

# Nameserver mit Bind unter Linux/Unix betreiben

Autor: Sebastian Wolfgarten, [sebastian@wolfgarten.com](mailto:sebastian@wolfgarten.com)

Homepage: <http://www.wolfgarten.com>

Erstellungsdatum: 13. März, 08:53 Uhr

## Einleitung:

Bind ist die Standardsoftware zum Betrieb von Nameservern unter Linux/Unix. Die vom Internet Software Consortium ([www.isc.org](http://www.isc.org)) entwickelte Software gibt es seit etlichen Jahren und ist inzwischen in der Version 9.2.0 erhältlich. Dieser Artikel beschreibt die Installation und Basiskonfiguration von Bind unter Linux anhand einfacher Beispiele.

## Installation:

Die aktuelle Version 9.2.0 gibt es u. a. auf <ftp://ftp.isc.org/isc/bind9/9.2.0/bind-9.2.0.tar.gz>. Die Datei entpacken wir durch Eingabe des Befehls „tar xvzf bind-9.2.0.tar.gz“ und rufen mit „./configure –help“ eine Übersicht der Konfigurationsoptionen auf. Ich habe mich für die nachfolgenden Optionen entschieden und die Software wie folgt konfiguriert:

```
./configure --prefix=/usr/local --localstatedir=/var --enable-threads --  
enable-libbind --sysconfdir=/etc --mandir=/usr/local/bind --  
infodir=/usr/local/bind --with-libtool --enable-shared --disable-ipv6"
```

Nach Abschluss der Konfiguration kann die Software durch Eingabe des Befehls „make“ und „make install“ übersetzt und installiert werden. Ein Test der Installation ist durch den Befehl „make test“ möglich.

## Konfiguration:

Viele Systeme (z.B. SuSE Linux) legen bei der Installation bereits eine Gruppe sowie einen Benutzer für den Bind an, auch wenn man die Software nicht installiert hat. Falls die Gruppe noch nicht existiert, kann diese durch den Befehl „groupadd named“ erzeugt werden. Ebenso verhält es sich mit dem Benutzer für den Bind, dieser kann durch Eingabe des Befehls „useradd -d /var/named -g named named“ angelegt werden.

Bevor wir mit der eigentlichen (sehr umfangreichen) Konfiguration beginnen, müssen wir erstmal das durch die Installation des Bind erstellte Verzeichnis /var/named an den neu eingerichteten Benutzer named durch den Befehl „chown -R named:named /var/named“ verschenken.

Danach legen wir mit „touch /etc/named.conf“ die eigentliche Konfigurationsdatei des Bind an. Neben dieser Datei brauchen wir mindestens noch zwei weitere Dateien, um den Bind vollständig zu installieren. Diese beiden Dateien erstellt der Befehl „touch /var/named/127.0.0.zone“ und „touch /var/named/localhost.zone“. Zusätzlich brauchen wir noch eine Datei für die

Definition der Rootzonen. Diese Datei kann man sich aus dem Internet durch den Befehl „**wget ftp://ftp.rs.internic.net/domain/named.root**“ ziehen und durch den Befehl „**mv named.root /var/named/root.hint**“ installieren. Falls der Nameserver als zweiter (sog. Secondary) Nameserver agieren soll, müssen wir noch das Verzeichnis **/var/named/slave** durch Eingabe des Befehls „**mkdir /var/named/slave**“ erzeugen.

