By Falko Timme
Published: 2008-05-27 18:18

Installing And Using The Unbound Name Server On Debian Etch

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 05/22/2008

Unbound is a validating, recursive, and caching DNS resolver, released under a BSD license. Version 1.0.0 was released on May 20, 2008. This tutorial explains how to install and use it on Debian Etch, including the creation of zones for your own domains.

This document comes without warranty of any kind! I want to say that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

1 Installing Unbound

Because there's no Debian package available yet, we have to install Unbound from the sources. First we install the prerequisites:

```
apt-get install build-essential libssl-dev
```

Then we download and install Unbound as follows:

```
cd /tmp

wget http://www.unbound.net/downloads/unbound-latest.tar.gz

tar xvfz unbound-latest.tar.gz

cd unbound-1.0.0/
```

```
./configure

make

make install
```

Next we create a user and group called *unbound*:

```
groupadd unbound

useradd -d /var/unbound -m -g unbound -s /bin/false unbound
```

We will use the directory */var/unbound* as the home directory of the Unbound name server - it will contain the Unbound configuration, and Unbound will run chrooted in it for security reasons.

Next we download the list of root name servers:

```
cd /var/unbound

wget ftp://ftp.internic.net/domain/named.cache
```

Then we create the directory */var/unbound/var/run* which will hold the Unbound PID file, *unbound.pid*, and create a symlink */var/run/unbound.pid* to it:

```
mkdir -p /var/unbound/var/run

chown -R unbound:unbound /var/unbound

ln -s /var/unbound/var/run/unbound.pid /var/run/unbound.pid
```

To start/stop/restart Unbound, we need an init script like this one:

```
vi /etc/init.d/unbound
```

```
#!/bin/sh
#
# unbound       This shell script takes care of starting and stopping
#               unbound (DNS server).

exec="/usr/local/sbin/unbound"
prog="unbound"
config="/var/unbound/unbound.conf"
pidfile="/var/run/unbound.pid"
rootdir="/var/unbound"

case "$1" in
   start)
      [ -x $exec ] || exit 5
      [ -f $config ] || exit 6
      echo -n $"Starting $prog: "

      # setup root jail
      if [ -s /etc/localtime ]; then
         [ -d ${rootdir}/etc ] || mkdir -p ${rootdir}/etc ;
         if [ ! -e ${rootdir}/etc/localtime ] || /usr/bin/cmp -s /etc/localtime ${rootdir}/etc/localtime; then
            cp -fp /etc/localtime ${rootdir}/etc/localtime
         fi;
      fi;
      if [ -s /etc/resolv.conf ]; then
         [ -d ${rootdir}/etc ] || mkdir -p ${rootdir}/etc ;
         if [ ! -e ${rootdir}/etc/resolv.conf ] || /usr/bin/cmp -s /etc/resolv.conf ${rootdir}/etc/resolv.conf; then
```

```
            cp -fp /etc/resolv.conf ${rootdir}/etc/resolv.conf
        fi;
    fi;
    if ! egrep -q '^/[^[:space:]]+[[:space:]]+'${rootdir}'/dev/log' /proc/mounts; then
        [ -d ${rootdir}/dev ] || mkdir -p ${rootdir}/dev ;
        [ -e ${rootdir}/dev/log ] || touch ${rootdir}/dev/log
        mount --bind -n /dev/log ${rootdir}/dev/log >/dev/null 2>&1;
    fi;
    if ! egrep -q '^/[^[:space:]]+[[:space:]]+'${rootdir}'/dev/random' /proc/mounts; then
        [ -d ${rootdir}/dev ] || mkdir -p ${rootdir}/dev ;
        [ -e ${rootdir}/dev/random ] || touch ${rootdir}/dev/random
        mount --bind -n /dev/random ${rootdir}/dev/random >/dev/null 2>&1;
    fi;

    # if not running, start it up here
    start-stop-daemon --start --quiet --pidfile $pidfile --exec $exec -- -c $config
    echo
    ;;

stop)
    echo -n $"Stopping $prog: "
    start-stop-daemon --stop --quiet --oknodo --pidfile $pidfile
    echo
    if egrep -q '^/[^[:space:]]+[[:space:]]+'${rootdir}'/dev/log' /proc/mounts; then
        umount ${rootdir}/dev/log >/dev/null 2>&1
    fi;
    if egrep -q '^/[^[:space:]]+[[:space:]]+'${rootdir}'/dev/random' /proc/mounts; then
        umount ${rootdir}/dev/random >/dev/null 2>&1
    fi;
    ;;

restart)
    start-stop-daemon --stop --quiet --oknodo --pidfile $pidfile
```

```
        start-stop-daemon --start --quiet --pidfile $pidfile --exec $exec -- -c $config
        ;;

    reload)
        start-stop-daemon --stop --signal 1 --quiet --oknodo --pidfile $pidfile --exec $exec
        ;;

    force_reload)
        start-stop-daemon --stop --signal 1 --quiet --oknodo --pidfile $pidfile --exec $exec
        ;;

    *)
        echo $"Usage: $0 {start|stop|restart|reload|force-reload}"
        exit 2
        ;;
esac

exit 0
```

