# Setting up an OpenDNSSEC server (july 2012)

*Written by Georg Sluyterman georg@sman.dk [mailto:georg@sman.dk]* *license: 2-clause BSD license*

The configuration adjustments are based on the defaults that are shipped with OpenDNSSEC.

The following is testet on Debian Squeeze (6.0).

## Installation

### SoftHSM

#### Install dependencies

```
apt-get install build-essential libbotan1.8-dev libsqlite3-dev sqlite3
```

#### Get and unpack SoftHSM

Get and unpack SoftHSM from https://www.opendnssec.org/download/ [https://www.opendnssec.org/download/]

```
./configure --enable-64bit
make
make install
```

### OpenDNSSEC

#### Install dependencies

```
apt-get install ruby rubygems libxml2-dev libsqlite3-dev libssl-dev libxml2-utils
```

When installing OpenDNSSEC 1.3.9 (for newer versions check release notes):

http://www.nlnetlabs.nl/downloads/ldns/ldns-1.6.13.tar.gz [http://www.nlnetlabs.nl/downloads/ldns/ldns-1.6.13.tar.gz]

```
./configure --disable-gost # GOST is a signature algorithm that may be used, but we are lazy ar
make
make install
```

When installing OpenDNSSEC 1.3.9 (for newer versions check release notes):

```
wget http://rubyforge.org/frs/download.php/75535/dnsruby-1.53.gem
gem install dnsruby-1.53.gem
```

#### Get and install OpenDNSSEC

Get OpenDNSSEC from https://www.opendnssec.org/download/ [https://www.opendnssec.org/download/]

```
./configure
make
make install
```

## Configuration

### SoftHSM

Configuration file: **/etc/softhsm.conf**. By default no editing of this file is necessary.

**Initial configuration:**

Enter som hard to guess PIN for SO and CU (use the same PIN for both, otherwise enforcerd seems to be unhappy..). It is be used later when configuring OpenDNSSEC. The label for the token may be chosen freely - it is used later when configuring OpenDNSSEC.

```
softhsm --init-token --slot 0 --label "dnssec-token01"
chown opendnssec:opendnssec /var/softhsm/slot0.db
chmod 600 /var/softhsm/slot0.db
```

## OpenDNSSEC

### conf.xml

For backup of HSM's, see the smart tricks on: https://wiki.opendnssec.org/display/DOCS/Key+Management [https://wiki.opendnssec.org/display/DOCS/Key+Management]

Fill out *TokenLabel* and *PIN* (Crypto User/CU PIN)

```
<Configuration>
        <RepositoryList>
                <Repository name="SoftHSM">
                        <Module>/usr/local/lib/softhsm/libsofthsm.so</Module>
                        <TokenLabel>dnssec-token01</TokenLabel>
                        <PIN>1234</PIN>
```

For higher security remove the comments for the <*Privileges*> section for all the components

Remove or outcomment the entire <Auditor> section. Auditor is deprecated from version 1.4. Insted use e.g. validns (or ods-auditor which will most probably will be continued as a seperate project).

Remove comments for <*NotifyCommand*> under <*Signer*>. Adapt to your needs; e.g. rndc reload if OpenDNSSEC is running on the master DNS server. In the following example a small script is used that checks the given zone with validns, moves the file to the master name server with scp and runs *rndc reload* for the given zone. If any step failes, the scripts sends an e-mail with the error and aborts.

```
<NotifyCommand>/usr/local/bin/check_and_distribute_zones.sh %zone</NotifyCommand>
```

See below (at the end) for an example for *check_and_distribute_zones.sh*

### Add opendnssec user

Since we do not wish to run OpenDNSSEC as root we need a user ods.

```
groupadd -g 2005 opendnssec
useradd -c "OpenDNSSEC pseudo user" -g opendnssec -s /bin/sh -u 2005 -d /var/opendnssec opendns
```

Adjust som file permissions and ownership since we do not run as root:

```
cd /etc/opendnssec
chown opendnssec:opendnssec zonelist.xml
chmod 644 conf.xml
chown opendnssec:opendnssec /var/softhsm/
chown -R opendnssec:opendnssec /var/opendnssec
```

### kasp.xml

If you want to use NSEC insetad of NSEC3: Remove the <NSEC3> part and insert only: <*NSEC*/>

A note to the NSEC3 hashing algorithm: Only *1* is supported at the time of writing (SHA-1 hashes.