Endlich können wir nun mit der eigentlichen Konfiguration beginnen und die Datei **/etc/named.conf** editieren. Mein Beispiel sieht wie folgt aus:

# Anmerkung: Alle Befehle muessen mit einem Semikolon und die  
# Konfigurationsdateien mit einer neuen Zeile abgeschlossen werden!

```
options {  
  
    # Arbeitsverzeichnis fuer Bind  
    directory "/var/named";  
  
    # DNS Abfragen koennen an externe Nameserver  
    # weitergeleitet werden, die sog. Forwarders  
    # (max. drei Stueck!) - dies ist beispielsweise  
    # dann sinnvoll, wenn man keine eigene Domains  
    # verwaltet.  
  
    forwarders {  
  
        195.158.131.2;  
        141.1.1.12;  
  
    };  
  
    # Wir moechten die DNS Anfragen direkt an die externen  
    # Nameserver weiterleiten, ohne unsere Nameserver  
    # abzufragen. Achtung: Nur aktivieren, wenn  
    # man keine eigenen Domains verwaltet!  
  
    forward first;  
  
    # Hier koennen beliebige IP Adressen und Ports angegeben werden, auf denen  
    # der Bind lauschen soll. Anstatt der IP Adresse kann auch „ANY“ angegeben  
    # werden, dadurch lauscht der Bind auf jeder lokalen IP Adresse.  
  
    listen-on port 53 { 127.0.0.1; };  
  
};  
  
# Dies sind drei Zonen fuer den Bind und  
# daran sollte nichts geaendert werden!  
# Sie definieren den lokalen Host (localhost),  
# die Reverse-Zone fuer den Localhost und eine  
# Definition fuer die Root Nameserver  
  
# Zone fuer Localhost
```

```

zone "localhost" in {
    type master;
    file "localhost.zone";
};

# Reverse Lookup fuer Localhost
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

# Zone fuer die Root-Server
zone "." in {
    type hint;
    file "root.hint";
};

```

Die Datei **/var/named/127.0.0.zone** für den Reverse Lookup der IP Adresse 127.0.0.1 auf den Localhost sieht bei mir wie folgt aus:

```

$TTL 1W
@      IN SOA      localhost.  root.localhost. (
        42        ; serial
        2D        ; refresh
        4H        ; retry
        6W        ; expire
        1W )      ; minimum

1      IN NS      localhost.
1      IN PTR     localhost.

```

Die erste Zeile gibt die sog. Time To Live (TTL) an, diese ist hier auf eine Woche gesetzt (1W) und gibt die Gültigkeit der Zonendefinition an. Dieser Wert braucht eigentlich nicht geändert werden.

Die zweite Zeile definiert, welcher Rechner und welche Person für welche Domain zuständig ist (SOA = State of Authority). Dabei stellt der erste Wert den Domainnamen dar bzw. in diesem Beispiel dient das „@“-Zeichen als Platzhalter für alle Domains. Der dritte Wert gibt den zuständigen Server für die Domain an, der letzte Wert dieser Zeile definiert die E-Mailadresse des Administrators dieser Domain, wobei das „@“-Zeichen der E-Mailadresse durch einen simplen Punkt ersetzt wird. **Wichtig ist bei allen Zonendefinitionen, dass alphanumerische Werte (Domainnamen, Servernamen etc.) immer mit einem Punkt abgeschlossen werden müssen (siehe die zweite und die letzten beiden Zeilen).** Die Zeilen drei bis sieben definieren diverse Werte der Gültigkeit einer solchen Zonendatei, wobei ich auf die einzelnen Werte später noch genauer eingehen werde.

Die vorletzte Zeile definiert einen universellen Nameserver für diese Domain, der auf dem Localhost läuft. Die letzte Zeile bestimmt die Auflösung der IP Adresse 127.0.0.1 auf den Hostnamen „localhost“. Das Auflösen von IP in Host- bzw. Domainnamen nennt man Reverse Lookup.

Die Datei `/var/named/localhost.zone` definiert die Auflösung des Namen localhost auf die IP Adresse 127.0.0.1 und sieht bei mir wie folgt aus:

```
$TTL 1W
@           IN SOA  @   root (
                42      ; serial
                2D      ; refresh
                4H      ; retry
                6W      ; expire
                1W )    ; minimum

                IN NS   @
                IN A    127.0.0.1
```

Die Datei ist genauso aufgebaut, wie die o. g. Zonendatei, wobei die letzten beiden Zeilen wiederum einen Nameserver definieren und den Namen „localhost“ in die IP Adresse 127.0.0.1 auflösen.

