# Implimenting secure DNS servers with OpenBSD and DJBDNS

This guide will assume you already have 2 servers each running OpenBSD. Each server will need 2 IP's, here is an example:

Server 1 – 10.0.0.2 – ns1
                10.0.0.4 – cache1

Server 2 – 10.0.0.3 – ns2
                10.0.0.5 – cache2

### Clean up the install

Disable all but one tty in /etc/ttys. I'm just anal about this, as it makes a **`ps`** a bit cleaner. You most likely won't be working from the console anyway :-P

Edit the **`/etc/rc.conf`** file. Set **`sendmail_flags=NO`**, **`inetd=NO`**. If you plan on limiting SSH access to only specific IP's set **`pf=YES`**.

Simple **`pf.conf`** to limit ssh access:

```
INT = xl0

block in log on $INT proto tcp from any to any port 22
pass in on $INT proto tcp from 192.168.1.1 to any port 22
pass in on $INT proto tcp from 10.0.0.2 to any port 22
```

Add yourself a user account and be sure to add yourself to the "wheel" group.

Edit **`/etc/sudoers`** with the **`visudo`** command. Delete everything except for the following line:

```
%wheel  ALL=(ALL)       ALL
```

Edit **`/etc/ssh/sshd_config`**, change **`Protocol`** to **`2`**, and **`PermitRootLogin`** to **`no`**

I normally do a **`chmod 700 /sbin/ping`**, **`/usr/bin/su`**, and **`/usr/sbin/traceroute`**. Even thou there probably won't be any regular users on the system, they don't need to be able to execute the above commands.

Reboot to allow all the above changes to take effect.

### Install daemontools and DJBDNS

Get **`ftp://ftp.openbsd.org/pub/OpenBSD/3.4/ports.tar.gz`** and **`untar`** to **`/usr`**

Get:

```
http://experimental.bug.it/tarballs/djbdns.tar.gz
http://experimental.bug.it/tarballs/daemontools.tar.gz
http://experimental.bug.it/tarballs/ucspi-tcp.tar.gz
```

And **`untar`** into **`/usr/ports`**

Before building the djbdns port, modify the **`Makefile`**

```
# $OpenBSD$
```

```
COMMENT=        "collection of Domain Name System tools"

DISTNAME=       djbdns-1.05
CATEGORIES=     experimental
MAINTAINER=     Giacomo Cariello <jwk@bug.it>
MASTER_SITES=   http://cr.yp.to/djbdns/ \
                ftp://ftp.id.wustl.edu/pub/qmail/
MASTER_SITES0=  ftp://ftp.innominate.org/gpa/djb/
HOMEPAGE=       http://cr.yp.to/djbdns.html
DISTFILES=      ${DISTNAME}${EXTRACT_SUFX}

PERMIT_PACKAGE_CDROM=   No
PERMIT_PACKAGE_FTP=     No
PERMIT_DISTFILES_CDROM= Yes
PERMIT_DISTFILES_FTP=   Yes

ALL_TARGET=     default
INSTALL_TARGET= setup check

RUN_DEPENDS=    supervise::experimental/daemontools \
                tcpclient::experimental/ucspi-tcp

pre-build:
        @echo ${CC} ${CFLAGS} > ${WRKSRC}/conf-cc
        @echo ${PREFIX} > ${WRKSRC}/conf-home

pre-install:
        @echo ${PREFIX} > ${WRKSRC}/conf-home

.include <bsd.port.mk>
```

and the **`pkg/PLIST`**

```
@comment $OpenBSD$
bin/axfr-get
bin/axfrdns
bin/axfrdns-conf
bin/dnscache
bin/dnscache-conf
bin/dnsfilter
bin/dnsip
bin/dnsipq
bin/dnsmx
bin/dnsname
bin/dnsq
bin/dnsqr
bin/dnstrace
bin/dnstracesort
bin/dnstxt
bin/pickdns
bin/pickdns-conf
bin/pickdns-data
bin/random-ip
bin/rbldns
bin/rbldns-conf
bin/rbldns-data
bin/tinydns
bin/tinydns-conf
bin/tinydns-data
bin/tinydns-edit
bin/tinydns-get
bin/walldns
bin/walldns-conf
share/djbdns/dnsroots.global
@exec if [ -f /etc/%f ]; then echo "\n!! WARNING !!\t/etc/%f exists\n\t\tPlease verify
with %D/%F" ; else /bin/cp %D/%F /etc; fi
@dirrm share/djbdns
@unexec echo "Please remove /etc/dnsroots.global manually"
```