*Some comment i made in the kasp.xml*

```
<Resign>PT2H</Resign> <!-- How often the signer checks wether it should sign anything  -->
<Refresh>P3D</Refresh> <!-- Time before expiration of signatures before a new signature is made
```

For *key signing keys* and *zone signing keys* we adjusted the following parameters:

- KSK: 4K keys, alg. 8 (RSA/SHA-256)
- ZSK: 2K keys, alg. 8 (RSA/SHA-256)

<*Serial*>: datecounter

Adjustments for .dk domains (DK-Hostmaster):

```
                <Parent>
                        <PropagationDelay>PT43200S</PropagationDelay>
                        <DS>
                                <TTL>P1D</TTL>
                        </DS>
                        <SOA>
                                <TTL>P1D</TTL>
                                <Minimum>PT3600S</Minimum>
                        </SOA>
                </Parent>
```

Remove or outcomment the <Audit> section

**zonelist.xml**

Generated by OpenDNSSEC, but you can modify it for your needs

**zonefetch.xml**

Delete the <ZoneFetch> section (only needed if you need to fetch your zones via AXFR).

# Running OpenDNSSEC

Se more at https://wiki.opendnssec.org/display/DOCS/Running+OpenDNSSEC [https://wiki.opendnssec.org/display/DOCS/Running+OpenDNSSEC]

Before you run the system for the first time you must import your policy and zone list into the database:

```
ods-ksmutil setup
```

## Startup scripts

OpenDNSSEC consist of two daemons, ods-signerd and ods-enforcerd. To start and stop them you may use the following commands:

```
ods-control start
```

But instead we copy the scripts from the Debian package and adjust the path:

**/etc/init.d/opendnssec-signer**

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          opendnssec-signer
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      0 1 6
# Short-Description: OpenDNSSEC Signer
# Description:       Daemon to periodically sign DNSSEC zone files.
### END INIT INFO

# Author: Ondřej Surý <ondrej@debian.org>
#
# Do NOT "set -e"

# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="OpenDNSSEC Signer"
NAME=ods-signerd
DAEMON=/usr/local/sbin/$NAME
DAEMON_ARGS=""
PIDFILE=/var/run/opendnssec/signerd.pid
SCRIPTNAME=/etc/init.d/opendnssec-signer

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
. /lib/lsb/init-functions

#
# Function to create piddir if it doesn't exists
#
create_piddir() {
        PIDDIR="$(dirname $PIDFILE)"
        [ -d "$PIDDIR" ] && return 0
        mkdir -p "$PIDDIR" || return 1
        chown opendnssec:opendnssec "$PIDDIR" || return 1
}

#
# Function that starts the daemon/service
#
do_start()
{
        # Return
        #   0 if daemon has been started
        #   1 if daemon was already running
        #   2 if daemon could not be started
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test > /dev/null
                || return 1
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON -- \
                $DAEMON_ARGS \
                || return 2
}

#
# Function that stops the daemon/service
#
do_stop()
{
        # Return
        #   0 if daemon has been stopped
        #   1 if daemon was already stopped
        #   2 if daemon could not be stopped
        #   other if a failure occurred
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE
        RETVAL="$?"
```

```
        [ "$RETVAL" = 2 ] && return 2

        # Many daemons don't delete their pidfiles when they exit.
        rm -f $PIDFILE
        return "$RETVAL"
}

create_piddir

case "$1" in
  start)
        [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
        do_start
        case "$?" in
                0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
                2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
        esac
        ;;
  stop)
        [ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
        do_stop
        case "$?" in
                0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
                2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
        esac
        ;;
  status)
       status_of_proc "$DAEMON" "$NAME" && exit 0 || exit $?
       ;;
  restart|force-reload)
        log_daemon_msg "Restarting $DESC" "$NAME"
        do_stop
        case "$?" in
          0|1)
                do_start
                case "$?" in
                        0) log_end_msg 0 ;;
                        1) log_end_msg 1 ;; # Old process is still running
                        *) log_end_msg 1 ;; # Failed to start
                esac
                ;;
          *)
                # Failed to stop
                log_end_msg 1
                ;;
        esac
        ;;
  *)
        #echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
        echo "Usage: $SCRIPTNAME {start|stop|status|restart|force-reload}" >&2
        exit 3
        ;;
esac