### Start:

Nun ist es endlich soweit: Wir können den Bind das erste Mal starten :-)  
Dazu geben wir folgenden Befehl ein: `„/usr/local/sbin/named -c /etc/named.conf“` und schauen uns durch Eingabe des Befehls `„tail -f /var/log/messages“` die Meldungen des Bind an:

```
Mar 13 13:12:29 serv3 named[18322]: starting BIND 9.2.0 -c /etc/named.conf
Mar 13 13:12:29 serv3 named[18322]: using 1 CPU
Mar 13 13:12:29 serv3 named[18324]: loading configuration from '/etc/named.conf'
Mar 13 13:12:29 serv3 named[18324]: listening on IPv4 interface eth0, 10.0.49.11#53
Mar 13 13:12:29 serv3 named[18324]: zone 0.0.127.in-addr.arpa/IN: loaded serial 42
Mar 13 13:12:29 serv3 named[18324]: zone localhost/IN: loaded serial 42
Mar 13 13:12:29 serv3 named[18324]: running
```

Sollten eventuelle Syntaxfehler in den Konfigurationsdateien existieren, würde der Bind sehr genau darauf hinweisen. Die o. g. Ausgabe ist aber so in Ordnung und wir können jetzt einen ersten Test machen. Dazu geben wir folgenden Befehl ein: `„/usr/local/sbin/nslookup localhost“`. Die Ausgabe dieses Befehls sollte in etwa so aussehen, ansonsten ist etwas mit den Zonendateien nicht in Ordnung:

```
serv1:/usr/local/sbin # nslookup localhost
Note: nslookup is deprecated and may be removed from future releases.
Consider using the `dig' or `host' programs instead. Run nslookup with
the `-sil[ent]` option to prevent this message from appearing.
Server:      127.0.0.1
Address:     127.0.0.1#53
```

```
Name: localhost
Address: 127.0.0.1
```

Dieser Befehl fragt den auf diesem Rechner laufenden Nameserver nach dem Host `„localhost“` ab und kriegt korrekterweise als Antwort 127.0.0.1 zurück.

Nun probieren wir einen Reverse Lookup der IP Adresse auf den Hostnamen localhost durch Eingabe des folgenden Befehls „**/usr/local/sbin/nslookup 127.0.0.1**“. Ebenso wie beim Auflösen des Hostnamen localhost auf die IP Adresse sollte die Ausgabe in etwa wie folgt aussehen, ansonsten sind die Zonendateien nicht in Ordnung:

```
serv1:/usr/local/sbin # nslookup 127.0.0.1
Note: nslookup is deprecated and may be removed from future releases.
Consider using the `dig' or `host' programs instead. Run nslookup with
the `-sil[ent]` option to prevent this message from appearing.
Server:      127.0.0.1
Address:     127.0.0.1#53
```

```
1.0.0.127.in-addr.arpa name = localhost.
```

Nun ist die Basiskonfiguration des Bind abgeschlossen und dieser steht nun in einer lauffähigen Version zur Verfügung. Die o. g. Konfiguration ist wirklich nur sehr knapp und minimal. Gerade Bind bietet eine schier unglaubliche Anzahl an Konfigurationsoptionen, mit denen man fast alles ändern und beeinflussen kann. Weitere Hilfe gibt u. a. das Bind Administrators Guide, welches kostenlos unter <http://www.nominum.com/resources/documentation/Bv9ARM.pdf> heruntergeladen werden kann.

### **Erweitere Konfiguration:**

In der oben vorgestellten Konfiguration leitet der Bind praktisch jede Anfrage an einen externen Nameserver weiter und ist für keine eigene Domain verantwortlich. In der Praxis wird Bind jedoch viel häufiger zur Verwaltung von eigenen Domains benutzt, wobei für jede Domain ein Eintrag in der **named.conf** vorgenommen werden muss. Außerdem muss für jede Domain eine eigene Zonendatei angelegt werden, optional ist eine Zonendatei für die Rückauflösung von IP Adressen auf Domainnamen. Hinweis: Der oben gemachte Eintrag „**forward first**“ sollte bei der Verwaltung von eigenen Domainnamen unbedingt gelöscht werden.