Goto the `/usr/ports/experimental` dir and build each port, starting with ucspi-tcp, then daemontools, then djbdns. To build goto each dir and do a `make; make install`

Do the following:

```
groupadd dnscache
groupadd tinydns
groupadd dnslog
useradd -c "Djbdns Caching" -d /var/empty -g dnscache -s /sbin/nologin dnscache
useradd -c "Djbdns Resolving" -d /var/empty -g tinydns -s /sbin/nologin tinydns
useradd -c "Djbdns Logging" -d /var/empty -g dnslog -s /sbin/nologin dnslog
```

mkdir /var/service

Make /usr/local/bin/svscan.sh

```
#!/bin/sh -e

export PATH=$PATH:/usr/local/bin

    case "$1" in
       start)
            echo -n "Starting djb services: svscan "
            svscan /var/service &
            echo $! > /var/run/svscan.pid
            echo "."
            ;;
       stop)
            echo -n "Stopping djb services: svscan "
            kill `cat /var/run/svscan.pid`
            echo -n "services "
            svc -dx /var/service/*
            echo -n " logging "
            svc -dx /var/service/*/log
            echo "."
            ;;
       restart|reload|force-reload)
            $0 stop
            $0 start
            ;;
       *)
            echo 'Usage: /usr/local/bin/svscan.sh {start|stop|restart}'
            exit 1
       esac

       exit 0
```

Make the tinydns & dnscache default data dir with the following:

```
/usr/local/bin/tinydns-conf tinydns dnslog /var/tinydns 10.0.0.1
/usr/local/bin/dnscache-conf dnscache dnslog /var/dnscache 10.0.0.2
```

add a user "dnsadmin"

```
[16:24][root@ns1:~]# adduser
Use option ``-silent'' if you don't want to see all warnings and questions.

Reading /etc/shells
Reading /etc/login.conf
Check /etc/master.passwd
Check /etc/group

Ok, let's go.
Don't worry about mistakes. I will give you the chance later to correct any input.
Enter username []: dnsadmin
Enter full name []:
Enter shell bash csh ksh nologin sh [sh]: bash
Uid [1005]:
Login group dnsadmin [dnsadmin]:
Login group is ``dnsadmin''. Invite dnsadmin into other groups: guest no
[no]:
Login class auth-defaults auth-ftp-defaults daemon default staff
[default]:
Enter password []:
Set the password so that user cannot logon? (y/n) [n]: y

Name:         dnsadmin
Password:     ****
Fullname:     dnsadmin
Uid:          1005
Gid:          1005 (dnsadmin)
Groups:       dnsadmin
```

```
Login Class: default
HOME:        /home/dnsadmin
Shell:       /usr/local/bin/bash
OK? (y/n) [y]:
Added user ``dnsadmin''
Copy files from /etc/skel to /home/dnsadmin
Add another user? (y/n) [y]: n
Goodbye!
```

Set the permission correct:

```
chown -R dnsadmin /var/tinydins/root/
```

generate a private key with no pass phrase

```
[16:28][dnsadmin@ns1:~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/dnsadmin/.ssh/id_dsa):
Created directory '/home/dnsadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dnsadmin/.ssh/id_dsa.
Your public key has been saved in /home/dnsadmin/.ssh/id_dsa.pub.
```

do the above on your secondary DNS server too, then add your `~/.ssh/id_dsa.pub` on dns1 to `~/.ssh/authorized_keys` on dns2

create the file `/var/tinydns/root/makedns.sh`

```
#!/bin/sh

make
/usr/local/bin/rsync -az -e ssh data.cdb ns102:/var/tinydns/root/data.cdb
```

## Configure the DNS cache

If you want say the 10.0.0 network to be able to resolve from the DNS cache, just:

```
touch /var/dnscache/root/ip/10.0.0
ln –s /var/dnscache /var/service
svc –t /var/service/dnscache
```

## Configure the DNS server

```
ln –s /var/dnscache /var/service
```

Example tinydns format:

```
# Domain info dallaslamers.or

.dallaslamers.org::ns101.sprocketdata.com
.dallaslamers.org::ns102.sprocketdata.com
@dallaslamers.org::mail.dallaslamers.org
+dallaslamers.org:66.100.167.111
+www.dallaslamers.org:66.100.167.111
+ftp.dallaslamers.org:66.100.167.111
+mail.dallaslamers.org:66.100.167.111
+rivendell.dallaslamers.org:66.100.167.111
```

This would go in the `/var/tinydns/root/data` file, as would each domain afterwards. After to edit this file run the `makedns.sh` script from that directory.

Reference

http://www.openbsd.org/
http://www.djbdns.com/
http://cr.yp.to/djbdns.html