Make the script executable and create the system startup links for it:

```
chmod 755 /etc/init.d/unbound

update-rc.d unbound defaults
```

That's it for the installation.

2 Configuring Unbound

Now we create the Unbound configuration file, */var/unbound/unbound.conf*. You can find a sample configuration file in */tmp/unbound-1.0.0/doc/example.conf* which has lots of explanations in it. You can also visit

http://www.unbound.net/documentation/unbound.conf.html to learn more about the Unbound configuration.

In the following configuration, I add two zones for domains (*example.com* and *example.net*) that I want to host on the Unbound name server. If you are familiar with the BIND name server, you can learn the Unbound syntax very fast. Adjust the zones to your needs, or leave them out if you only need a local resolver:

```
vi /var/unbound/unbound.conf
```

```
server:
        verbosity: 1
        interface: 0.0.0.0
        port: 53
        do-ip4: yes
        do-ip6: yes
        do-udp: yes
        do-tcp: yes
        do-daemonize: yes
        access-control: 0.0.0.0/0 allow
        #access-control: 0.0.0.0/0 refuse
        #access-control: 127.0.0.0/8 allow
        chroot: "/var/unbound"
        username: "unbound"
        directory: "/var/unbound"
        use-syslog: yes
        pidfile: "/var/run/unbound.pid"
        root-hints: "/var/unbound/named.cache"

        local-zone: "example.com." static
        local-data: "example.com. 86400 IN NS ns1.hostingcompany.com."
        local-data: "example.com. 86400 IN NS ns2.hostingcompany.com."
        local-data: "example.com. 86400 IN SOA ns1.hostingcompany.com. hostmaster.hostingcompany.com. 2008052201 28800 7200 604800 86400"
```

```
    local-data: "example.com. 86400 IN A 1.2.3.4"
    local-data: "www.example.com. 86400 IN CNAME example.com."
    local-data: "mail.example.com. 86400 IN A 1.2.3.4"
    local-data: "example.com. 86400 IN MX 10 mail.example.com."
    local-data: "example.com. 86400 IN TXT v=spf1 a mx ~all"

    local-zone: "example.net." static
    local-data: "example.net. 86400 IN NS ns1.hostingcompany.com."
    local-data: "example.net. 86400 IN NS ns2.hostingcompany.com."
    local-data: "example.net. 86400 IN SOA ns1.hostingcompany.com. hostmaster.hostingcompany.com. 2008052201 28800 7200 604800 86400"
    local-data: "example.net. 86400 IN A 1.2.3.4"
    local-data: "www.example.net. 86400 IN CNAME example.net."
    local-data: "mail.example.net. 86400 IN A 1.2.3.4"
    local-data: "example.net. 86400 IN MX 10 mail.example.net."
    local-data: "example.net. 86400 IN TXT v=spf1 a mx ~all"
```

I've used *`interface: 0.0.0.0`* here which means that Unbound listens on all network interfaces, and *`access-control: 0.0.0.0/0 allow`* which means that anybody can connect to Unbound. If you just want localhost to be allowed to connect, for example, you'd use

```
[...]
    access-control: 0.0.0.0/0 refuse
    access-control: 127.0.0.0/8 allow
[...]
```

instead.

To check if the syntax of your Unbound configuration is correct, you can use the *`unbound-checkconf`* command:

```
unbound-checkconf /var/unbound/unbound.conf
```

```
server1:~# unbound-checkconf /var/unbound/unbound.conf
  unbound-checkconf: no errors in /var/unbound/unbound.conf
server1:~#
```

If the syntax is ok, you can finally start Unbound:

```
/etc/init.d/unbound start
```

To learn more about Unbound, please refer to the **Unbound documentation**.

3 Links

- Unbound: **http://www.unbound.net/index.html**
- Debian: **http://www.debian.org**