Dies will ich nun an einem konkreten Beispiel erläutern. Wir möchten gerne eine neue Domain wolfgarten.com verwalten und die IP Adresse 62.146.56.2 soll auf diese Domain verweisen. Diese Domain soll die Aliase www, ftp, pop, smtp, mail und einen universellen Eintrag haben.

Dazu legen wir in der Datei **/etc/named.conf** eine neue Sektion für wolfgarten.com an:

```
zone "wolfgarten.com" IN {
    type master;
    file "wolfgarten.com.db";
};
```

Dies definiert eine (Internet) Masterzone namens **wolfgarten.com.db** für die Domain wolfgarten.com. Eine Slavezone für einen Secondary Nameserver muss als Typ „**slave**“ mit Angabe des Masterservers enthalten. Nun legen wir mit „**touch /var/named/wolfgarten.com.db**“ die entsprechende Zonendatei an und füllen diese mit den entsprechenden Werten:

```

$TTL 2D
wolfgarten.com. IN SOA rootbeer.nexxium.de. sebi.rootbeer.nexxium.de. (
    2001100101 ; serial
    10800      ; refresh (3 hours)
    3600       ; retry (1 hour)
    604800    ; expire (1 week)
    86400     ; minimum (1 day)
)
NS ns1.nexxium.de.
NS ns2.nexxium.de.
MX 5 rootbeer.nexxium.de.
ftp A 62.146.56.2
mail A 62.146.56.2
pop A 62.146.56.2
www A 62.146.56.2

```

Diese Zonendatei entspricht in weiten Teilen der Zone für den Localhost, allerdings arbeiten wir hier mit richtigen Domainnamen. In den letzten fünf Zeilen habe ich zusätzlich noch den Eintrag „IN“ für die Aliase weggelassen, da dieser optional ist.

Die erste Zeile definiert die Time to Live, die Zweite gibt die Domain (wolfgarten.com) an, für die der Server (rootbeer.nexxium.de) zuständig ist, sowie die E-Mailadresse des Administrators (sebi.rootbeer.nexxium.de, =sebi@rootbeer.nexxium.de).

Die Zahlen in der dritten bis siebten Zeile machen Angaben über die Gültigkeit dieser Zone. Die dritte Zeile stellt die sog. Serial da, die man bei jeder Änderung erhöhen muss, damit der Bind Änderungen an der Zonendatei überhaupt wahrnimmt. Diese Serial kann ein einfacher numerischer Wert sein, oder sich aus dem aktuellen Jahr, dem Monat, dem Tag und einer fortlaufenden Nummer zusammensetzen (Beispielwert: 01.10.2001, entspricht 2001100101).

In der Zeile acht und neun werden die beiden Nameserver für diese Domain definiert. Der Eintrag in Zeile 10 ist der sog. Mail Exchange Record (MX) und definiert den Server, der für diese Domain die E-Mail empfängt. Durch die Angabe einer Zahl vor dem vollen Namen des Servers können Prioritäten festgelegt werden, wenn es sich um mehrere Server handelt.

Die letzten fünf Einträge definieren Aliase von Hostnamen auf die Zielipadresse 62.146.56.2. Der erste Eintrag davon ist der Universaleintrag, die restlichen Einträge sind jeweils selbstständige Aliase. Beispiel: Die letzte Zeile definiert den Alias für www.wolfgarten.com.