:
```

**/etc/init.d/opendnssec-enforcer**

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          opendnssec-enforcer
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: OpenDNSSEC Enforcer
# Description:       Daemon to create and enforce DNSSEC KASP policy
### END INIT INFO
```

```
# Author: Ondřej Surý <ondrej@debian.org>
#
# Do NOT "set -e"

# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="OpenDNSSEC Enforcer"
NAME=ods-enforcerd
DAEMON=/usr/local/sbin/$NAME
DAEMON_ARGS=""
PIDFILE=/var/run/opendnssec/enforcerd.pid
SCRIPTNAME=/etc/init.d/opendnssec-enforcer

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
. /lib/lsb/init-functions

#
# Function to create piddir if it doesn't exists
#
create_piddir() {
        PIDDIR="$(dirname $PIDFILE)"
        [ -d "$PIDDIR" ] && return 0
        mkdir -p "$PIDDIR" || return 1
        chown opendnssec:opendnssec "$PIDDIR" || return 1
}

#
# Function that starts the daemon/service
#
do_start()
{
        # Return
        #   0 if daemon has been started
        #   1 if daemon was already running
        #   2 if daemon could not be started
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test > /dev/null
                || return 1
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON -- \
                $DAEMON_ARGS \
                || return 2
}

#
# Function that stops the daemon/service
#
do_stop()
{
        # Return
        #   0 if daemon has been stopped
        #   1 if daemon was already stopped
        #   2 if daemon could not be stopped
        #   other if a failure occurred
        start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE --name $NAME
        RETVAL="$?"
        [ "$RETVAL" = 2 ] && return 2

        # Many daemons don't delete their pidfiles when they exit.
        rm -f $PIDFILE
```

```
        return "$RETVAL"
}

create_piddir

case "$1" in
  start)
        [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
        do_start
        case "$?" in
                0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
                2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
        esac
        ;;
  stop)
        [ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
        do_stop
        case "$?" in
                0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
                2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
        esac
        ;;
  status)
        status_of_proc "$DAEMON" "$NAME" && exit 0 || exit $?
        ;;
  restart|force-reload)
        log_daemon_msg "Restarting $DESC" "$NAME"
        do_stop
        case "$?" in
          0|1)
                do_start
                case "$?" in
                        0) log_end_msg 0 ;;
                        1) log_end_msg 1 ;; # Old process is still running
                        *) log_end_msg 1 ;; # Failed to start
                esac
                ;;
          *)
                # Failed to stop
                log_end_msg 1
                ;;
        esac
        ;;
  *)
        #echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
        echo "Usage: $SCRIPTNAME {start|stop|status|restart|force-reload}" >&2
        exit 3
        ;;
esac

:
```

**Start ods**

```
update-rc.d opendnssec-signer defaults
update-rc.d opendnssec-enforcer defaults
service opendnssec-signer start
service opendnssec-enforcer start

$ ps ax | grep od[s]
25204 ?        Ss     0:00 /usr/local/sbin/ods-enforcerd
25211 ?        Ssl    0:00 /usr/local/sbin/ods-signerd
```

## New zone

```
ods-ksmutil zone add --zone <zone>
ods-ksmutil update zonelist # signing is started right away
```

Immediate signing:

```
ods-signer reload
ods-signer sign <zone>
```

#### Get DS keys

```
ods-ksmutil key export --zone <zone> --ds
```

# Checking and pushing zone files

## validns

If you wish to check your zone files before deploying them.

```
wget http://www.validns.net/download/validns-0.5.tar.gz
tar zxvf validns-0.5.tar.gz
cd validns-0.5
apt-get installlibjudy-dev && cpan -i Test::Command::Simple
make
cp validns /usr/local/bin/
```

## check_and_distribute_zones.sh

```
apt-get install bsd-mailx
```

```
#!/bin/bash
# Check and distribute zone files from a signing host to a (remote) name server
# Georg Sluyterman <georg@sman.dk>
# License: 2-clause BSD license
# Version 0.1, 2012-07-22

MAIL_RCPT="hostmaster@sman.dk"
SIGNED_ZONES_DIR="/var/opendnssec/signed/"
DST_HOST="masterdns.sman.dk"
DST_FOLDER="/etc/bind/zones/"
TMP_FILE="/tmp/check_and_distribute_zones_lastcmd.log"

prg="$0"
zone="$1"

# Basic input validation
if [ "$#" -ne 1 ]
then
    echo "Usage: $0 <zone>"
    echo "Edit $0 for properties."
    exit 1
fi

# Truncate output file
echo > "${TMP_FILE}"

# Used when an error occurred
err_msg() {
    cat "${TMP_FILE}" | mailx -s "$1" ${MAIL_RCPT}
    echo "$1" >&2
    cat "${TMP_FILE}" >&2
    rm "${TMP_FILE}"
    exit 1
}

/usr/local/bin/validns "${SIGNED_ZONES_DIR}/${zone}" > "${TMP_FILE}" 2>&1
if [ $? -eq 0 ]
then
    scp "${SIGNED_ZONES_DIR}/${zone}" root@${DST_HOST}:${DST_FOLDER} > "${TMP_FILE}" 2>&1
    if [ $? -ne 0 ]
```

```
    then
        err_msg "$prg: failed copying zone ${zone} to host ${DST_HOST}"
    fi
    ssh root@${DST_HOST} "rndc reload ${zone}" > "${TMP_FILE}" 2>&1
    if [ $? -ne 0 ]
    then
        err_msg "$prg: failed to reload zone ${zone} on host ${DST_HOST}"
    fi
else
    err_msg "$prg: validns found an error in zone ${zone}. Zone is not pushed to name servers"
fi
rm -f "${TMP_FILE}"
```

---

- tech/infrastructure/servers/ods.sman.dk.txt · Last modified: 2012/07/26 21:49 by georg
-