf auf beiden Servern anfassen und folgendes hinzufügen:

```
key "TRANSFER" {  
algorithm hmac-md5;  
secret "---HASHKEY---";  
};
```

Auf dem Master DNS fügen wir außerdem folgendes hinzu:

```
server 192.168.254.2 {  
keys {  
TRANSFER;
```

Copyright bleibt bei Teris Cooper und kann jederzeit über www.root-projekte.de verändert werden.



```
};  
};
```

Der Slave Server erhält zusätzlich diese Einstellung:

```
server 192.168.254.1 {  
keys {  
TRANSFER;  
};  
};
```

Ebenso erstellen wir auch wieder, eine Zone, diesmal jedoch die "slave_linux.lan"

Quellcode

1. nano /etc/bind/zones/slave_linux.lan

```
zone "linux.lan" {  
type slave;  
file "/etc/bind/zones/slave_linux.lan";  
masters { 192.168.254.1; };  
allow-notify { 192.168.254.1; };  
};
```

Zum schluss fügen wir auf beiden Servern in die "/etc/bind/named.conf" noch den CronJob "rndc" hinzu:

Quellcode

1. nano /etc/bind/named.conf

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

Desweiteren sollten wir noch den Zeitsynchronisator installieren:

Quellcode

1. apt-get -y install ntpdate

Zu guter letzt, starten wir beide NameServer (Bind9) neu. Ein einfacher Reload kann hier Fehler verursachen, aus diesem Grund, empfehlen wir hier ein Neustart.

Quellcode

1. service bind9 restart

Gratulation, wir haben es geschafft und die Domain kann nun über die eigenen NameServer verwendet und verwaltet werden. 😊