Wenn wir nun die Rückauflösung von IP Adressen auf Host- und Domainnamen erreichen wollen, müssen wir eine weitere Zonendatei durch den Befehl „**touch**“

`/var/named/62.146.56.zone`“ erstellen. Diese muss nun mit den entsprechenden Werten gefüllt werden:

```
$TTL 2D
56.146.62.in-addr.arpa. IN SOA  rootbeer.nexxium.de. sebastian.nexxium.de. (
    2001092901    ; serial
    1D            ; refresh
    2H            ; retry
    1W            ; expire
    2D )          ; minimum

                IN NS      ns1.nexxium.de.
                IN NS      ns2.nexxium.de.

2.56.146.62.in-addr.arpa.    IN PTR  rootbeer.nexxium.de.
2.56.146.62.in-addr.arpa.    IN PTR  ns1.nexxium.de.
```

Diese Zone unterscheidet sich eigentlich nicht wesentlich von den anderen Beispielen. Interessant sind die letzten beiden Zeilen sowie die zweite Zeile.

In der zweiten Zeile stehen praktisch die ersten drei Stellen der IP Adresse in (byteweiser) umgekehrter Reihenfolge, gefolgt von dem historisch gewachsenen Zusatz „**in-addr.arpa.**“ (nicht weglassen!). Die letzten beiden Zeilen definieren die Auflösung der vollen IP Adresse (62.146.56.2, in umgekehrter Reihenfolge) auf den Namen (rootbeer.nexxium.de).

Außerdem müssen wir in der Konfigurationsdatei `/etc/named.conf` noch einen neuen Eintrag für die neue Zone vornehmen. Dieser sieht wie folgt aus:

```
zone "56.146.62.in-addr.arpa" in {
    type master;
    file "62.146.56.zone";
};
```

Die Anlegung der neuen Domain ist nun fertig und der Bind muss nun neu gestartet werden. Eventuelle Fehlermeldungen erscheinen in `/var/log/messages`, man sollte auch durch die Tools „dig“, „host“ und „nslookup“ (veraltet) die neue Domain testen.

### Das (optionale) Sahnehäuptchen:

Der Bind ist nun lauffähig und funktioniert. Leider ist diese Software nicht sonderlich sicher und deshalb sollte man mithilfe des Parameters „**allow-query**“ die Abfragemöglichkeiten des Nameservers einschränken. Weitere Möglichkeiten der Absicherung sind u. a. die Verwendung einer sog. Chroot-Umgebung. Eine Chroot-Umgebung sperrt ein Programm bzw. einen User in eine feste Umgebung innerhalb des Dateisystems ein, ohne das dieser die Möglichkeit hat, diese Umgebung zu verlassen. Dabei werden Kopien aller wichtigen Systemteile dem Programm bzw. User in einem separaten Verzeichnis zur Verfügung gestellt und dort findet eine exklusive Nutzung dieser Systemteile durch den User bzw. das entsprechende Programm statt. Bricht ein Eindringling nun über ein Programm, welches in einer Chroot-Umgebung läuft, in das System ein, ist er ebenfalls in

diesem gefangen und kann diese Umgebung nur mit Administratorrechten verlassen.

Die Installation einer Chroot-Umgebung gestaltet sich leider etwas schwieriger als eine normale Installation, der Aufwand ist es allerdings wert. Zunächst brauchen wir eine funktionierende Installation des Bind. Danach erzeugen wir durch die Eingabe der nachfolgenden Befehle die Chroot-Umgebung mit den benötigten Unterverzeichnissen:

```
mkdir -p /var/named.chroot  
mkdir -p /var/named.chroot/etc  
mkdir -p /var/named.chroot/var/named  
mkdir -p /var/named.chroot/var/named/slave  
mkdir -p /var/named.chroot/var/run  
mkdir -p /var/named.chroot/bin  
mkdir -p /var/named.chroot/usr/sbin  
mkdir -p /var/named.chroot/usr/local/sbin  
mkdir -p /var/named.chroot/dev  
mkdir -p /var/named.chroot/lib
```

Danach müssen wir durch Eingabe von "**cp -p /etc/named.conf /var/named.chroot/etc/**" und "**cp -pR /var/named /var/named.chroot/var**" die Konfigurations- und Zonendateien kopieren.

Zusätzlich müssen wir durch die nachfolgenden Befehle die Berechtigungen korrekt setzen:

```
chown -R root:root /var/named.chroot/  
chmod 711 /var/named.chroot/*  
chown -R named:named /var/named.chroot/var/named/  
chown -R named:named /var/named.chroot/var/run/  
chmod 644 /var/named.chroot/etc/*  
chmod 755 /var/named.chroot/lib/*  
chmod 711 /var/named.chroot/usr  
chmod 711 /var/named.chroot/usr/sbin  
chmod 711 /var/named.chroot/usr/local/sbin
```

Außerdem braucht der Bind noch diverse Bibliotheken, damit er funktioniert. Diese kopieren wir auch in die Chroot-Umgebung durch Eingabe dieser Befehle:

```
cp -p /usr/local/lib/* /var/named.chroot/lib/  
cp -p /lib/libpthread.so.0 /var/named.chroot/lib/  
cp -p /lib/libc.so.6 /var/named.chroot/lib/
```

Neben diesen Dateien braucht der Bind noch weitere Dateien aus dem System. Diese kopieren wir ebenfalls:

```
cp -a /dev/null /var/named.chroot/dev/  
cp -a /dev/random /var/named.chroot/dev/  
cp /etc/localtime /var/named.chroot/etc/  
cp /etc/named.conf /var/named.chroot/etc/  
cp /usr/local/sbin/named* /var/named.chroot/usr/local/sbin/  
chmod 755 /var/named.chroot/usr/local/sbin/*
```

Selbstverständlich braucht eine Chroot-Umgebung auch eigene Passwortdateien. Diese erzeugt folgender Aufruf:

```
cat /etc/passwd | grep 'named' > /var/named.chroot/etc/passwd
cat /etc/group | grep 'named' > /var/named.chroot/etc/group
```

In der Datei `/var/named.chroot/etc/named.conf` muss das Arbeitsverzeichnis des Bind noch auf das neue Chroot-Verzeichnis geändert werden. Dazu wird der Eintrag `„directory "/var/named";“` in `„directory "/var/named.chroot";“` geändert. Schließlich muss noch ein symbolischer Link erzeugt werden, bevor der Bind in seiner Chroot-Umgebung gestartet werden kann: `„ln -s /var/named.chroot/var/run/named.pid /var/run/“`.

Der große Augenblick ist gekommen und wir können den Bind in unserer Chroot-Umgebung starten: `„/usr/local/sbin/named -u named -t /var/named.chroot -c /etc/named.conf“`. Etwaige Fehlermeldungen sollten in `/var/log/messages` entstehen, ansonsten kann durch den Befehl `„ps aux | grep named“` überprüft werden, ob der Bind läuft. Die Ausgabe dieses Befehls sollte in etwa so aussehen:

```
serv1:/ # ps aux | grep named
named 18809 0.0 0.8 9548 2104 ? S 14:23 0:00 /usr/local/sbin/named -u
named -t /var/named.chroot -d 5 -c /etc/named.conf
named 18810 0.0 0.8 9548 2104 ? S 14:23 0:00 /usr/local/sbin/named -u
named -t /var/named.chroot -d 5 -c /etc/named.conf
named 18811 0.0 0.8 9548 2104 ? S 14:23 0:00 /usr/local/sbin/named -u
named -t /var/named.chroot -d 5 -c /etc/named.conf
named 18812 0.0 0.8 9548 2104 ? S 14:23 0:00 /usr/local/sbin/named -u
named -t /var/named.chroot -d 5 -c /etc/named.conf
named 18813 0.0 0.8 9548 2104 ? S 14:23 0:00 /usr/local/sbin/named -u
named -t /var/named.chroot -d 5 -c /etc/named.conf
```

Durch die Tools `dig`, `host` und `nslookup` kann die korrekte Funktionsweise des Nameservers nun überprüft werden. Viel Spaß damit...